LEARNING STATE-DEPENDENT REACH-AVOID BOUNDS WITH PAC GUARANTEES

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned control policies can succeed on average yet fail from particular initial states. Existing reach-avoid certificates either require a model of the dynamics with Lipschitz bounds and a grid over the state space, or certify a single success probability. We introduce Reach-Avoid Probability Certification (RAPC), which learns a function $q(x)$ that lower-bounds the probability that a policy started at x reaches a target before an unsafe set, and certifies it from simulator rollouts alone. With probability at least $1 - \delta$ over learning and certification, a returned q overstates the true success probability only on initial states of total probability at most ϵ, even when the policy and bound are updated adaptively. The key step links an unobservable event, an overstated bound at a start, to an observable one: a failed binomial confidence check on repeated rollouts from that start. Calibrating q to these confidence limits gives a learner-independent lower bound on its acceptance probability and, for a fixed calibration pipeline, certifies q without a separate validation batch. On a 12-dimensional Hopper environment and a PointGoal environment with 44-dimensional observations, RAPC certifies bounds in all ten seeds of each task. On a wider Hopper start distribution, the learned bound averages 0.82 over ten seeds, against 0.51 for the largest constant accepted by the same test. We release our code at `https://anonymous.4open.science/r/iclr-pac-BBBD/`.

1 INTRODUCTION

A control policy can succeed on average while remaining unreliable from particular starting states. In reach-avoid tasks, where success means reaching a target before entering an unsafe set (Summers & Lygeros, 2010), a navigation policy may work well in open space and fail next to obstacles, and a legged robot may clear a low ceiling from some starting positions but not from others. A lower bound on the success probability that depends on the starting state tells a user where the policy can be trusted, and a learned bound can be evaluated at a new start without further rollouts.

Existing tools either need a model of the dynamics, certify a single number, or give no guarantee. Certificate-based methods, such as barrier certificates (Prajna et al., 2007; Xue et al., 2021) and reach-avoid supermartingales (Žikelić et al., 2023; Badings et al., 2025), prove a bound that holds uniformly over a set of initial states. They require a model of the dynamics and, for learned certificates, Lipschitz bounds and a discretization of the state space, so their cost grows quickly with dimension; on our three-dimensional Planar task, most runs of the latter time out (Appendix D.4). Statistical model checking and direct Monte Carlo estimation need only simulations (Younes & Simmons, 2002; 2006), but they certify one probability: the success rate from a given start, or averaged over a distribution of starts. Value functions learned for safety and reach-avoid problems (Fisac et al., 2019; Hsu et al., 2021; Bansal & Tomlin, 2021) score every start, but the learned values come with no finite-sample guarantee.

We propose Reach-Avoid Probability Certification (RAPC). RAPC learns a function q that predicts, for each initial state, a lower bound on a policy's probability of success within a fixed horizon, and certifies it from restartable simulator rollouts alone. The certificate is a probably approximately correct (PAC) statement with the two-level structure of PAC prediction sets (Park et al., 2020): with confidence at least $1 - \delta$ over the entire procedure, the returned function lies below the true success probability at a fresh random start with probability at least $1 - \epsilon$. It therefore permits an exceptional

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.

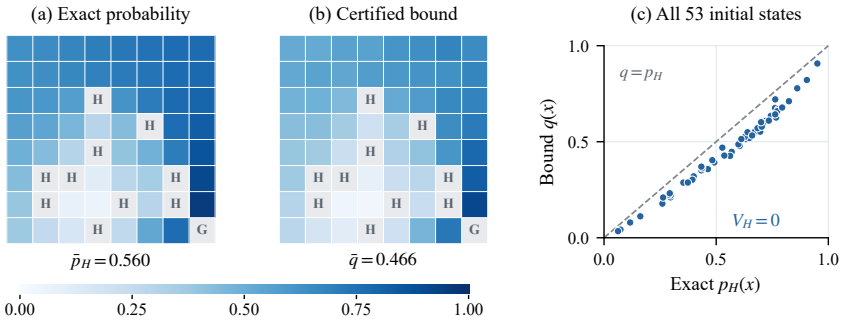

Figure 1. A bound that varies with the starting state. For one Frozen Lake policy, (a) shows the exact reach-avoid probability, (b) the learned and certified lower bound, and (c) all 53 nonterminal starts; here every prediction lies below the exact probability. Gray cells are terminal states (Appendix D.2).

set of starts, of probability at most ϵ, and in exchange needs no model, Lipschitz constant, or grid. Figure 1 shows such a bound next to exact success probabilities.

The difficulty is that the quantity being bounded is itself an unknown probability. A rollout reveals success or failure, but not whether q overestimates the success probability at its start. Our key observation is that repeated rollouts from a sampled start turn this unobservable event into an observable one: if the bound overstates the success probability at that start, a binomial confidence check on the rollouts fails with high probability. The observed failure rate of these checks therefore bounds the probability mass of overstated starts, and a binomial test on a fresh batch of starts certifies the bound. Giving each round of learning its own share of the error budget keeps the guarantee valid when the policy and the bound are refined after a failed test.

A function trained to predict success probabilities usually fails this test, because finitely many rollouts cannot confirm values close to the truth. We therefore calibrate the learned function to the confidence limits that the rollouts can support, using an order statistic of held-out residuals (Vovk, 2012). The resulting lower bound on the probability of passing the test is the same for every learner; the learner determines only how high the certified bound is. With enough calibration data, the calibration sample certifies the bound by itself, without a separate validation batch.

