VETO: A SOUND SAFETY ENVELOPE FOR REINFORCEMENT LEARNING UNDER DISTRIBUTION SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety envelopes filter the actions of a reinforcement learning policy through a predicted cost bound, and their appeal is that the bound is checked at every step. However, we find that the natural envelope fails in two ways under distribution shift. First, its fallback path reuses actions certified under an older calibration or at a different state, so the envelope is unsound and can emit an action its own bound rejects. Second, its calibration is fitted once, so the coverage the bound promises drifts with the deployment distribution. We introduce VETO, which combines a re-check of every emitted action at the current monitor state, a single temperature refit online from a scalar drift statistic, and a margin on the ensemble spread. Soundness then holds for any cost model and is machine-checked in the Rocq prover, and coverage transfer reduces to a Kolmogorov–Smirnov bound on one scalar, because coverage is a threshold event, which is why one temperature replaces per-step quantile tracking. In a pilot on two Safety-Gymnasium tasks with three constrained-optimization base policies, VETO lowers the per-step violation rate in 58 of 60 evaluation cells. The naive fallback emits inadmissible actions on up to 75.2% of steps, and a conformal filter with the same ensemble and no re-check has a violation rate at or above VETO's in 54 of 60 cells. The price is an intervention rate that we report cell by cell. A sound envelope with online calibration is therefore what lets a per-step guarantee survive shift.

1 INTRODUCTION

A trained controller can satisfy its safety constraint on average and still fail on the steps that matter (Altman, 1999; Ray et al., 2019; García & Fernández, 2015). Consider a cellular scheduler that satisfies its latency constraint in expectation while the traffic it sees stays close to training. During a stadium event the traffic shifts, the policy starts dropping packets, and each drop breaks a service agreement. At 100 Hz the policy makes 86,400 decisions a day, a 1% per-step violation rate is 864 broken agreements, and the average constraint remains satisfied throughout. The gap is not special to networks. A manipulator that meets its force constraint in expectation still exceeds the force limit on some grasps once friction changes, which is the failure Recovery RL was built for (Thananjeyan et al., 2021), and a driving policy that holds its average margin in training breaks the minimum following distance in traffic it never saw (Leurent, 2018). Figure 1(a) makes the gap concrete, with an episode whose mean cost is exactly $0.5d$ and crosses d on 11% of its steps; among cost laws with that mean the per-step violation probability can be anywhere up to 50%, the Markov bound $\mathbb{E}[c]/d$, which is the only bound an average constraint implies (Appendix D). A safety envelope, a layer that checks a predicted cost bound before each action executes, controls the per-step event instead.

The natural envelope predicts the cost and its spread with an ensemble (Lakshminarayanan et al., 2017), passes an action when its upper bound clears a threshold, projects or replaces a failed action, and calibrates the bound once on training data. Under shift we find that this design fails in two ways, shown in Figure 1(b).

Stale certificates. When projection fails, the natural envelope falls back on an action it admitted earlier, and that action carries a certificate issued under an older calibration or at a different state. Stated over the envelope's own changing state, soundness has content, and under that reading the

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.

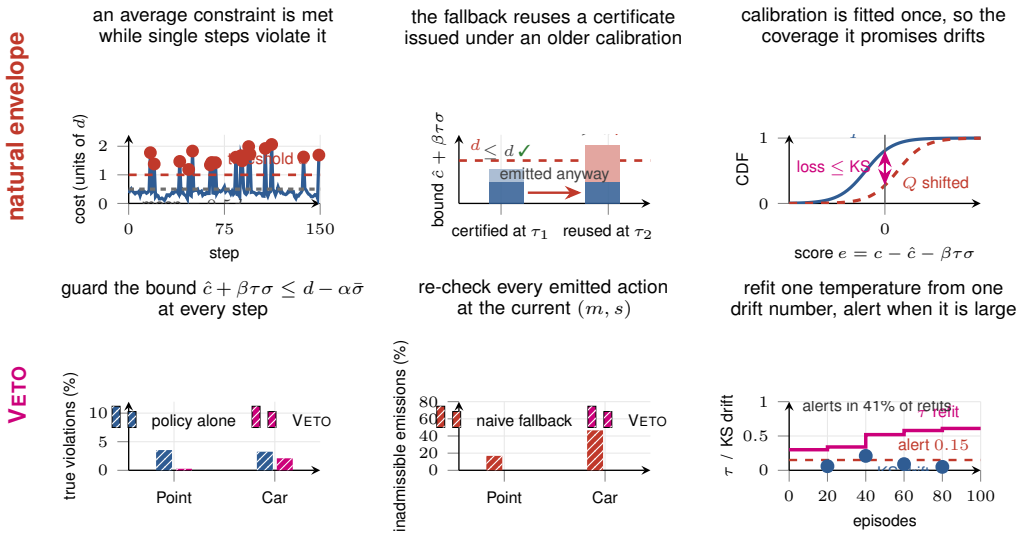


Figure 1: **VETO repairs two failure modes of the natural safety envelope with two changes. Left.** An average constraint is met while single steps violate it, on a measured episode with mean exactly $0.5d$; VETO guards the bound at every step, and the bars are measured violation rates without shift. **Middle.** The fallback reuses a certificate from an older calibration (schematic); VETO re-checks every emission at the current state, and the bars are the measured share of inadmissible emissions, naive fallback against VETO's zero. **Right.** A calibration fitted once lets coverage drift, by at most the Kolmogorov–Smirnov distance between score distributions (schematic); VETO refits one temperature from that number and alerts above 0.15 (schematic timeline), which fired in 41% of the pilot's refits.

natural design is unsound, because a fallback can reuse a certificate that the current state rejects. No proof of its soundness can exist without assuming something about the cost model (Proposition 1). The fix is one line, a re-check of every emitted action at the current monitor and environment state, and it makes soundness provable for any cost model (Theorem 3).

Drifting calibration. A predictor that flags unsafe actions is only as good as its calibration, and calibration is what a shift destroys (Ames et al., 2017). Our key insight is that *coverage under shift is a threshold event on a single scalar*. The event that the true cost stays under the bound, $\{c \le \mathrm{ucb}\}$, is $\{e \le 0\}$ for the score $e = c - \hat{c} - \beta\tau\sigma$. For a threshold event the Kolmogorov–Smirnov distance between two score distributions, the largest gap between their cumulative distributions, bounds the change in coverage directly,

$$|P_T(e \le 0) - Q(e \le 0)| \;\le\; \mathrm{KS}(P_T, Q), \tag{1}$$

which is immediate from the definition of that distance (Massey, 1951). Drift in one number therefore bounds the loss in safety. Maintaining calibration then reduces to watching that scalar and turning one temperature, a scalar that widens the bound, so the decision path carries one number rather than the window a per-step quantile needs (Theorem 6).

In this paper we introduce VETO, which addresses both failures with the re-check, a temperature refit every 20 episodes from the drift statistic, and a margin proportional to the running spread. We keep its two guarantees deliberately separate, since soundness fails through a logic error and is repaired in the algorithm whereas coverage fails through drift and is repaired by recalibration. Section 2 places VETO against constrained optimization, which controls an average (Achiam et al., 2017), against hard-constraint layers fitted before deployment (Alshiekh et al., 2018; Ames et al., 2019), and against the adaptive methods closest to it (Gibbs & Candès, 2021; Hamel-De le Court et al., 2025).

Our contributions follow this diagnosis. First, we separate two guarantees that are usually reported as a single safety claim, an unconditional soundness property of the envelope, independent of prediction quality, and a statistical guarantee on true cost that is conditional on assumptions VETO tests at runtime (Sections 5 and 6). Second, we prove that the naive envelope is unsound through calibration staleness and state dependence (Proposition 1), that monotonicity removes only the first obstacle

(Lemma 2), and that a re-check at emission repairs both for any cost model (Theorem 3). Third, we mechanize these results in the Rocq prover (The Rocq Development Team, 2025), an interactive proof checker, against an uninterpreted cost model, together with the deterministic step of the coverage transfer (Appendix B). Fourth, we show that coverage transfer under shift reduces to a Kolmogorov–Smirnov bound on threshold events (Theorem 6), which yields a decision path with $O(1)$ state and time against the $O(W)$ window a per-step quantile update needs. Fifth, we evaluate the envelope around PPO-Lagrangian, CPO and PID-Lagrangian policies on Safety-Gymnasium PointGoal and CarGoal under dynamics, observation-noise and rare-event shifts, with two controls that isolate the re-check, the naive fallback and a conformal filter built on the same ensemble, and we release the code with every per-step log (Section 7).

2 Related Work

Constrained reinforcement learning and shielding. Constrained policy optimization and its descendants enforce an expected-cost constraint (Achiam et al., 2017; Stooke et al., 2020; Zhang et al., 2020; As et al., 2022; Gao et al., 2025; García & Fernández, 2015). Step-wise formulations target per-step violations (Xiong et al., 2023), primal–dual methods pursue zero violation (Bai et al., 2022), and probabilistic constraints generalize the expectation (Chen & Paternain, 2024). These shape the policy during training, whereas VETO wraps a trained policy and asks what still holds once the deployment distribution has moved. Shields synthesize a filter from a model of the environment (Alshiekh et al., 2018; Könighofer et al., 2020; Carr et al., 2023), probabilistic shielding certifies against an abstraction (Hamel-De le Court et al., 2025), and adaptive shielding tracks hidden-parameter dynamics shifts through a parametric model of the shift and filters with conformal bounds (Kwon et al., 2026). Adaptive shielding is the closest method to ours, and we did not run it because its shield infers the hidden parameter from a family of training environments that our single-environment policies never saw. Safety layers project actions with a learned constraint model (Dalal et al., 2018), control barrier functions certify forward invariance, that the state never leaves a safe set, given a dynamics model (Ames et al., 2017; 2019; Cheng et al., 2019), predictive safety filters add model predictive control (Wabersich & Zeilinger, 2023), and Recovery RL learns a recovery policy (Thananjeyan et al., 2021). All certify against a model fixed before deployment or adapted within a parametric family. VETO assumes no dynamics model and certifies its own decision logic for every cost model.