On Hopper and PointGoal, with 12-dimensional states and 44-dimensional observations respectively, RAPC certifies bounds in all ten seeds of each task. When starts differ in difficulty, the learned bound is far above the best constant that the same test accepts: on a wider Hopper start distribution, 0.82 against 0.51, averaged over ten seeds.

Contributions.

- A PAC certificate for a learned, state-dependent lower bound on reach-avoid success probabilities from restartable rollouts, valid across adaptive updates (Theorem 1).
- Calibration to finite-rollout confidence limits, with a learner-independent lower bound on acceptance and a condition under which calibration alone certifies the bound (Section 5.2).
- Evidence on Hopper and PointGoal that the state-dependent bound improves on the best constant where success varies across starts and supports certified restriction to reliable starts (Section 6.2).

2 Related Work

Reach-avoid certificates. Model-based stochastic reachability computes success probabilities from a known model (Abate et al., 2008; Summers & Lygeros, 2010; Vinod & Oishi, 2021). Barrier-like conditions (Xue, 2025; 2026), RASM (Žikelić et al., 2023), and LogRASM (Badings et al., 2025) certify bounds uniformly over an initial set; data-driven variants learn barrier certificates from samples (Nejati et al., 2023; Salamati et al., 2024), certify interval MDPs (Nazeri et al., 2025), or bound violation probabilities by scenario optimization (Calafiore & Campi, 2006; Campi & Garatti, 2011), and Vincent et al. (2024) certify performance statistics of a policy from rollouts. Our certificate gives up uniformity for an exceptional set of small probability mass and in exchange needs only

restartable rollouts (Appendix B.7). **Statistical verification.** Statistical model checking verifies probabilistic properties from simulated executions and hypothesis tests (Younes & Simmons, 2002; 2006); its nested properties, whose inner tests carry their own errors, resemble our local checks inside a batch test, and smoothed model checking learns satisfaction probabilities over model parameters under smoothness assumptions (Bortolussi et al., 2016). SMC certifies a probability for a given initial condition, and smoothed model checking gives a Bayesian estimate over model parameters; we certify a function of the initial state, with frequentist guarantees across adaptive updates. **Conformal calibration.** PAC prediction sets (Park et al., 2020), training-conditional conformal inference (Vovk, 2012), and risk control (Bates et al., 2021; Angelopoulos et al., 2021) calibrate losses observed on labeled data; our label, the success probability at a start, is never observed and is not identified by one binary outcome per start (Barber, 2020), so we calibrate local rejections from repeated rollouts instead. Lindemann et al. (2024) survey conformal methods in control, and Lin & Bansal (2024) verify neural reachable tubes. **Policy learning.** Constrained and reach-avoid reinforcement learning (Achiam et al., 2017; Stooke et al., 2020; Ni & Kamgarpour, 2025) shapes the policy but returns no certified bound; we certify its outputs (Appendix B.10).

3 Problem Setup

We model the system as a controlled Markov process with measurable state space $\mathcal{X}$ and action space $\mathcal{U}$. The unknown kernel $\mathsf{P}(dx' \mid x, u)$ describes the next state after action u at state x. A fixed Markov policy $\pi(du \mid x)$ gives the closed-loop kernel $\mathsf{P}^\pi(dx' \mid x) = \int \mathsf{P}(dx' \mid x, u)\pi(du \mid x)$, and $\mathbb{P}_x^\pi$ denotes the law of the closed-loop trajectory $x_0 = x$, $x_{t+1} \sim \mathsf{P}^\pi(\cdot \mid x_t)$. Let $\mathcal{T}$ and $\mathcal{F}$ be disjoint measurable target and failure sets, and let μ be the initial-state distribution of interest. Let $\tau_\mathcal{T} = \inf\{t \geq 0 : x_t \in \mathcal{T}\}$ and $\tau_\mathcal{F} = \inf\{t \geq 0 : x_t \in \mathcal{F}\}$ be the first hitting times, with $\inf \varnothing = \infty$.

Finite-horizon objective. Success means reaching the target before failure within an integer horizon $H \geq 1$. For policy π starting at x, its probability is

$$p_H^\pi(x) = \mathbb{P}_x^\pi(\phi_H), \qquad \phi_H = \{\tau_\mathcal{T} \leq H,\ \tau_\mathcal{T} < \tau_\mathcal{F}\}. \tag{1}$$

Reaching neither set by H counts as failure. Starting in $\mathcal{T}$ succeeds immediately and starting in $\mathcal{F}$ fails immediately, so a rollout can stop at either set without changing its label. We assume only a simulator that can be restarted at any queried full state and returns independent rollouts (Assumption 1, Section 4.2); the method needs no analytical form of P and no Lipschitz constant, and every guarantee concerns the law that generates these rollouts, for a specified horizon, target and failure sets, policy, and initial-state distribution.

Policy and certificate. We seek a policy π and a measurable function $q : \mathcal{X} \to [0, 1]$ reporting lower bounds on success. The *violation mass* is the probability of drawing a start at which this proposed bound is too high:

$$V_H(\pi, q) = \Pr_{x\sim\mu}\{q(x) > p_H^\pi(x)\}. \tag{2}$$

Given $\epsilon, \delta \in (0, 1)$, we want to limit the chance of returning a pair whose violation mass exceeds ϵ:

$$\Pr\{\text{the procedure returns } (\pi, q) \text{ with } V_H(\pi, q) > \epsilon\} \leq \delta. \tag{3}$$

The probability in (3) includes all learning and validation randomness. Equivalently, with confidence at least $1 - \delta$ over the complete procedure, every returned pair satisfies

$$\boxed{\mathbb{P}_{x\sim\mu}\big\{p_H^\pi(x) \geq q(x)\big\} \geq 1 - \epsilon.} \tag{4}$$

The outer probability is over a fresh random start, so $[q(x), 1]$ covers the latent target $p_H^\pi(x)$ with probability at least $1-\epsilon$. The guarantee permits an exceptional set of starts of mass at most ϵ; it is not a separate confidence statement at each specified start. For example, at $\epsilon = 0.01$ and $\delta = 0.05$, at most 5% of runs return a bound that overstates success on more than 1% of the initial-state distribution; a run may also return no bound. Table 3 in Appendix A collects the notation.

Informativeness. The function $q \equiv 0$ is always valid but uninformative, so we seek both successful policies and high valid bounds. When $V_H(\pi, q) \leq \epsilon$, average success satisfies

$$\mathbb{E}_\mu[p_H^\pi(x)] \geq \max\{0, \mathbb{E}_\mu[q(x)] - \epsilon\}. \tag{5}$$

Algorithm 1 Learning reach-avoid bounds with budgeted certification (L_β and $B_{N,a}$ as in (6) and (8))

Input: $\mu, \mathcal{T}, \mathcal{F}, H, \epsilon, \delta$; maximum rounds R; N, M, β; compute budget
Budget rule: at any stage, **return** UNKNOWN if the compute budget is exhausted
Train an initial policy; learn and calibrate a candidate using non-validation trajectories
for $t = 1, \ldots, R$ **do**
  Freeze (π_t, q_t); set $\delta_t = \delta/R$ and initialize an empty list $\mathcal{C}_t$
  Draw fresh $x_1, \ldots, x_N \overset{\text{iid}}{\sim} \mu$
  For each x_i: draw M independent horizon-H rollouts under π_t, count successes S_i, and append x_i to $\mathcal{C}_t$ if $L_\beta(S_i; M) < q_t(x_i)$
  if $B_{N,(1-\beta)\epsilon}(|\mathcal{C}_t|) \le \delta_t$ **then**
    return (π_t, q_t) with (4) at procedure-level confidence $1 - \delta$
  end if
  Refine the policy using learning rollouts, including starts from $\mathcal{C}_t$
  Collect new learning rollouts under the updated policy; refit and recalibrate the bound
end for
return UNKNOWN

Appendix A.5 gives the proof and the corresponding bound after conditioning on a subset of initial states.

4 Statistical Certification of Learned Bounds

RAPC (Algorithm 1) alternates learning (Section 5) and verification. Verification freezes a candidate pair and checks it on fresh data at two levels, a local confidence check at each sampled start and a binomial test on how many starts fail it (Section 4.1); splitting the error budget across rounds keeps the guarantee valid when learning resumes after a failed test (Section 4.2).

4.1 Verifying a Candidate Bound

At state x, let S_x count successes in M rollouts of horizon H, so $S_x \sim \mathrm{Bin}(M, p^\pi_H(x))$. We compute a one-sided Clopper–Pearson (CP) lower confidence limit (Clopper & Pearson, 1934) with error level $\beta \in (0, 1)$:

$$L_\beta(0; M) = 0, \qquad L_\beta(s; M) = \text{the unique } \ell \in (0,1) \text{ with } \Pr_{S' \sim \mathrm{Bin}(M,\ell)}\{S' \ge s\} = \beta \quad (s \ge 1). \tag{6}$$

The local check passes when $L_\beta(S_x; M) \ge q(x)$: the confidence limit supports the proposed bound. Otherwise the check rejects the bound at x, and we call x a rejected start. A local rejection means insufficient evidence, so even a valid bound can be rejected at some starts; acceptance or rejection of the candidate as a whole refers to the batch test below.

Lemma 1 (Local detection of overestimation)**.** *For every $p \in [0,1]$, if $S \sim \mathrm{Bin}(M, p)$, then $\Pr\{L_\beta(S; M) > p\} \le \beta$. Consequently, at any state with $q(x) > p^\pi_H(x)$, the probability that the local check supports $q(x)$ is at most β.*

Lemma 1 says that each state with an overstated bound is rejected with probability at least $1 - \beta$. Let $Z = \mathbf{1}\{L_\beta(S_x; M) < q(x)\}$ and $r = \Pr(Z = 1)$ for a fresh $x \sim \mu$ and its rollouts. Integrating over states gives

$$r \ge (1 - \beta) V_H(\pi, q). \tag{7}$$

Thus $r \le (1-\beta)\epsilon$ implies $V_H(\pi, q) \le \epsilon$. We test for excessive r using a fresh batch. Set $a = (1-\beta)\epsilon$, draw N independent states and their rollouts, and count rejections $K_N = \sum_{i=1}^N Z_i$. Define the binomial lower tail

$$B_{N,a}(k) = \sum_{j=0}^{k} \binom{N}{j} a^j (1-a)^{N-j}. \tag{8}$$

Set $B_{N,a}(k) = 0$ for $k < 0$. At round t of the procedure, the test has error level δ_t, its share of the total budget δ (Section 4.2). It accepts the candidate when $B_{N,a}(K_N) \le \delta_t$, that is, when at most $k_* = \max\{k : B_{N,a}(k) \le \delta_t\}$ of the N starts are rejected. A candidate whose rejection probability is at least a would produce so few rejections with probability at most δ_t.

4.2 Controlling False Acceptance across Rounds

Each new candidate may use earlier training and test results. To keep repeated testing valid, we assign each round part of the total error budget and analyze its fresh validation data conditional on everything observed before that round. The analysis needs the following sampling conditions.