Conformal prediction for control. Conformal prediction turns held-out residuals into a distribution-free bound, and adaptive conformal inference keeps that bound covering under shift by tracking a quantile online (Gibbs & Candès, 2021). Conformal prediction has also been used for runtime verification of signal temporal logic (Lindemann et al., 2023; Cleaveland et al., 2024), made robust to distribution shift (Zhao et al., 2024), turned into a safety filter by conformalizing predicted trajectories (Strawn et al., 2023), and lifted to decisions (Lekeufack et al., 2024). VETO shares the distribution-free spirit but needs no dynamics model, certifies its own decision logic, and tracks a single scalar whose drift a Kolmogorov–Smirnov distance bounds, which is what keeps its decision path at constant state.

Formal methods and mechanized proofs. Justified speculative control combines differential dynamic logic with learning (Fulton & Platzer, 2018; 2019), inductive synthesis produces verifiable shields from neural policies (Zhu et al., 2019), and CertRL mechanizes convergence proofs for value and policy iteration in Coq (Vajjha et al., 2021). To our knowledge, no prior envelope machine-checks a soundness property independent of the cost model while tracking calibration under shift with constant state on the decision path.

3 Preliminaries

We work in a constrained Markov decision process $(\mathcal{S}, \mathcal{A}, P, r, c, d, \gamma)$ (Altman, 1999), where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, P the transition kernel, r the reward, d the safety threshold, and $\gamma \in [0, 1)$ the discount factor. The cost $c : \mathcal{S} \times \mathcal{A} \to [0, c_{\max}]$ we call the true cost, to distinguish it from any prediction of it. The standard objective maximizes $\mathbb{E}[\sum_t \gamma^t r_t]$ subject to $\mathbb{E}[\sum_t \gamma^t c_t] \leq d$.

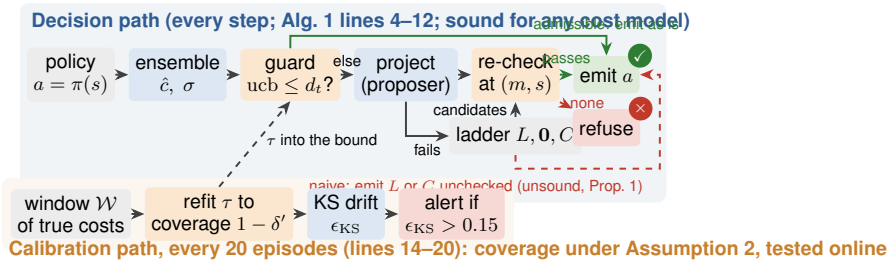


Figure 2: VETO, whose decision path runs every step and whose calibration path runs every 20 episodes, with the natural envelope's fallback as the dashed red path.

That constraint is an average, so a policy with $\mathbb{E}[\sum_t c_t] = 0.5d$ may still exceed $c(s_t, a_t) > d$ on a large fraction of steps, and we ask for something stronger.

Definition 1 (Zero-violation safety)**.** A policy π is (T, H, δ)-zero-violation safe if over T episodes of horizon H,

$$\mathbb{P}\big[\,\forall t \in [T],\ h \in [H] : c(s_t^h, a_t^h) \le d\,\big] \ge 1 - \delta.$$

The pilot of Section 7 evaluates $T = 100$ or 50 episodes of horizon $H = 200$ per seed and cell, 10,000 to 20,000 decisions. It targets the per-step miscoverage $\delta' = 0.02$ of Section 6 rather than a trajectory-level δ, which at this budget would need a bound far tighter than an ensemble provides.

VETO estimates cost with an ensemble of $K = 5$ three-layer perceptrons with 256 hidden units, trained on bootstrap resamples with Adam at learning rate 3×10^{-4} (Lakshminarayanan et al., 2017; Kingma & Ba, 2015), giving a mean $\hat{c}(s, a)$ and a spread $\sigma(s, a)$ that grows where the ensemble has seen little data.

Definition 2 (Calibrated bound)**.** For temperature $\tau \ge 1$ and base multiplier $\beta > 0$, the calibrated upper confidence bound is $\mathrm{ucb}(\tau, s, a) = \hat{c}(s, a) + \beta\tau\sigma(s, a)$.

Definition 3 (Nonconformity score)**.** The nonconformity score is $e = c - \hat{c} - \beta\tau\sigma$. The coverage event $\{c \le \mathrm{ucb}\}$ is exactly the threshold event $\{e \le 0\}$.

Shift protocol. We evaluate under a fixed protocol of three shift types (dynamics, observation noise, rare-event impulses) at magnitudes $m \in \{0.5, 1.0, 2.0\}$ relative to training, applied at evaluation time only, with type, magnitude, and seed recorded for every run (Appendix E).

4 THE VETO ENVELOPE

Both guarantees of the introduction are properties of an object with state, and this section defines that object. VETO sits between a base policy and the environment; we call the system an envelope and its formal object a monitor. The monitor holds state of its own, which we make explicit because that is what lets us reason about it.

Definition 4 (Monitor state)**.** A monitor state is a tuple $m = (\tau, \bar{\sigma}, L, C, k)$, where $\tau \ge 1$ is the calibration temperature, $\bar{\sigma} \ge 0$ is the running mean ensemble spread, L is the last admitted action, C is a cache of previously certified actions, and k is the episode counter that triggers recalibration; k only schedules recalibration and does not enter the admissible set below.

Definition 5 (Admissible set)**.** Given monitor state m and environment state s, the admissible set is $\mathrm{Adm}(m, s) = \{a \in \mathcal{A} : \mathrm{ucb}(m.\tau, s, a) \le d_t(m.\bar{\sigma})\}$, where the adaptive threshold is $d_t(\bar{\sigma}) = d - \alpha\bar{\sigma}$ with margin $\alpha > 0$.

The margin $\alpha\bar{\sigma}$ leaves room for prediction error near the boundary, and Adm depends on both the monitor state and the environment state, which is what matters in the following section.

Algorithm 1 gives the procedure and Figure 2 shows what each of its two paths is responsible for. The decision path, lines 4–12, reads only τ from the calibration path and touches the window only to append the observed cost. Its state is therefore constant, and Section 5 can certify it without looking at the network. The calibration path, lines 14–20, is where Assumption 2 and the statistical guarantee

Algorithm 1 VETO, where line 9 is the re-check that makes emission sound.

Require: policy π, ensemble $\{f_k\}_{k=1}^{K}$, threshold d, window W
1: $\tau \leftarrow 1.0$; $\mathcal{W} \leftarrow \emptyset$
2: **for** episode $t = 1, \ldots, T$ **do**
3: **for** step $h = 1, \ldots, H$ **do**
4: $a \leftarrow \pi(s)$
5: $\hat{c}, \sigma \leftarrow \textsc{Ensemble}(s, a)$
6: **if** $\mathrm{ucb}(\tau, s, a) > d_t(\bar{\sigma})$ **then**
7: $a \leftarrow \textsc{Project}(s, a)$ ▷ untrusted proposer
8: **end if**
9: **if** $\mathrm{ucb}(\tau, s, a) > d_t(\bar{\sigma})$ **then**
10: $a \leftarrow \textsc{Fallback}(s, \tau, \bar{\sigma}, L, C)$ ▷ re-check at emission
11: **end if**
12: execute a; observe true cost c; store in $\mathcal{W}$
13: **end for**
14: **if** $t \bmod 20 = 0$ **then**
15: $\tau \leftarrow \textsc{Recalibrate}(\mathcal{W}, 1 - \delta')$
16: $\epsilon_{\mathrm{KS}} \leftarrow \textsc{KSDrift}(\mathcal{W}_{\mathrm{old}}, \mathcal{W}_{\mathrm{new}})$
17: **if** $\epsilon_{\mathrm{KS}} > 0.15$ **then**
18: raise drift alert
19: **end if**
20: **end if**
21: **end for**

of Section 6 live. The dashed red path is the natural envelope's fallback, which Section 5 shows to be unsound. At each step the base policy proposes an action. If it lies outside $\mathrm{Adm}(m, s)$, VETO searches for a nearby admissible action by minimizing $\|a' - a\|_2^2$ subject to $\mathrm{ucb}(m.\tau, s, a') \le d_t(m.\bar{\sigma})$, by gradient descent on the bound surface with three restarts. Because every emission is re-checked, this search needs no correctness guarantee of its own and acts purely as a proposer. If it fails, VETO falls back to the last admitted action, then the zero action, then the cache, and every emitted action passes a final check, which we motivate next. Every 20 episodes VETO refits τ by binary search on the most recent $W = 5000$ samples so that empirical coverage reaches $1 - \delta'$, and computes the Kolmogorov–Smirnov distance between consecutive windows to test its own assumption at runtime (lines 14–20).

5 SOUNDNESS OF THE ENVELOPE

This section establishes the first guarantee, the one that holds for any cost model. A monitor that edits an action stream is an enforcement mechanism, a component allowed to alter what passes through it, and such mechanisms are judged on two criteria (Schneider, 2000; Ligatti et al., 2005). We state both, because stating only the first is what makes the envelope look trivial. The monitor is a transition system whose states are the tuples of Definition 4, whose initial state has $\tau = 1$ and $C = \emptyset$, and whose step relation is one iteration of the inner loop of Algorithm 1. A state is reachable if some finite run produces it. One modeling choice does the heavy lifting. We treat $\hat{c}$ and σ as uninterpreted functions, so every claim in this section concerns the decision logic alone and none of it reasons about the network. Whether the bound is correct is the statistical question of Section 6.

Definition 6 (Soundness and transparency)**.** A monitor is *sound* if for every reachable monitor state m and every environment state s, the action it emits lies in $\mathrm{Adm}(m, s)$. It is *transparent* if it emits the proposed action unchanged whenever that action already lies in $\mathrm{Adm}(m, s)$.

Soundness is a property of the predicted bound, not of the true cost. A reader from control may hear "sound" as safe against the true cost, so we keep the two words apart, and whether the bound is correct is the statistical question of Section 6. The content is in the quantifier, since Adm is evaluated at the monitor state in force at emission, not at the state where the action was certified. Transparency holds immediately, since VETO alters an action only when the branch in line 6 fires; soundness is where the natural envelope fails.

Proposition 1 (The naive envelope is unsound)**.** *Without the re-check in line 9, there exist a reachable monitor state m, an environment state s, and ensemble outputs for which* VETO *emits an action outside* $\mathrm{Adm}(m, s)$.

Proof sketch. Two independent counterexamples exist (Appendix A). In the first, an action admitted and cached at $\tau = 1.0$, $\bar{\sigma} = 0.1$ becomes inadmissible after drift raises τ to 2.0 and $\bar{\sigma}$ to 0.2, because raising τ inflates the bound while raising $\bar{\sigma}$ lowers the threshold. In the second, no recalibration occurs, but the cached action was certified at some $s' \neq s$; since $\hat{c}$ and σ are arbitrary functions of the state, $\mathrm{ucb}(\tau, s', a)$ places no constraint on $\mathrm{ucb}(\tau, s, a)$. □

The second case is the stronger one, because no proof of soundness for the naive fallback can exist without assuming something about the cost model, which is exactly what an envelope should avoid assuming. Both failures live on the red path of Figure 2, the only place a certificate is reused, which is why a single re-check at emission suffices.

Lemma 2 (Monotone reuse)**.** *Let a be certified at environment state s under monitor state m_a, so that $\mathrm{ucb}(m_a.\tau, s, a) \leq d_t(m_a.\bar{\sigma})$. If a later reachable monitor state m satisfies $m.\tau \leq m_a.\tau$ and $m.\bar{\sigma} \leq m_a.\bar{\sigma}$, then $a \in \mathrm{Adm}(m, s)$.*

Proof sketch. $\tau \mapsto \mathrm{ucb}(\tau, s, a)$ is nondecreasing because $\beta > 0$ and $\sigma \geq 0$, and $\bar{\sigma} \mapsto d_t(\bar{\sigma})$ is nonincreasing because $\alpha > 0$; chaining the two with the certification hypothesis gives the claim (Appendix A). □

Storing $(\tau_a, \bar{\sigma}_a)$ alongside each cached action therefore yields a sufficient condition for safe reuse whenever calibration has only loosened. The lemma removes the temporal obstacle of Proposition 1 but not the spatial one, since it holds only at the certification state while the fallback paths reuse actions across states. The repair is to evaluate the bound at the current monitor and environment state immediately before emission and to refuse when nothing passes, which is line 9 of Algorithm 1 and costs one ensemble evaluation on the small fraction of steps that reach a fallback.

Theorem 3 (Envelope soundness)**.** *With the re-check at emission,* VETO *is sound. For every reachable monitor state m, every environment state s, and every $\hat{c}$ and σ, the emitted action lies in* $\mathrm{Adm}(m, s)$.

Proof sketch. Induct on reachable monitor states and case on the emission path. Paths 1 and 2 satisfy the bound by their own branch conditions, evaluated at the current m and s; paths 3 and 4 pass through the re-check, which admits an action only if $\mathrm{ucb}(m.\tau, s, a) \leq d_t(m.\bar{\sigma})$ and otherwise refuses. The argument never inspects $\hat{c}$ or σ, so it holds for arbitrary cost models (Appendix A). □

Corollary 4 (Safety and progress)**.** VETO *is sound unconditionally and emits an action whenever at least one of the actions it examines, namely the proposal $\pi(s)$, its projection, the last admitted action L, the zero action, or a cached action in C, lies in* $\mathrm{Adm}(m, s)$. *When none of them does it refuses and raises a safety exception rather than emitting an uncertified action.*

A refusal is a signal, not a crash. In our experiments it hands control to a fixed safe policy and logs the state for review (Appendix F).

Assumption 1 (Safe actions exist)**.** For all reachable s, there exists a with $c(s, a) \leq d$.

Assumption 1 is not tested but observed, through refusals. Under it and adequate calibration refusal does not occur; Section 7 reports how often the exception of Corollary 4 was raised in our runs.

Mechanized in Rocq. We have checked the results of this section in the Rocq prover (The Rocq Development Team, 2025). Appendix B maps each statement to its lemma and records what the development does not cover. Because the development declares $\hat{c}$ and σ as uninterpreted functions, the proof checker enforces cost-model independence rather than our own discipline, and soundness survives retraining, architecture changes, and calibration error. Theorem 3 is checked for every monitor state rather than for reachable ones only, so the induction disappears and the argument reduces to a case split on the four emission paths. The formalization also corrected Corollary 4, whose earlier form claimed emission whenever $\mathrm{Adm}(m, s) \neq \emptyset$, which is not provable for Algorithm 1 as written because the fallback searches a finite candidate list rather than $\mathcal{A}$.

6 CALIBRATION TRANSFER UNDER SHIFT

This section establishes the second guarantee, that the bound covers the true cost, under an assumption VETO tests online.

Assumption 2 (Bounded score drift)**.** For consecutive calibration windows, $\mathrm{KS}(P^e_t, P^e_{t-W}) \le \epsilon_{\mathrm{drift}}$.

This is weaker than bounding the shift in the state–action distribution, because it constrains only the scalar e, evaluated at the fitted τ that stays in force until the next refit. It is also testable, and VETO computes it every 20 episodes and alerts when it exceeds 0.15. The test is retrospective, since a window's drift is known at its end, so the operating rule of Section 7 acts on the following window, and under the refit schedule of Algorithm 1 consecutive-window drift is the calibration-to-deployment drift the bound needs.

Proposition 5 (Threshold events)**.** *For distributions P, Q on $\mathbb{R}$ and any $t \in \mathbb{R}$, $|P(Z \le t) - Q(Z \le t)| \le \mathrm{KS}(P, Q)$.*

Proof. By definition $\mathrm{KS}(P, Q) = \sup_x |F_P(x) - F_Q(x)|$ (Massey, 1951). Taking $x = t$ gives the claim. □

Theorem 6 (Coverage transfer)**.** *Under Assumption 2, if the scores of the calibration window are independent draws from P^e_{cal} and calibration attains empirical coverage $1 - \delta'$ on that window of size W, then with probability at least $1 - \delta''$ over the window,*

$$\mathrm{Cov}_{\mathrm{deploy}} \;\ge\; 1 - \delta' - \epsilon_{\mathrm{drift}} - \sqrt{\frac{\ln(2/\delta'')}{2W}}.$$

Proof sketch. The empirical cumulative distribution function concentrates by the Dvoretzky–Kiefer–Wolfowitz inequality, a uniform bound on the gap between an empirical and a true distribution function, with the tight constant (Dvoretzky et al., 1956; Massart, 1990), contributing the square-root term. Proposition 5 with Assumption 2 bounds the change across windows by $\epsilon_{\mathrm{drift}}$. The triangle inequality combines them (Appendix A), and this deterministic step is machine-checked as well (Appendix B). □

Summing over steps gives a worst-case violation bound of $TH\,(\delta' + \epsilon_{\mathrm{drift}} + O(1/\sqrt{W}))$. With $\delta' = 0.02$, $\epsilon_{\mathrm{drift}} = 0.05$, and $TH = 20{,}000$, one no-shift cell of the pilot, the first two terms permit 1,400 violations, the concentration term adds 2.3 points at $W = 5000$ and $\delta'' = 0.01$. Figure 6 tests both statements where the deployment law is under our control, on synthetic scores with heavy-tailed residuals (Appendix D). Across 2,400 trials with windows from 500 to 10,000 scores and location, scale, and rare-event shifts, the bound of Theorem 6 was never crossed against a nominal allowance of 5%. The loss at the threshold stayed below the measured drift in every trial, as Proposition 5 requires. The bound is tight where it matters. For rare-event shifts, the ones that move mass across the safety threshold, the median loss is 93% of the drift, while for location shifts it is 17%, since the supremum in the Kolmogorov–Smirnov distance is then attained far from the threshold. In panel (b) the gap between the observed surface and the bound is the finite-sample term, six points at $W = 500$ and under two at $W = 10{,}000$. Adaptive conformal inference recomputes its quantile from an $O(W)$ window at every step (Gibbs & Candès, 2021) and needs no drift assumption; VETO pays Assumption 2, which it tests.

7 EXPERIMENTS

The introduction made three testable claims. The envelope lowers the per-step violations of a trained constrained-optimization policy under shift, the naive fallback of Proposition 1 emits inadmissible actions in practice while VETO does not, and refusal is a signal rather than safety when no admissible action exists.

Table 1: Per-step true violations without shift for each base policy alone and under the three envelopes, with VETO's intervention and refusal rates and the reward change over 5 seeds, the lowest rate in blue and the highest in red.

Environment	Policy	True violations (%)				VETO (% of steps)		Reward
		alone	filter	naive	VETO	interv.	refuse	alone → VETO
PointGoal	PPO-LAG	6.02	0.97	0.48	**0.03**	99	99	2.44 → 0.01
	CPO	2.66	0.89	0.48	**0.20**	97	97	1.16 → 0.05
	PID-LAG	1.93	1.40	1.15	**0.57**	60	60	0.04 → -0.02
CarGoal	PPO-LAG	4.84	3.52	3.95	**2.61**	43	41	4.11 → 2.25
	CPO	2.77	2.35	3.07	**1.92**	32	30	2.58 → 1.61
	PID-LAG	2.14	**1.66**	2.19	1.77	22	21	0.48 → 0.34

Setup. We evaluate on Safety-Gymnasium (Ji et al., 2023) PointGoal1 and CarGoal1, whose per-step cost is 1 when the agent touches a hazard and 0 otherwise, so the threshold $d = 0.5$ makes any hazard contact a violation. Base policies are PPO-Lagrangian (Ray et al., 2019), CPO (Achiam et al., 2017), and PID-Lagrangian (Stooke et al., 2020) as implemented in OmniSafe (Ji et al., 2024), each trained for 500k environment steps at the episodic cost limit 25 with 5 seeds, 30 policies in all. Each policy is evaluated alone and under three envelopes that share one ensemble, VETO as in Algorithm 1, the naive fallback of Proposition 1 as a control, and a conformal filter that keeps the quantile of adaptive conformal inference (Gibbs & Candès, 2021) without a re-check. The ensemble has $K = 5$ members trained on 50,000 steps of the base policy, and the envelope uses $\beta = 1$, $\alpha = 0.5$, $\delta' = 0.02$ and $W = 5000$ (Appendix E). Three shifts are applied at evaluation time only, a dynamics shift that scales gravity, friction and mass, the hidden-parameter setting of Kwon et al. (2026), additive observation noise, the observational-perturbation setting of Liu et al. (2023), and rare-event impulses, each at magnitudes 0.5, 1.0 and 2.0. The no-shift and magnitude-1.0 cells run 100 episodes of horizon 200 per seed and the other cells 50, so a cell holds 10,000 to 20,000 decisions per seed and the VETO cells hold 4200K decisions in all. We report the per-step true violation rate with its standard error over seeds, the intervention and refusal rates, and the reward per episode. A refusal hands control to the zero action.