Assumption 1 (Rollout access and validation sampling)**.** *The simulator can restart the process at each queried full state. At each validation round, the candidate pair and the sampling parameters are determined by the history available before validation. Conditional on this history, the N initial states are independent draws from μ. Conditional on the history and these states, all NM rollouts are mutually independent. Each follows P from its assigned state under the frozen candidate policy, including its randomization. Validation uses fresh random draws at each round.*

Theorem 1 (PAC coverage under adaptive learning and certification)**.** *Fix H, μ, and $\epsilon, \delta \in (0,1)$. At round t, let (π_t, q_t) and $N_t, M_t, \beta_t, \delta_t$ depend arbitrarily on the past, but be fixed before new validation data are observed. Assume Assumption 1, $N_t, M_t \geq 1$, $\beta_t, \delta_t \in (0,1)$, and $\sum_t \delta_t \leq \delta$ almost surely. Accept a pair only after a complete validation batch and only if $B_{N_t,(1-\beta_t)\epsilon}(K_{N_t}) \leq \delta_t$. Then, with confidence at least $1-\delta$ over all learning and validation data, every accepted pair (π_t, q_t) satisfies* (4)*. Equivalently, the probability of accepting any pair with $V_H(\pi_t, q_t) > \epsilon$ is at most δ. Returning the first accepted pair therefore satisfies* (3)*.*

Proof sketch. Conditional on the preceding history, the pair is fixed and its rejection count is binomial. By (7), an invalid pair has rejection probability greater than $(1-\beta_t)\epsilon$, so the test accepts it with probability at most δ_t. Taking expectations and summing over rounds proves the claim (Appendix A).

For at most R attempts, $\delta_t = \delta/R$ suffices. The guarantee holds for any learner and sequence of policy updates, without a feasible initial policy, convergence, or an approximation bound. If no candidate passes within R rounds or the compute budget, the procedure returns UNKNOWN; as with undecided outcomes in statistical model checking (Younes, 2006), this controls false acceptance without an indifference region, a band near ϵ left uncontrolled.

4.3 Choosing the Verification Budget

The batch test needs enough starts to detect violations of mass ϵ, and the local check needs enough rollouts to support the proposed value. Even with no rejections, acceptance requires $N \geq \lceil \log \delta_t / \log(1-(1-\beta)\epsilon) \rceil$; and since M successes give only $L_\beta(M; M) = \beta^{1/M}$, supporting a bound value $b \in (0,1)$ at a start requires $M \geq \lceil \log \beta / \log b \rceil$. Both conditions are necessary but not sufficient. The returned function may exceed the local ceiling $\beta^{1/M}$ at a few starts, because the batch test tolerates rejections. A round uses at most NMH simulated transitions (Appendix C.1).

5 Learning and Calibrating Candidate Bounds

Theorem 1 holds for any frozen candidate, so the choices below affect how often candidates pass and how high the certified bound is, which we call its *height*, but not the validity of the certificate.

5.1 Candidate Learners

We use candidates of the form $q(x) = \min\{\sigma(z(x))/\widetilde{\rho}(x), 1\}$, where $\sigma(u) = 1/(1+e^{-u})$ is the logistic function, z is a learned logit, and $\widetilde{\rho}(x) \in (0,1]$ is a denominator that calibration keeps fixed. Monte Carlo (MC) regression fits binary success labels, and Probability-TD fits the undiscounted success recursion by temporal differences (TD); both use $\widetilde{\rho} \equiv 1$. Ratio-TD learns the discounted value $v(x) = \mathbb{E}^\pi_x[\gamma^{\tau_\mathcal{T}} \mathbf{1}_{\phi_H}]$, with discount $\gamma \in (0,1)$, by TD. Since $v = \rho\, p^\pi_H$ with $\rho(x) = \mathbb{E}^\pi_x[\gamma^{\tau_\mathcal{T}} \mid \phi_H]$ (Proposition 2), dividing v by an estimate $\widetilde{\rho}(x)$ of this conditional discount, obtained from an auxiliary hitting-time network, recovers the success probability. Because TD targets bootstrap from partial trajectories, Ratio-TD's discounted value can be learned alongside the policy's critic during training, whereas MC regression needs complete rollouts of the final policy; calibration below corrects the bias of either. We also compare an uncorrected discounted value and a constant (losses and budgets in Appendices C.3 and D.3).

5.2 Calibrating a Candidate to the Verifier's Evidence

Even an accurate probability estimate is rejected by the local check whenever it exceeds the confidence limit that M rollouts can support. Calibration therefore targets that confidence limit rather than the probability. It keeps the policy fixed and has two steps.

Fit. At each of many fitting states (8192 in the control experiments), m_{fit} additional rollouts (64 there) with S_x^{fit} successes give the success frequency $\widehat{p}(x)$ and the training target

$$\widehat{p}(x) = S_x^{\mathrm{fit}}/m_{\mathrm{fit}}, \qquad y(x) = L_\beta\big(\,\mathrm{round}(M\widehat{p}(x)); M\big), \tag{9}$$

the confidence limit that M validation rollouts would give at that frequency. We refit z to $\mathrm{logit}(\widetilde{\rho}\, y)$; this plug-in target can raise or lower predictions.