What the envelope does to a trained policy. Table 1 reports the no-shift cells. VETO lowers the violation rate of every policy on both tasks, on PointGoal from 6.02% to 0.03% for PPO-Lagrangian and from 2.66% to 0.20% for CPO, on CarGoal from 4.84% to 2.61% and from 2.77% to 1.92%. On PointGoal the envelope refuses on 99.4% of PPO-Lagrangian's steps and 96.9% of CPO's, so the agent mostly stands still and the reward falls from 2.44 to 0.01. On CarGoal refusals are 40.9% and 29.6%. The recalibration logs show why, since the fitted temperatures on PointGoal stay below 6.49 while the ensemble's spread on a binary cost is large, so the margin $\alpha\bar{\sigma}$ consumes most of d. Table 1 also shows that the intervention and refusal columns nearly coincide, so almost every intervention is a refusal, and a refusal is the zero action. The envelope therefore has no way to act on a rejected step except to stop, which is a property of its fallback, not of its guarantee, and Section 8 returns to it. The conformal filter shares the ensemble and lacks only the re-check, so it isolates that component, and its violation rate is at or above VETO's in 54 of 60 cells. The two tasks therefore show two regimes of one envelope. PointGoal is the conservative regime, where the margin dominates and the agent mostly stops, and CarGoal is the working regime, where refusals stay near a third of steps and violations still fall. The dial between them is α, fixed a priori at 0.5 here.

The naive fallback in practice. When projection fails, the naive fallback emits the last admitted action without the re-check, and the harness records whether it was admissible. It was not on 28.5% of steps on average and on up to 75.2% in the worst cell. The naive fallback's violation rate is at or above VETO's in 58 of 60 cells, and the two coincide only where both refuse every step. This is Proposition 1 observed, and Theorem 3 in the same logs, since VETO emitted no inadmissible action in 4200K decisions.

Behaviour as the shift grows. Under the dynamics shift of Figure 3, VETO keeps the lowest violation rate at every magnitude on both tasks and beats every policy alone in all 18 dynamics cells, while the policies alone stay between 3 and 4%. The reward panels show the price, near zero on PointGoal and half to two thirds of the policy's reward on CarGoal (per cell in Table 3, Appendix C). Rare-event impulses are the case the envelope cannot handle, and the logs say so. At magnitude

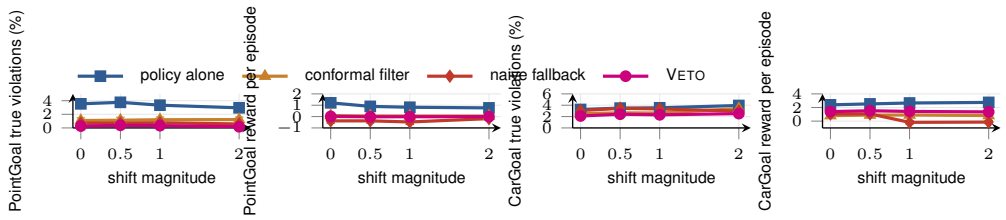


Figure 3: True violations and reward per episode against the dynamics shift magnitude, averaged over three base policies and 5 seeds.

2.0 on PointGoal VETO refuses on 100.0% of steps and still violates on 4.65% of them, against 3.27% for CPO alone. An impulse moves the agent into a hazard whatever action is emitted, which is Assumption 1 failing and refusal being a signal rather than safety. The drift monitor saw these shifts, with the Kolmogorov–Smirnov distance above the alert threshold in 41% of the 960 refits, so the guarantee of Theorem 6 is declared void exactly where the violations appear. Realized coverage before each refit stays between 96.2% and 99.8% at magnitude 2.0 across shifts and tasks, against the nominal 98%, within the allowance of Theorem 6 at $W = 5000$ (Figure 4, Appendix C).

Cost. On one core of an Intel Xeon at 2.10 GHz a NumPy reference implementation of Algorithm 1 adds 0.14 ms on a step the guard passes and 3.2 ms on a step that projects, inside a 10 ms control period, and the PyTorch envelope used here is slower (Appendix E).

8 DISCUSSION

Use and price. VETO suits settings where a single violation is unacceptable and shift is expected, at the price in Table 1. Its threshold $d_t = d - \alpha\bar{\sigma}$ makes it deliberately not transparent (Definition 6), and α is to be chosen once on a validation shift and frozen (Appendix E). The PointGoal refusals point at a second dial, the fallback itself. A rejected step need not stop the agent, since the policy's own distribution supplies candidates, and emitting the most probable sample that passes the re-check keeps Theorem 3 intact. The released envelope implements this sampled fallback, which we did not evaluate in time for this version.

Soundness does not imply feasibility. Theorem 3 says that every emitted action is admissible now, not that admissible actions keep existing, so the filtered policy can reach a state where Corollary 4 forces a refusal. Predictive safety filters and control barrier functions rule this out with a control invariant set from a dynamics model (Wabersich & Zeilinger, 2023; Ames et al., 2019), which VETO lacks by design, so recursive feasibility rests on Assumption 1.

Limitations. Our evaluation is in simulation, and transfer to hardware introduces shift we have not measured. The evaluation is a pilot, with two Safety-Gymnasium tasks, base policies trained for 500k steps, three shift types, and α fixed a priori rather than chosen on a validation shift. The within-window independence that Theorem 6 assumes is a modeling assumption in a Markov environment that we leave open.

9 CONCLUSION

VETO keeps two guarantees deliberately separate, soundness for any cost model and statistical constraint satisfaction under an assumption it tests online. In a pilot on two Safety-Gymnasium tasks it lowered the per-step violation rate in 58 of 60 cells while the naive fallback emitted inadmissible actions on up to 75.2% of steps. A safety layer should say which of its guarantees survive when its predictions do not.

REPRODUCIBILITY STATEMENT

Every result can be checked from the paper and the supplementary material. Appendix A gives the full proofs, and Appendix B describes the Rocq development that checks Section 5, with a table from

each statement to its lemma and a list of what the development does not cover. Appendix E lists the shift protocol, the ensemble and its optimizer, the calibration schedule and window, the baseline tuning procedure, and the episode and horizon accounting behind every count. Appendix C holds the extended results and Appendix D the synthetic experiments. The anonymized supplementary material contains the code, the shift wrappers, the aggregated tables behind Table 1 and Appendix C with the scripts that regenerate them from the per-step logs, the scripts and data behind Appendix D, the Rocq development, and the sources of every figure and number macro in this paper. The per-step logs themselves (690 MB) are released with the code.

Ethics statement

VETO is a safety mechanism evaluated in simulation. The paper uses no human subjects and no private data. Two points matter for anyone who deploys it. The soundness guarantee is about the predicted bound, not the true cost, so it must not be read as a guarantee of true safety. Section 6 states the assumption that true safety needs. A refusal must be bound to a fixed safe fallback, because an unhandled refusal is itself a failure. VETO alters an action only when the guard of line 6 fires and logs every alteration, so its behaviour can be audited. We see no dual-use concern beyond those of safe control in general.

AI use statement

We used a large language model as an assistant for drafting and editing the text, for drafting the TikZ and PGFPlots figure code, for writing the evaluation harness, the shift wrappers, the envelope implementation from our algorithm and specifications, the table and figure generators, and the synthetic experiments of Appendix D, for running the training and evaluation runs of Section 7 under our direction on our own machine, and for finding references and checking them against publisher records. The research idea, the design of the envelope, the theorems and their proofs in Appendix A, and the Rocq development of Section 5 are the authors' own. Every reported number is recomputed by the released scripts from the released logs, and the authors reviewed all AI-assisted text and code. We take full responsibility for the content of this paper.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31, 2017.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 2669–2678, 2018. doi: 10.1609/aaai.v32i1.11797.

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

Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8): 3861–3876, 2017. doi: 10.1109/TAC.2016.2638961.

Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *Proceedings of the 18th European Control Conference (ECC)*, pp. 3420–3431, 2019. doi: 10.23919/ECC.2019.8796030.

Abhishek Anand, Andrew Appel, Greg Morrisett, Zoe Paraskevopoulou, Randy Pollack, Olivier Savary Bélanger, Matthieu Sozeau, and Matthew Weaver. CertiCoq: A verified compiler for Coq. In *The Third International Workshop on Coq for Programming Languages (CoqPL)*, 2017.

Yarden As, Ilnura Usmanova, Sebastian Curi, and Andreas Krause. Constrained policy optimization via Bayesian world models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=PRZoSmCinhf.

Qinbo Bai, Amrit Singh Bedi, Mridul Agarwal, Alec Koppel, and Vaneet Aggarwal. Achieving zero constraint violation for constrained reinforcement learning via primal-dual approach. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, volume 36, pp. 3682–3689, 2022. doi: 10.1609/aaai.v36i4.20281.

Steven Carr, Nils Jansen, Sebastian Junges, and Ufuk Topcu. Safe reinforcement learning via shielding under partial observability. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, volume 37, pp. 14748–14756, 2023. doi: 10.1609/aaai.v37i12.26723.

Weiqin Chen and Santiago Paternain. Generalized constraint for probabilistic safe reinforcement learning. In *Proceedings of the 6th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 242 of *Proceedings of Machine Learning Research*, pp. 1606–1618, 2024. URL https://proceedings.mlr.press/v242/chen24b.html.

Richard Cheng, Gábor Orosz, Richard M. Murray, and Joel W. Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 3387–3395, 2019. doi: 10.1609/aaai.v33i01.33013387.

Koen Claessen and John Hughes. QuickCheck: A lightweight tool for random testing of Haskell programs. In *Proceedings of the 5th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, pp. 268–279. ACM, 2000. doi: 10.1145/351240.351266.

Matthew Cleaveland, Insup Lee, George J. Pappas, and Lars Lindemann. Conformal prediction regions for time series using linear complementarity programming. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, pp. 20984–20992, 2024.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, 27(3):642–669, 1956. doi: 10.1214/aoms/1177728174.

Nathan Fulton and André Platzer. Safe reinforcement learning via formal methods: Toward safe control through proof and learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 6485–6492, 2018. doi: 10.1609/aaai.v32i1.12107.

Nathan Fulton and André Platzer. Verifiably safe off-model reinforcement learning. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 11427 of *Lecture Notes in Computer Science*, pp. 413–430. Springer, 2019. doi: 10.1007/978-3-030-17462-0_28.

Shiqing Gao, Jiaxin Ding, Luoyi Fu, and Xinbing Wang. Controlling underestimation bias in constrained reinforcement learning for safe exploration. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18290–18316, 2025.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Isaac Gibbs and Emmanuel J. Candès. Adaptive conformal inference under distribution shift. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pp. 1660–1672, 2021.

Edwin Hamel-De le Court, Francesco Belardinelli, and Alexander W. Goodall. Probabilistic shielding for safe reinforcement learning. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, volume 39, pp. 16091–16099, 2025. doi: 10.1609/aaai.v39i15.33767.

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

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Bettina Könighofer, Florian Lorber, Nils Jansen, and Roderick Bloem. Shield synthesis for reinforcement learning. In *Leveraging Applications of Formal Methods, Verification and Validation (ISoLA)*, volume 12476 of *Lecture Notes in Computer Science*, pp. 290–306. Springer, 2020. doi: 10.1007/978-3-030-61362-4_16.

Minjae Kwon, Tyler Ingebrand, Ufuk Topcu, and Lu Feng. Adaptive shielding for safe reinforcement learning under hidden-parameter dynamics shifts. arXiv preprint arXiv:2506.11033, version 2, 2026.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 2017.

Jordan Lekeufack, Anastasios N. Angelopoulos, Andrea Bajcsy, Michael I. Jordan, and Jitendra Malik. Conformal decision theory: Safe autonomous decisions from imperfect predictions. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11668–11675, 2024.

Pierre Letouzey. A new extraction for Coq. In *Types for Proofs and Programs (TYPES 2002), Selected Papers*, volume 2646 of *Lecture Notes in Computer Science*, pp. 200–219. Springer, 2003. doi: 10.1007/3-540-39185-1_12.

Edouard Leurent. An environment for autonomous driving decision-making. GitHub repository, https://github.com/eleurent/highway-env, 2018.

Jay Ligatti, Lujo Bauer, and David Walker. Edit automata: Enforcement mechanisms for run-time security policies. *International Journal of Information Security*, 4(1–2):2–16, 2005. doi: 10.1007/s10207-004-0046-8.

Lars Lindemann, Xin Qin, Jyotirmoy V. Deshmukh, and George J. Pappas. Conformal prediction for STL runtime verification. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 142–153, 2023.

Zuxin Liu, Zijian Guo, Zhepeng Cen, Huan Zhang, Jie Tan, Bo Li, and Ding Zhao. On the robustness of safe reinforcement learning under observational perturbations. In *International Conference on Learning Representations (ICLR)*, 2023.

Pascal Massart. The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *The Annals of Probability*, 18(3):1269–1283, 1990. doi: 10.1214/aop/1176990746.

Frank J. Massey, Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951. doi: 10.2307/2280095.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. Technical report, OpenAI, 2019.

Fred B. Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1):30–50, 2000. doi: 10.1145/353323.353382.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID Lagrangian methods. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9133–9143, 2020.

Kegan J. Strawn, Nora Ayanian, and Lars Lindemann. Conformal predictive safety filter for RL controllers in dynamic environments. *IEEE Robotics and Automation Letters*, 8(11):7833–7840, 2023.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery RL: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3): 4915–4922, 2021. doi: 10.1109/LRA.2021.3070252.

The Rocq Development Team. The Rocq prover. Version 9.1, 2025.

Koundinya Vajjha, Avraham Shinnar, Barry Trager, Vasily Pestun, and Nathan Fulton. CertRL: Formalizing convergence proofs for value and policy iteration in Coq. In *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP)*, pp. 18–31, 2021. doi: 10.1145/3437992.3439927.

Kim P. Wabersich and Melanie N. Zeilinger. Predictive control barrier functions: Enhanced safety mechanisms for learning-based control. *IEEE Transactions on Automatic Control*, 68(5):2638–2651, 2023. doi: 10.1109/TAC.2022.3175628.

Nuoya Xiong, Yihan Du, and Longbo Huang. Provably safe reinforcement learning with step-wise violation constraints. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.

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

Yiqi Zhao, Bardh Hoxha, Georgios Fainekos, Jyotirmoy V. Deshmukh, and Lars Lindemann. Robust conformal prediction for STL runtime verification under distribution shift. In *Proceedings of the ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 169–179, 2024. arXiv:2311.09482.

He Zhu, Zikang Xiong, Stephen Magill, and Suresh Jagannathan. An inductive synthesis framework for verifiable reinforcement learning. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pp. 686–701, 2019. doi: 10.1145/3314221.3314638.

A Full Proofs

A.1 Proposition 1 (the naive envelope is unsound)

Two independent counterexamples exist.

First counterexample (stale calibration). Take $\beta = 3$, $\alpha = 0.5$, $d = 1.0$, and a state where $\hat{c} = 0.5$ and $\sigma = 0.1$. At certification time $\tau = 1.0$ and $\bar{\sigma} = 0.1$, so $\mathrm{ucb} = 0.5 + 3 \cdot 1.0 \cdot 0.1 = 0.8$ and $d_t = 1.0 - 0.5 \cdot 0.1 = 0.95$, and the action is admitted and cached. Drift then raises τ to 2.0 and $\bar{\sigma}$ to 0.2, giving $\mathrm{ucb} = 0.5 + 3 \cdot 2.0 \cdot 0.1 = 1.1$ and $d_t = 1.0 - 0.5 \cdot 0.2 = 0.9$. The cached action is now inadmissible, because raising τ inflates the bound while raising $\bar{\sigma}$ lowers the threshold. The naive fallback nevertheless emits it when the proposal and its projection both fail the guard. In the Rocq development this is `naive_unsound_temporal`; the companion statement `temporal_certificate_was_valid` confirms that the certificate was genuinely admissible when it was issued, and the module `Witness` instantiates a one-state, one-action system with $\hat{c} = 1/2$ and $\sigma = 1/10$ to show that the hypotheses are satisfiable, so the statement is not vacuous.

Second counterexample (state mismatch). Fix τ and $\bar{\sigma}$ so that no recalibration occurs. The cached action was certified at some state $s' \neq s$, so $\mathrm{ucb}(\tau, s', a) \leq d_t(\bar{\sigma})$. Since $\hat{c}$ and σ are arbitrary functions of the state, this places no constraint on $\mathrm{ucb}(\tau, s, a)$; in particular any cost model with $\mathrm{ucb}(\tau, s, a) > d_t(\bar{\sigma})$ is consistent with the certificate, and under such a model the naive fallback emits a outside $\mathrm{Adm}(m, s)$. No numbers are needed, and no proof of soundness for the naive fallback can exist without assuming something about the cost model. In the Rocq development this is `naive_unsound_spatial`. □

A.2 LEMMA 2 (MONOTONE REUSE)

Since $\beta > 0$ and $\sigma \geq 0$, the map $\tau \mapsto \mathrm{ucb}(\tau, s, a) = \hat{c}(s, a) + \beta\tau\sigma(s, a)$ is nondecreasing, so $m.\tau \leq m_a.\tau$ gives $\mathrm{ucb}(m.\tau, s, a) \leq \mathrm{ucb}(m_a.\tau, s, a)$. Since $\alpha > 0$, the map $\bar{\sigma} \mapsto d_t(\bar{\sigma}) = d - \alpha\bar{\sigma}$ is nonincreasing, so $m.\bar{\sigma} \leq m_a.\bar{\sigma}$ gives $d_t(m_a.\bar{\sigma}) \leq d_t(m.\bar{\sigma})$. Chaining the two with the certification hypothesis $\mathrm{ucb}(m_a.\tau, s, a) \leq d_t(m_a.\bar{\sigma})$ gives $\mathrm{ucb}(m.\tau, s, a) \leq d_t(m.\bar{\sigma})$, that is, $a \in \mathrm{Adm}(m, s)$. Note the quantification, since the conclusion is at the same environment state s at which the certificate was issued, which is exactly why the lemma removes the temporal obstacle and leaves the spatial one untouched. In the Rocq development this is `monotone_reuse`, built from `ucb_mono_tau` and `dthr_antitone`. □

A.3 THEOREM 3 (ENVELOPE SOUNDNESS)

We show that for every reachable monitor state m, every environment state s, and every pair of functions $\hat{c}, \sigma$, the emitted action lies in $\mathrm{Adm}(m, s)$. Induct on the length of the run producing m; the inductive hypothesis is not needed for the emission itself, only to know that m is the state in force at emission. Case on the emission path of Algorithm 1.

Path 1 (guard passes). The branch condition in line 6 is false, so $\mathrm{ucb}(m.\tau, s, \pi(s)) \leq d_t(m.\bar{\sigma})$, evaluated at the current m and s; the emitted action $\pi(s)$ lies in $\mathrm{Adm}(m, s)$.

Path 2 (projection succeeds). The projected action a^* satisfies the re-check in line 9, that is, $\mathrm{ucb}(m.\tau, s, a^*) \leq d_t(m.\bar{\sigma})$ at the current m and s; hence $a^* \in \mathrm{Adm}(m, s)$. Nothing about the correctness of PROJECT is used.