Shift. On n separate states with M rollouts each, we compute the residuals

$$e_i = z(x_i) - \mathrm{logit}\,\big(\widetilde{\rho}(x_i) L_\beta(S_i; M)\big), \qquad i = 1, \ldots, n, \tag{10}$$

where $\mathrm{logit}(0) = -\infty$, so a state with no successes has $e_i = +\infty$. The calibrated candidate subtracts a scalar shift c from the logit:

$$q_c(x) = \min\{\sigma(z(x) - c)/\widetilde{\rho}(x), 1\}. \tag{11}$$

The local check rejects q_c at a state exactly when its residual exceeds c, so taking c as a high order statistic of the residuals caps the local rejection rate for every learner and makes the probability of passing the next validation batch predictable from the sampling parameters alone. Let $\alpha \in (0, 1)$ be the tolerated rate of local rejections, and let $e_{(k)}$ denote the k-th smallest residual.

Lemma 2 (Calibration controls local rejections and acceptance)**.** *Fix π, measurable $z : \mathcal{X} \to \mathbb{R}$ and $\widetilde{\rho} : \mathcal{X} \to (0, 1]$, integers $n, M, N \geq 1$, and $\alpha, \beta, \epsilon, \delta_t \in (0, 1)$. Let $(x_i, S_i)_{i=1}^n$ be iid with $x_i \sim \mu$ and $S_i \mid x_i \sim \mathrm{Bin}(M, p_H^\pi(x_i))$. Set $k = \lceil (n+1)(1-\alpha) \rceil$ and take $c = e_{(k)}$ if $k \leq n$; otherwise set $c = \infty$ and $q_c \equiv 0$. Let r_c be the probability, given the calibration sample, that the local check rejects q_c at a fresh start.*

(i) For an independent pair (x, S) with the same law,

$$\Pr\{L_\beta(S; M) < q_c(x)\} = \mathbb{E}[r_c] \leq \alpha. \tag{12}$$

For continuous residuals and $k \leq n$, $r_c \sim \mathrm{Beta}(n + 1 - k, k)$; if $k > n$, $r_c = 0$.

(ii) A fresh validation batch satisfying Assumption 1 passes the test of Section 4.1 with probability $\mathbb{E}[B_{N,r_c}(k_*)] \geq \mathbb{E}[B_{N,W}(k_*)]$, where $W \sim \mathrm{Beta}(n + 1 - k, k)$ ($W = 0$ if $k > n$), with equality for continuous residuals. This lower bound depends only on $n, \alpha, N, \beta, \epsilon$, and δ_t.

The proofs are in Appendix A.4. With $n = 2048$, $\alpha = \epsilon/2$, $N = 10^4$, and $k_* = 74$, the acceptance probability is at least 0.92. The learner determines the height of q_c through the residual quantile.

In words, the calibration sample of the exact-rank shift can itself serve as the certificate.

Proposition 1 (Training-conditional certificate from exact calibration)**.** *Under the conditions of Lemma 2 with $k \leq n$,*

$$\Pr\{V_H(\pi, q_c) > \epsilon\} \leq B_{n,(1-\beta)\epsilon}(n - k),$$

where the probability is over the calibration sample, conditional on the history before shift calibration (Vovk, 2012).

At $n = 2048$ and $\alpha = 0.005$ this tail is $0.0037 < \delta_t = 0.005$, so exact calibration certifies the candidate at level δ_t and the validation batch (Algorithm 1, lines 6–10) can be omitted; fresh calibration blocks extend this across rounds. The shift is global, however: under the exact rule, more than $n - k$ calibration states without successes (10 of 2048 here) force $c = \infty$ and $q_c \equiv 0$, however reliable the other starts are. Our synthesis runs therefore use a linearly interpolated residual quantile with confidence limits clipped away from zero, which keeps such residuals finite but is not covered by Lemma 2, and certify the result by independent validation (Appendix A.4); Section 6.3 compares the two routes.

5.3 Policy Improvement and Verification Feedback

We improve policies with RESPO (Ganai et al., 2023), which optimizes target rewards and failure costs with reward shaping toward the target; since the certificate covers only the returned pairs, any other optimizer could take its place. After a batch test rejects the candidate, half the training starts come from its rejected starts. We then collect new trajectories under the updated policy, refit the candidate on them (the same data for every learner, for a controlled comparison), recalibrate it with the Ratio-TD denominator $\widetilde{\rho}$ held fixed, and verify with fresh data. We call this mode *Feedback* and compare it with continued training (*Train*) and with recalibration under a fixed policy (*Calibrate*).

6 Experiments

We evaluate three questions. (Q1) Does the verifier control false certification, and does the complete procedure return informative certificates? (Q2) What does a state-dependent bound add over a constant? (Q3) How do calibration, the learner, and sampling budgets affect bound height and cost?

Setup. We use four control tasks (Appendix D.1). Linear (two-dimensional state, adapted from Žikelić et al. (2023)) and Planar (three-dimensional) are low-dimensional stochastic systems. In Hopper, a planar legged robot with a 12-dimensional state must bring its head into a goal region beyond a ceiling block without the head entering the block. In PointGoal, a point robot with a six-dimensional physical state, observed through 44 lidar and inertial readings, must reach a goal disk without touching eight hazards or leaving the arena. Policies and bounds read observations that are functions of the state, so the guarantee applies unchanged. Candidates use Ratio-TD unless stated otherwise. We report the mean certified bound $\bar{q}$ over each accepted validation batch. Comparators are the largest constant that the same batch accepts, direct Monte Carlo certification of a fixed region, the uniform certificates of RASM and LogRASM on Linear and Planar (Appendix D.4), and, under a frozen policy, the five candidate learners. Control synthesis uses $\epsilon = 0.01$ and $\delta = 0.05$ over at most ten rounds ($\delta_t = 0.005$), with $N = 10{,}000$ starts, $M = 1000$ rollouts per start, $H = 400$, and $\beta = 10^{-3}$; a batch passes with at most $k_* = 74$ rejected starts. Appendix C.2 gives the settings for the fixed-policy comparisons and Frozen Lake.