Paths 3 and 4 (fallback to L, the zero action, or the cache). FALLBACK scans the finite candidate list and returns the first candidate that passes $\mathrm{ucb}(m.\tau, s, a) \leq d_t(m.\bar{\sigma})$ at the current m and s, and refuses if none passes. Any emitted action therefore lies in $\mathrm{Adm}(m, s)$, regardless of the state or calibration under which it was originally certified.

In every case the emitted action lies in $\mathrm{Adm}(m, s)$. The argument never inspects $\hat{c}$ or σ, so it holds for arbitrary cost models. The proof is over the reals; a floating-point implementation is faithful up to the rounding of the single comparison in the re-check. In the Rocq development this is `emit_sound`; because every path is guarded by the decidable test `admb` at the current (m, s), the mechanized statement quantifies over all monitor states, reachable or not, which is strictly stronger and makes the induction unnecessary (`emit_sound_all_paths`). Transparency is `emit_transparent`. □

A.4 COROLLARY 4 (SAFETY AND PROGRESS)

Soundness is Theorem 3. For progress, suppose some candidate a among $\pi(s)$, its projection, L, the zero action, and the cache lies in $\mathrm{Adm}(m, s)$. If $\pi(s)$ or its projection passes, paths 1 or 2 emit it. Otherwise FALLBACK scans a list containing a, and since a passes the re-check, the scan returns some candidate that passes (the first such), which lies in $\mathrm{Adm}(m, s)$. If no candidate passes, FALLBACK returns nothing and the envelope refuses. In the Rocq development the two clauses are `emit_progress` (with `first_adm_complete`) and `emit_refuses_or_admissible`. The earlier form of this corollary, emission whenever $\mathrm{Adm}(m, s) \neq \emptyset$, is not provable for Algorithm 1 as written, because the fallback never searches $\mathcal{A}$ beyond its candidate list. □

A.5 PROPOSITION 5 (THRESHOLD EVENTS)

By definition $\mathrm{KS}(P, Q) = \sup_{x \in \mathbb{R}} |F_P(x) - F_Q(x)|$ (Massey, 1951), where $F_P(x) = P(Z \leq x)$. Taking $x = t$ gives $|P(Z \leq t) - Q(Z \leq t)| = |F_P(t) - F_Q(t)| \leq \mathrm{KS}(P, Q)$. □

A.6 THEOREM 6 (COVERAGE TRANSFER)

Let P^e_{cal} denote the distribution of the score e over the calibration window and P^e_{deploy} its distribution over the deployment window that follows, with F_{cal} and F_{deploy} the corresponding cumulative distribution functions and $\hat{F}_{\mathrm{cal}}$ the empirical one over the W calibration samples. Coverage is $F(0)$, namely $\mathrm{Cov}_{\mathrm{deploy}} = F_{\mathrm{deploy}}(0)$ and the calibration step enforces $\hat{F}_{\mathrm{cal}}(0) \geq 1 - \delta'$. By the Dvoretzky–Kiefer–Wolfowitz inequality with the tight constant (Dvoretzky et al., 1956; Massart,

1990), which requires the W scores to be independent draws from P^e_{cal}, with probability at least $1 - \delta''$,

$$\sup_x |\hat{F}_{\text{cal}}(x) - F_{\text{cal}}(x)| \le \sqrt{\ln(2/\delta'')/(2W)}.$$

By Proposition 5 and Assumption 2, $|F_{\text{deploy}}(0) - F_{\text{cal}}(0)| \le \text{KS}(P^e_{\text{deploy}}, P^e_{\text{cal}}) \le \epsilon_{\text{drift}}$. The triangle inequality gives

$$F_{\text{deploy}}(0) \ \ge \ \hat{F}_{\text{cal}}(0) - \sqrt{\ln(2/\delta'')/(2W)} - \epsilon_{\text{drift}} \ \ge \ 1 - \delta' - \epsilon_{\text{drift}} - \sqrt{\ln(2/\delta'')/(2W)}.$$

Summing the per-step failure probability over TH steps gives the worst-case violation ceiling $TH\,(\delta' + \epsilon_{\text{drift}} + O(1/\sqrt{W}))$ quoted in Section 6; the ceiling treats steps as independent and is therefore loose. □

B THE ROCQ DEVELOPMENT

The supplementary file `VetoProof.v` formalizes Section 5. The cost model is a pair of uninterpreted functions `chat` and `spread`; no lemma inspects them, which is the formal content of the claim that soundness holds for an arbitrary cost model. The base policy and the projection proposer are arbitrary functions as well, matching the "untrusted proposer" status of line 7. Table 2 gives the correspondence. The hypotheses `beta_pos`, `alpha_pos`, and `spread_nonneg` are used only by `monotone_reuse`; `emit_sound` does not need them. `Print Assumptions emit_sound` reports only the classical axioms of the standard-library reals. The development also checks the deterministic step of Theorem 6. That step goes from a bound on the gap between two distribution functions to a bound on the coverage of the threshold event, and then combines it with the empirical coverage and the Dvoretzky–Kiefer–Wolfowitz gap. The probabilistic event itself, that the empirical distribution lies within that gap, is a hypothesis of the lemma and is not derived. The file compiles under Coq 8.18 with the standard-library prefix of Rocq 9 mapped to Coq's, and `Print Assumptions` lists only the classical reals and functional extensionality of the standard library for every lemma.

Table 2: Correspondence between the paper's statements, the Rocq lemmas that check them, and the hypotheses or caveats each carries, numbered as in the paper.

Paper	Rocq name	Note
Definition 2 (ucb)	`ucb`	calibrated bound
Definition 5 (d_t, Adm)	`dthr`, `Adm`, `admb`	`admb` is the decidable test the monitor runs
Definition 4	`MState`	episode counter k omitted; soundness does not depend on it
Algorithm 1, lines 4–12	`step1`, `emit`	`naive_emit` drops the re-check of line 9
Lemma 2	`monotone_reuse`	via `ucb_mono_tau`, `dthr_antitone`
Proposition 1, case 1	`naive_unsound_temporal`	with `temporal_certificate_was_valid`, module `Witness`
Proposition 1, case 2	`naive_unsound_spatial`	no numeric hypotheses
Theorem 3	`emit_sound`, `emit_sound_all_paths`	over all monitor states, not only reachable ones
Definition 6, transparency	`emit_transparent`	
Corollary 4	`emit_progress`, `emit_refuses_or_admissible`	finite candidate list; see the caveat in the file
Repair	`repair_blocks_stale`	the repaired monitor never emits the stale action
Equation equation 1, Proposition 5	`threshold_coverage_transfer`	coverage is the distribution function at 0; any pointwise bound on the gap bounds it
Theorem 6, deterministic step	`coverage_transfer_arith`	the DKW event is a hypothesis; only the arithmetic is checked

What the mechanization does not cover. Because the monitor is a small stateful program over an uninterpreted cost model, it is a suitable target for a proof assistant, and three limits should be read alongside the development. First, the object verified is the functional implementation of the decision logic and not the implementation we experimented with. The verified implementation can be extracted to executable OCaml (Letouzey, 2003), or compiled with CertiCoq (Anand et al., 2017), which guarantees the soundness of its compilation to C. Second, the arithmetic is over the reals, while the implementation uses floating point, so a faithful account must still relate the two. Third, the development covers the decision logic and the deterministic arithmetic of the coverage transfer, and it says nothing about the true cost, about the probabilistic Dvoretzky–Kiefer–Wolfowitz event, or about the recalibration and drift tests (lines 14–20 of Algorithm 1), which remain the statistical claims of Section 6. A machine-checked soundness proof therefore raises confidence in the envelope logic alone, and it is not evidence that the whole deployment is safe.

C Additional Results

Table 4 gives the per-step true violation rate of every base policy alone and under the three envelopes in every evaluation cell, averaged over 5 seeds. Per-seed values, intervention and refusal rates, rewards and per-decision times for every cell are in `tables/seeds.csv` of the supplementary material. The recalibration logs record τ, the empirical coverage before each refit, and the drift statistic, whose maximum reached 0.75 on PointGoal and 0.51 on CarGoal. Table 3 gives VETO's intervention and refusal rates and the reward change against the policy alone for the no-shift and dynamics cells, the price behind the rates of Table 4. Figure 4 reports the realized coverage of the bound on the calibration window immediately before each refit, averaged over policies, seeds and refits, against the shift magnitude. On PointGoal it stays above the nominal 0.98 everywhere. On CarGoal it sits between 96.2% and 97.7% at magnitude 2.0, a shortfall of at most 1.8 points, which Theorem 6 allows at $W = 5000$, where the concentration term alone is 2.3 points. The lowest single-window coverage over all 960 refits was 91.3%.

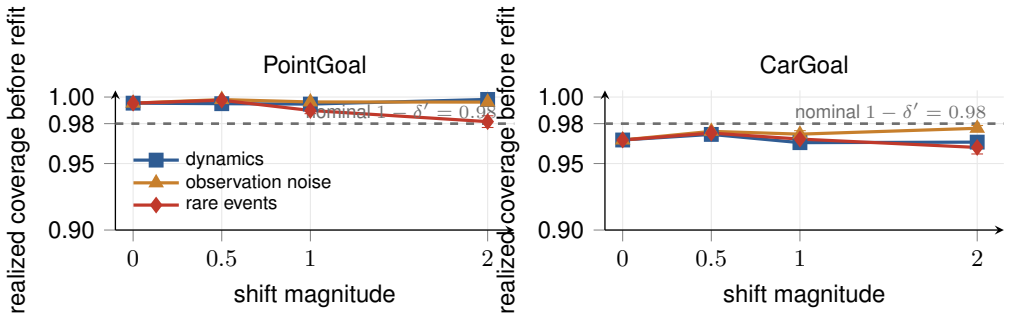


Figure 4: Realized coverage of the bound on the calibration window before each refit against shift magnitude, for three shift types on PointGoal and CarGoal, with the nominal level dashed.

Table 3: VETO's intervention and refusal rates (% of steps) and the reward per episode of the policy alone against VETO, over 5 seeds, for the no-shift and dynamics cells.

Env.	Policy	no shift			dynamics 1.0			dynamics 2.0		
		interv.	refuse	reward	interv.	refuse	reward	interv.	refuse	reward
PointGoal	PPO-Lag	99	99	2.44→0.01	100	100	2.01→0.01	100	100	1.81→0.00
	CPO	97	97	1.16→0.05	99	98	0.59→0.01	100	100	0.52→0.00
	PID-Lag	60	60	0.04→-0.02	71	70	-0.14→-0.06	74	74	-0.04→-0.02
CarGoal	PPO-Lag	43	41	4.11→2.25	50	48	4.61→2.24	57	52	5.02→2.12
	CPO	32	30	2.58→1.61	37	34	2.76→1.59	35	31	2.51→1.57
	PID-Lag	22	21	0.48→0.34	24	24	0.61→0.43	42	41	0.71→0.36

Table 4: Per-step true violations (%) in every evaluation cell over 5 seeds, with columns for the shift type and magnitude.

Env.	Policy	Mode	none	dyn 0.5	dyn 1.0	dyn 2.0	obs 0.5	obs 1.0	obs 2.0	rare 0.5	rare 1.0	rare 2.0
PointGoal	PPO-LAG	policy alone	6.02	5.11	4.20	3.64	5.38	5.35	4.23	5.86	5.99	6.64
		conformal filter	0.97	1.24	0.71	0.70	1.23	0.89	1.02	1.35	2.29	5.10
		naive fallback	0.48	0.22	0.42	0.00	0.35	0.53	0.29	0.45	1.82	4.65
		VETO	0.03	0.00	0.03	0.00	0.00	0.00	0.00	0.16	1.45	4.65
	CPO	policy alone	2.66	3.84	3.20	3.20	2.40	2.76	2.73	2.70	2.84	3.27
		conformal filter	0.89	0.55	0.79	1.12	1.23	1.13	0.72	0.79	2.00	4.75
		naive fallback	0.48	0.42	0.55	0.38	0.36	0.59	0.84	1.10	3.12	4.65
		VETO	0.20	0.49	0.23	0.15	0.24	0.15	0.23	0.43	1.85	4.65
	PID-LAG	policy alone	1.93	2.33	2.61	2.03	1.09	1.23	1.19	2.03	2.00	3.40
		conformal filter	1.40	1.61	2.11	1.85	0.81	0.94	1.16	1.59	2.57	4.38
		naive fallback	1.15	1.51	1.29	1.38	1.01	1.31	0.78	1.66	2.27	3.77
		VETO	0.57	0.67	0.74	0.38	0.38	0.57	0.74	0.57	1.79	4.32
CarGoal	PPO-LAG	policy alone	4.84	5.07	5.31	5.62	4.72	4.27	3.98	4.90	5.49	6.00
		conformal filter	3.52	3.62	3.35	5.21	3.45	2.64	2.58	3.54	3.02	4.25
		naive fallback	3.95	4.47	4.02	3.66	3.27	3.78	3.91	3.27	3.96	5.30
		VETO	2.61	3.03	3.07	3.39	2.69	2.45	2.57	2.53	2.99	5.16
	CPO	policy alone	2.77	2.89	2.76	3.29	2.84	2.28	1.37	2.43	2.66	2.67
		conformal filter	2.35	2.40	2.45	2.66	2.13	1.89	2.04	2.64	1.87	3.76
		naive fallback	3.07	2.97	3.02	2.64	2.40	2.61	1.53	3.11	3.54	3.52
		VETO	1.92	2.03	1.86	2.14	1.76	1.76	1.13	1.84	1.74	2.21
	PID-LAG	policy alone	2.14	2.51	2.56	2.95	1.42	1.38	1.38	1.90	2.11	2.56
		conformal filter	1.66	1.96	2.06	2.35	1.34	1.12	1.29	1.65	1.58	2.86
		naive fallback	2.19	2.89	3.00	2.57	1.32	1.61	1.26	2.17	2.02	2.23
		VETO	1.77	2.17	1.99	2.04	1.13	1.26	1.20	1.77	1.62	2.01

D SYNTHETIC VALIDATION OF THE THEORY

The experiments in this appendix involve no environment data; they test the three formal claims on inputs we control, and every number is regenerated by the scripts in `experiments/` of the supplementary material. Panel (a) of Figure 1 is one simulated episode of an autoregressive baseline with Bernoulli bursts, rescaled so that its mean is exactly $0.5d$, on which 11.3% of the 150 steps exceed d; panel (b) is exact, with $\Pr[c > d]$ computed in closed form for two-point, lognormal, and gamma cost laws whose mean is pinned at $0.5d$, and it reaches 49.99% for two-point bursts, 12.0% for the lognormal family, and 15.9% for the gamma family, all consistent with the Markov bound $\mathbb{E}[c]/d = 0.5$, which is the only bound an average constraint implies.

Coverage transfer (Figure 6). The true cost is $c = \hat{c} + \sigma z$ with a Student-$t(4)$ residual z and a spread σ drawn uniformly from $[0.1, 0.4]$, so the score of Definition 3 is $e = \sigma(z - \beta\tau)$ with $\beta = 3$ and the coverage event is $z \le \beta\tau$. Each trial draws a calibration window of $W \in \{500, 1000, 2000, 5000, 10000\}$ scores, fits τ by the binary search of line 15 to empirical coverage $1 - \delta' = 0.98$, shifts the residual law (location $z + \mu$, scale $(1 + \mu)z$, or a rare-event tail that adds a gamma excursion with probability $\mu/4$), measures the Kolmogorov–Smirnov distance ϵ_{KS} between the calibration and deployment score distributions at the fitted τ on 60,000 draws each, and measures the deployment coverage on the same draws, so that the loss $(1 - \delta') - P_{\mathrm{dep}}(e \le 0)$ and ϵ_{KS} are computed on exactly the quantities Proposition 5 relates. Over $3 \times 8 \times 5 \times 20 = 2{,}400$ trials with $\delta'' = 0.05$ the bound of Theorem 6 was violated 0 times, the loss never exceeded ϵ_{KS}, and the median ratio of loss to drift was 0.93 for rare-event shifts, 0.66 for scale shifts, and 0.17 for location shifts; the largest loss at any drift below the alert threshold 0.15 was 13.9 points, and below 0.05 it was 6.2 points. Figure 5 shows the same data as heat maps over shift magnitude and window size, where the observed 95th-percentile loss grows with the magnitude and is nearly flat in W, while the slack to the bound is the finite-sample term, 6.1 points at $W = 500$ and 1.4 at $W = 10{,}000$.

Fuzzing the emission logic (Figures 7 and 8). Property-based testing checks a stated property on random inputs rather than on hand-written cases (Claessen & Hughes, 2000); here the property is the conclusion of Theorem 3, the inputs are cost models and drift sequences, and the fuzz complements the mechanized proof by exercising an implementation the proof abstracts. We ran Algorithm 1 and its naive counterpart side by side on 1,000 independent monitors per configuration, each with its own random cost model over 8 states and 16 actions, a sticky random Markov structure, a base policy that proposes a random action, and a local proposer that searches only the five actions

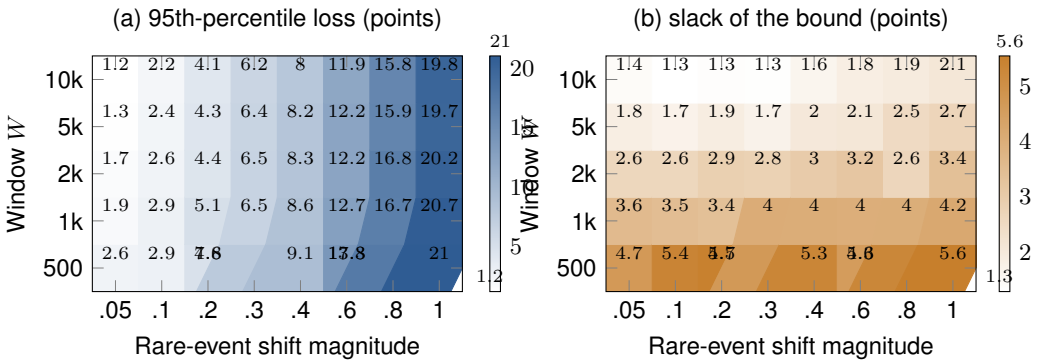


Figure 5: Rare-event shifts, with the observed 95th-percentile coverage loss in panel (a) and the slack between the bound of Theorem 6 and the observed loss in panel (b).

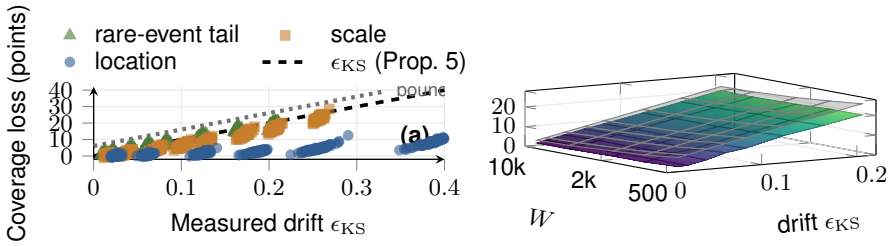


Figure 6: Coverage transfer under score drift on synthetic scores, as loss at the threshold against measured drift over 2,400 trials in panel (a) and as the 95th-percentile loss in points (vertical axis) under the bound of Theorem 6 for rare-event shifts in panel (b).

nearest the proposal, so that, like gradient descent with restarts, it can fail while $\mathrm{Adm}(m, s)$ is non-empty. The temperature follows a mean-reverting random walk in $[1, \tau_{\max}]$ refit every 10 steps, with $\tau_{\max} \in \{1, 1.25, 1.5, 2, 3\}$, and the cost models come from four families, uniform, clustered, heavy-tailed, and a near-boundary family in which every predicted cost sits close to d_t. Over the resulting 10^7 monitor steps the naive design reached a fallback on 1.89% of steps and emitted an inadmissible action on 14.7% of those fallbacks, 27,757 times in all, of which 20,319 reused a certificate issued at a different state and 7,438 one issued at the same state under a lower temperature; with no drift ($\tau_{\max} = 1$) the temporal failures vanish exactly and the spatial ones remain, which is the content of the two cases of Proposition 1. Lemma 2's monotone condition licensed 144,190 same-state reuses and every one of them was admissible when re-checked; VETO emitted an inadmissible action 0 times in 10^7 steps and refused on 0.154% of them, with Figure 7 giving the counts by family and drift level and Figure 8 the trace of one near-boundary monitor, where the unsound naive emissions cluster in the periods after the temperature has risen and most stored certificates are stale.