6.1 Q1: Verification Errors and Certified Outputs

We first test the verifier where success probabilities are known: synthetic finite-state candidates and Frozen Lake policies ($N = 2000$, $M = 1000$, 5000 repetitions per configuration). In all 68 configurations (Appendix C.4), the analytic acceptance probability lies within the 95% CP interval of the simulated acceptance frequency. For an invalid candidate with $V_H = 1.1\epsilon$, spending the full δ in each of ten tests gives false-return frequency 0.119, and allocating δ/R reduces it to 0.012, below 0.05 (95% CP intervals in Appendix C.4). A valid candidate can still fail: for $q = [p - m]_+$ with exact probability p, increasing the margin m from 0.08 to 0.09 raises acceptance from 0.086 to 0.9998 (Appendix C.4).

The complete procedure with Feedback refinement returns certified pairs in all ten seeds on each of Linear, Planar, Hopper, and PointGoal (Table 1); on Hopper, policy training recovers an initially unsuccessful seed that fixed-policy calibration cannot certify within ten rounds. RASM and LogRASM need an analytical or Lipschitz-bounded model and a state-space grid, which Hopper and PointGoal do not provide (Appendix D.11); on Linear, LogRASM certifies 0.9999, above our local ceiling 0.9931, whereas on Planar both time out in most runs within 30 minutes even for a target of 0.1, while RAPC takes about two minutes per seed (Table 13 in Appendix D.4). Comparisons with continued training and weak initialization are in Appendix D.4.

6.2 Q2: What State Dependence Adds

Compared with a constant. The best constant that the same test accepts is the (k_*+1)-th smallest local confidence limit of the batch, an optimistic comparator for any constant fixed in advance (Table 1). On PointGoal, mean bounds are 0.940 versus 0.909; for the two imperfect Hopper policies they are 0.747 and 0.816 versus 0.165 and 0.709, because the exception mass ϵ is too small to discard the entire low-success region.

Under the original Hopper distribution, every start is far from the ceiling and eight of ten policies succeed almost everywhere. Rerunning synthesis from the same initial pairs with μ widened to the policies' learning-start range without terminal starts (Appendix D.5) makes starts next to the ceiling harder: the learned bound then exceeds the best constant by more than 0.1 in six of ten seeds, against two of ten before. Where 2–11% of starts succeed less than 90% of the time, the constant falls to 0.01–0.24 while the learned mean stays at 0.58–0.89. Figure 2(c) shows why for one such seed: success drops only next to the ceiling, which pulls the constant down to 0.24, while both learned bounds stay above 0.85 elsewhere. In two seeds the Ratio-TD bound is 0.04 below the constant, because one calibration shift lowers its distant starts more; an MC learner, run afterwards, is above the constant in all nine seeds it certifies (0.97 against 0.61).

The gain tracks variation across starts over all 174 accepted batches (Figure 2(a,b); weak-initialization runs in Appendix D.4), whose empirical success spans 0.75–1.00. The gain is near zero when per-start success has standard deviation below 0.01, where the policy is nearly perfect and both bounds approach the local ceiling $\beta^{1/M}$, and averages 0.30 above 0.1.

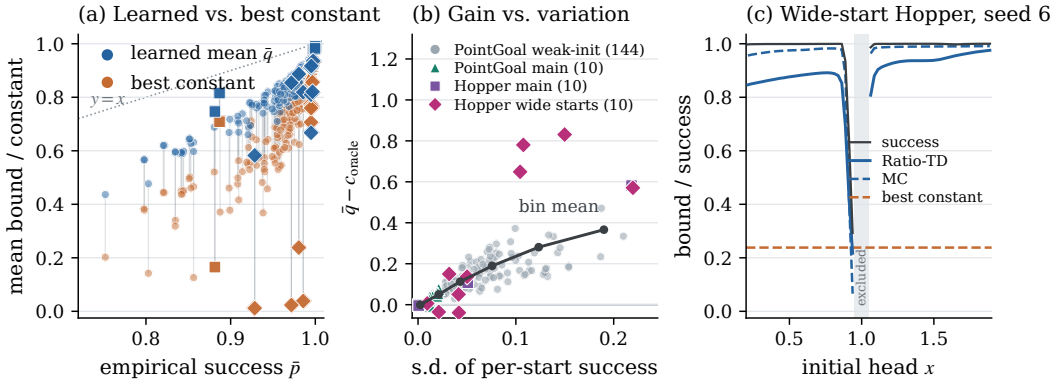

Figure 2. (a,b) Learned bound versus the best constant the same batch accepts, over 174 accepted batches (20 main runs; 10 wide-start Hopper runs with Ratio-TD; 144 weak-initialization rounds, including passing rounds not returned because those runs also require a mean bound of at least 0.9). (a) Vertical segments pair the two bounds of a batch; the dotted line is $y = x$. (b) Gain versus the standard deviation of per-start empirical success; the dark line gives bin means. Shapes mark groups; colors mark bound type in (a) and group in (b). Weak-initialization rounds are correlated within seeds. (c) Wide-start Hopper, seed 6: bin means over the initial head x of empirical success and of the Ratio-TD and MC bounds, with the best constant of the Ratio-TD batch; the shaded ceiling interval is excluded from μ.