Cost of the decision path (Figure 9). Both designs run the same ensemble pass and the same guard, so we time only the calibration work each puts on the decision path, reading τ and appending the score for VETO, against recomputing the quantile of the W most recent scores after the $O(1)$ update of adaptive conformal inference, in a NumPy reference implementation on one core of an Intel Xeon at 2.10 GHz, as the median over seven runs of 2,000 decisions. The temperature check costs 0.20 to 0.24 μs for every W from 250 to 40,000 with 8 bytes of state, whereas the sliding-window quantile grows from 36 μs to 506 μs with $8W$ bytes of state, a ratio of 337$\times$ at the $W = 5000$ used in Section 7; the absolute numbers are machine-specific, the scaling is not.

E EXPERIMENTAL DETAILS

Environment and cost. Safety-Gymnasium (Ji et al., 2023) `SafetyPointGoal1-v0` and `SafetyCarGoal1-v0`, MuJoCo navigation tasks in which the agent must reach a goal while

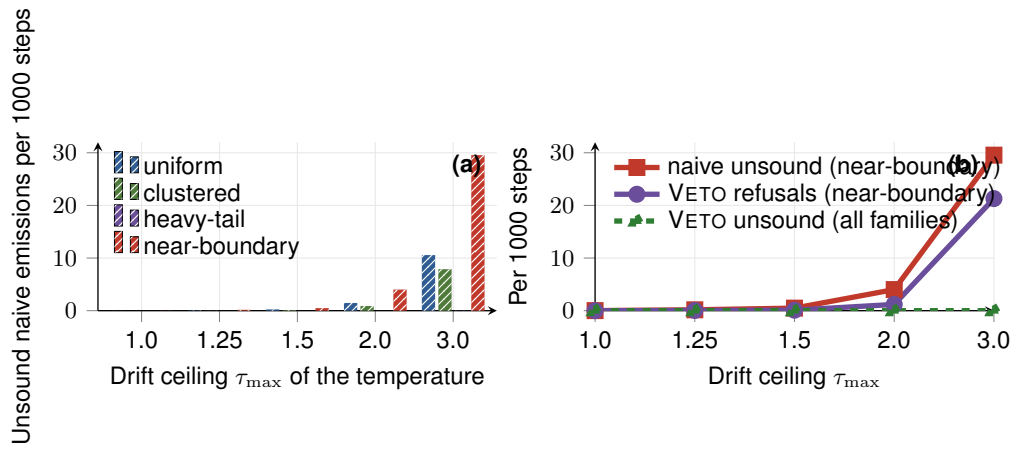


Figure 7: Emission-logic fuzzing over 10^7 monitor steps, with unsound naive emissions by cost-model family and drift ceiling in panel (a) and the near-boundary family against VETO's refusals in panel (b).

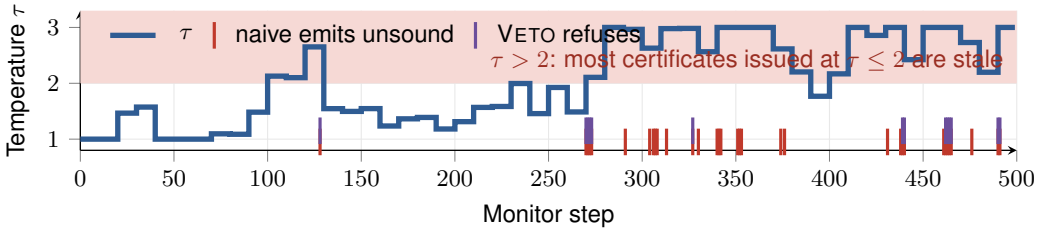


Figure 8: One fuzzed monitor, showing its temperature, the unsound naive emissions, and the VETO refusals.

avoiding hazards; the per-step cost is the hazard indicator, so $d = 0.5$ counts every hazard contact as a violation, and episodes are truncated at horizon 200 for evaluation.

Base policies. OmniSafe 0.5.0 implementations of PPO-Lagrangian (`PPOLag`), CPO and PID-Lagrangian (`CPPOPID`) with OmniSafe's default hyperparameters, an episodic cost limit of 25, 500k environment steps, and seeds 0 to 4, trained on 32 vCPUs with two threads per run; the final checkpoint of each run is evaluated deterministically with the checkpoint's observation normaliser.

Envelope. The ensemble has $K = 5$ members, each a three-layer perceptron with 256 tanh units per layer, trained with Adam at learning rate 3×10^{-4} on bootstrap resamples of (s, a, c) triples. The triples come from 50,000 steps of the base policy with 30% uniform exploration on the unshifted task, with standardised observations, and each member trains for 30 epochs. The base multiplier is $\beta = 1$, since only $\beta\tau$ enters the bound and τ is calibrated. The margin coefficient is $\alpha = 0.5$ and the target miscoverage is $\delta' = 0.02$. The temperature τ is the split-conformal quantile of the normalised score $(c - \hat{c})/\sigma$ on the most recent $W = 5000$ scores, refit every 20 episodes, and the Kolmogorov–Smirnov distance between consecutive windows raises an alert above 0.15. Projection is gradient descent on the bound surface with three restarts batched into one pass and at most ten iterations. The fallback ladder is the last admitted action, the zero action, and a cache of the four most recent admitted actions, each re-checked at the current state, and a refusal hands control to the zero action. The released envelope also offers a sampled fallback, which draws sixteen candidates from the policy's own distribution and emits the most probable one that passes the re-check before it turns to the ladder; Section 7 did not use it. The naive fallback emits the last admitted action without the re-check, as in Proposition 1, and the conformal filter uses the same ensemble with a quantile updated by adaptive conformal inference and no re-check.

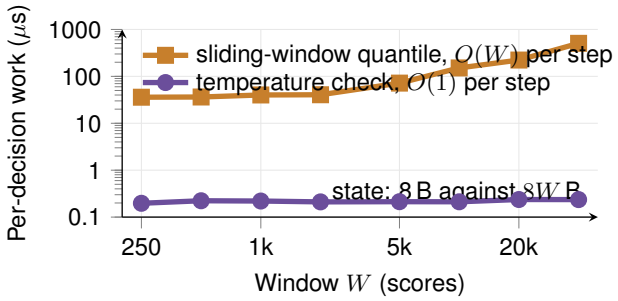


Figure 9: Per-decision calibration work against the window size.

Choosing α. The margin coefficient trades reward for the per-step bound, and the pilot fixed it at $\alpha = 0.5$ before any run. The procedure we recommend, and did not have time to run, holds out one shift type as a validation shift, evaluates $\alpha \in \{0, 0.1, 0.25, 0.5\}$ on it, chooses the smallest α whose empirical coverage meets $1 - \delta'$, and freezes it for every test shift. The released harness takes α as a flag and caches the ensemble, so the sweep is evaluation only.

Shift protocol. The released wrappers also implement reward, cost and initial-state shifts, which the pilot did not run. Dynamics shift scales gravity, friction and body mass by $1 + 0.15m$, $1 + 0.30m$ and $1 + 0.25m$; observation noise adds Gaussian noise of standard deviation $0.05m$ in normalised units; rare-event shift applies a velocity impulse of standard deviation $0.5m$ with probability $0.01m$ per step, for magnitudes $m \in \{0.5, 1.0, 2.0\}$; magnitude-1.0 and no-shift cells run 100 episodes per seed and the others 50.

Compute. Per-decision cost is measured on a NumPy reference implementation of Algorithm 1 (script `e5_reference_impl.py`) run end to end on a toy constrained navigation task, on one core of an Intel Xeon at 2.10 GHz, over 24,000 decisions with five recalibrations. A step the guard passes costs 0.137 ms, a step that projects 3.20 ms, the 95th percentile over all decisions is 3.45 ms, and recalibration (lines 14–20) is off the decision path and amortized over 20 episodes. The sliding-window quantile that adaptive conformal inference recomputes at every step costs 0.198 ms per step at $W = 5000$ on the same core. The PyTorch envelope used in Section 7 costs 19.47 ms per decision on PointGoal and 9.37 ms on CarGoal for PPO-Lagrangian, because projection runs on nearly every PointGoal step. The absolute numbers are machine-specific, and the split between the constant guard cost and the occasional projection cost is not. Training the 30 base policies took about two hours on a 32-vCPU machine at 16 concurrent runs, and evaluating the 1,200 cells took about five hours on the same machine, most of it in projection.

F The Feasibility Gap in Full

Theorem 3 says that every emitted action is admissible now. It does not say that admissible actions will keep existing. VETO filters one step at a time, so nothing in the analysis prevents the filtered policy from entering a state where $\mathrm{Adm}(m, s) = \emptyset$ and Corollary 4 forces a refusal. The gap is in fact wider than an empty admissible set, because the fallback examines only a finite candidate list rather than searching $\mathcal{A}$. A refusal therefore signals that none of the actions VETO examined was admissible, which may happen while admissible actions exist elsewhere in $\mathcal{A}$. Enlarging the candidate list narrows this second gap but does not close it, and only the first gap is intrinsic. Model predictive safety filters and control barrier functions avoid this by computing a control invariant set or a terminal safe set in advance (Wabersich & Zeilinger, 2023; Ames et al., 2019), which is what makes them recursively feasible. That construction needs a dynamics model, and VETO has none by design. We therefore state the gap rather than hide it. Assumption 1 is exactly where recursive feasibility enters, and it is assumed rather than established. Closing it for a model-free filter is open, and we regard it as the most valuable next step.

Handling a refusal. In our experiments a refusal hands control to a fixed safe policy, such as braking to a stop or holding the last rate, and logs the state for review. An unhandled refusal is itself a failure, so a deployment must bind such a fallback before the envelope is put in the loop.