Coverage within selected starting regions. Consider new starts in $A_\lambda = \{x : q(x) \geq \lambda\}$. Whenever the certificate holds ($V_H(\pi, q) \leq \epsilon$), every such region of positive mass satisfies $\Pr_{x\sim\mu(\cdot|A_\lambda)}\{p_H^\pi(x) \geq q(x) \geq \lambda\} \geq \max\{0, 1 - \epsilon/\mu(A_\lambda)\}$. For Hopper seed 2, 69,235 of 10^5 fresh initial states have $q \geq 0.8$, and a one-sided Dvoretzky–Kiefer–Wolfowitz–Massart band bounds the region mass below by 0.688; hence at least 98.5% of starts in this region have true success probability at least their predicted bound, itself at least 0.8. The band is uniform over thresholds, so this holds for every λ at once, with confidence at least 0.925 including the certificate's δ (Appendix A.5).

Table 1. Main results of complete synthesis with Feedback refinement, ten seeds per task. Certified: seeds that return a certified pair within ten rounds. Rounds, empirical success, learned mean bound $\bar{q}$, and best constant c_{oracle} are computed on the first accepted validation batch (mean $\pm$ s.d. over certified seeds). c_{oracle} is the largest constant the same batch accepts (Section 6.2). Wide μ: Hopper rerun from the same initial pairs with head x uniform on $[0.2, 1.9] \setminus [0.95, 1.05]$ (Appendix D.5).

Task	State / input	Certified	Rounds	Success	Learned $\bar{q}$	c_{oracle}
Linear	2 / 2	10/10	1.1	1.000 ± 0.000	0.991 ± 0.001	0.993 ± 0.000
Planar	3 / 3	10/10	1.0	0.999 ± 0.001	0.988 ± 0.004	0.986 ± 0.008
Hopper	12 / 13	10/10	1.3	0.976 ± 0.049	0.945 ± 0.088	0.878 ± 0.265
seeds 2 / 4					0.747 / 0.816	0.165 / 0.709
wide μ, Ratio-TD	12 / 13	10/10	2.8	0.984 ± 0.021	0.821 ± 0.114	0.511 ± 0.382
wide μ, MC	12 / 13	9/10	2.6	0.990 ± 0.006	0.966 ± 0.009	0.610 ± 0.261
PointGoal	6 / 44	10/10	1.1	0.995 ± 0.002	0.940 ± 0.011	0.909 ± 0.029

Table 2 reports average-success bounds for fixed thresholds: for the same region, the bound rises from 0.732 to 0.834 and empirical success from 0.881 to 0.996.

Table 2. Average-success bounds in selected regions, $A_\lambda = \{q \geq \lambda\}$. $\widehat{\mu}(A_\lambda)$ is the region mass on 10^5 fresh initial states per seed (no rollouts). The mean bound is $\max\{0, \bar{q}_{\text{lo}} - \epsilon/\mu_{\text{lo}}\}$, using one-sided Hoeffding lower limits at level 0.975 each for $\mathbb{E}[q \mid A_\lambda]$ and $\mu(A_\lambda)$ (Appendix A.5). Gates were fixed from the accepted validation batch before the fresh sample; combined confidence is at least 0.90. PointGoal averages ten per-seed bounds; success columns are empirical rates on the validation batch.

Task	**Gate** λ	$\widehat{\mu}(A_\lambda)$	**Mean bound**	**Succ. mean**	**Succ. min**	**Share** < 0.9
PointGoal (10 seeds)	none	1.000	0.925	0.995	0.51	0.0034
	0.90	0.905	0.932	0.997	0.60	0.0008
	0.925	0.776	0.934	0.998	0.69	0.0005
Hopper seed 2	none	1.000	0.732	0.881	0.16	0.258
	0.70	0.773	0.825	0.986	0.83	0.040
	0.80	0.692	0.834	0.996	0.94	0.000

The certified function also locates weak regions and scores new starts without rollouts: on Hopper seed 2, $q < 0.5$ on 14% of starts, with empirical success 0.16 at the worst sampled start. For one region fixed in advance, direct Monte Carlo is cheaper: at the observed success rate, 10^5 single rollouts from $\mu(\cdot \mid A_{0.8})$ and Markov's inequality would give at least 97.8% of starts with $p_H^\pi \geq 0.8$ at about one percent of our rollout cost, but no per-start bounds and no statement about other regions (Appendix B.6).

6.3 Q3: Calibration, Learning, and Cost

Two certification routes. On 40 accepted Feedback runs (ten per task), we recalibrate each frozen policy–candidate pair once on fresh data and compare the implemented shift of Section 5.2 with the exact order-statistic rule of Lemma 2 (Table 4). Mean bound heights differ by at most 0.003 per task, and both rules pass a fresh verifier batch in 37/40 cases, consistent with the learner-independent lower bound 0.92 of Lemma 2(ii) for the exact rule. The exact rule certifies from its calibration sample alone (Proposition 1), with 2.57 million rollouts per round versus 12.57 million for calibration plus validation. Exact calibration is thus the cheaper certificate for a fixed pipeline; validation costs about five times as much but accepts any candidate construction. **Role of the learner.** In fixed-policy comparisons, direct probability learners fail every test on PointGoal and Frozen Lake before calibration; after calibration, the learner families attain similar heights, 0.939–0.949 on PointGoal and 0.460–0.484 on Frozen Lake at full budget, and all 45 Frozen Lake calibration-budget runs have exact $V_H = 0$ (Appendix D.6). Starting from weak PointGoal policies and requiring a batch-mean bound of at least 0.9 before returning, Feedback succeeds in 7/10 seeds and Train in 5/10, with greater variability under Feedback (Appendix D.4). **Computational cost.** In the recalibration study above, a validation batch costs on average 7.5×10^7 simulated transitions on PointGoal and 8.3×10^8 on Hopper, counting steps until each rollout terminates. On one H20 GPU, complete synthesis takes about two minutes on Linear and Planar, 15–62 minutes on Hopper, and 48–77 minutes on PointGoal (per-run transition counts in Appendix D.9, sampling sweeps in Appendix D.8).

7 Discussion and Conclusion

RAPC learns a lower bound on reach-avoid success at each new start, with PAC coverage across adaptive learning rounds and, after calibration, a learner-independent guarantee on acceptance. The state-dependent bound improves on the best constant where success varies across starts, and for a fixed pipeline, exact calibration replaces validation at about one fifth of the certification rollouts. The main limitations are these: the guarantee concerns the simulated transition law and the chosen initial distribution, requires independent restarts, and permits an exceptional set of starts; deployment needs evidence linking them to the real system, and safety during training is not certified. The evidence covers these simulators, budgets, and one policy optimizer; the tested initial distributions vary mainly in a few coordinates (Appendix D.1), and the gain from state dependence appears only where success varies across starts and depends on the learner (Appendix D.5). The calibration study reuses accepted policies for one round, and sampling cost grows as ϵ decreases or the requested bound approaches one.

AI USE STATEMENT

Generative AI tools assisted with manuscript drafting and editing, mathematical derivations, pseudocode, LaTeX preparation, and preparation of tables and figures from experimental records.

ETHICS STATEMENT

This work studies probabilistic guarantees for simulated control systems. The guarantees allow an exceptional set of initial states and concern the transition law represented by the rollout interface. Their use in safety-critical applications requires assessing whether this law and the initial-state distribution represent deployment conditions. The method does not establish safety of the policy-learning process or account for harms outside the specified target and failure sets.

REPRODUCIBILITY STATEMENT

Code for all experiments, including the environments, the certification procedure, the analysis scripts, and the figure scripts, is available at `https://anonymous.4open.science/r/iclr-pac-BBBD/`; its README gives the commands that reproduce the main runs in one container. The problem and the sampling assumptions are specified in Section 3 and Assumption 1, and the guarantee in Theorem 1. Algorithm 1 and Section 5 describe synthesis and candidate learning. Appendices A and B contain proofs, including the limits of the quantile-based auxiliary construction. Appendix C gives the binomial-tail implementation, confidence thresholds, calibration protocol, and exact-reference tests. Appendix D specifies experimental settings, complete candidate comparisons, seed results, sensitivity analyses, and transition accounting.

AUTHOR CONTRIBUTIONS

Author contributions are omitted for anonymous review.

ACKNOWLEDGMENTS

Acknowledgments are omitted for anonymous review.

REFERENCES

Alessandro Abate, Maria Prandini, John Lygeros, and Shankar Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008.

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

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2021. URL `https://arxiv.org/abs/2110.01052`.

Thom Badings, Wietze Koops, Sebastian Junges, and Nils Jansen. Policy verification in stochastic dynamical systems using logarithmic neural certificates. In *International Conference on Computer Aided Verification*, pp. 349–375. Springer, 2025.

Somil Bansal and Claire J Tomlin. Deepreach: A deep learning approach to high-dimensional reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1817–1824. IEEE, 2021.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2242–2253. IEEE, 2017.

Rina Foygel Barber. Is distribution-free inference possible for binary regression? *Electronic Journal of Statistics*, 14(2):3487–3524, 2020.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021.

Luca Bortolussi, Dimitrios Milios, and Guido Sanguinetti. Smoothed model checking for uncertain continuous time Markov chains. *Information and Computation*, 247:235–253, 2016.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. Jax: Autograd and xla. *Astrophysics Source Code Library*, pp. ascl–2111, 2021.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Giuseppe Carlo Calafiore and Marco C Campi. The scenario approach to robust control design. *IEEE Transactions on automatic control*, 51(5):742–753, 2006.

Marco C Campi and Simone Garatti. A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *Journal of optimization theory and applications*, 148(2): 257–280, 2011.

Zhipeng Cao, Peixin Wang, Luke Ong, Đorđe Žikelić, Dominik Wagner, and Bai Xue. Comparative analysis of barrier-like function methods for reach-avoid verification in stochastic discrete-time systems. *arXiv preprint arXiv:2512.05348*, 2025.

Yu Chen, Yuda Li, Shaoyuan Li, and Xiang Yin. Distributionally robust control synthesis for stochastic systems with safety and reach-avoid specifications. *arXiv preprint arXiv:2501.03137*, 2025.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/biomet/26.4.404.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Jaime F Fisac, Neil F Lugovoy, Vicenç Rubies-Royo, Shromona Ghosh, and Claire J Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8550–8556. IEEE, 2019.

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

Milan Ganai, Zheng Gong, Chenning Yu, Sylvia Herbert, and Sicun Gao. Iterative reachability estimation for safe reinforcement learning. *Advances in Neural Information Processing Systems*, 36:69764–69797, 2023.

Kai-Chieh Hsu, Vicenç Rubies-Royo, Claire J Tomlin, and Jaime F Fisac. Safety and liveness guarantees through reach-avoid reinforcement learning. *arXiv preprint arXiv:2112.12288*, 2021.

Roger Koenker and Gilbert Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978. doi: 10.2307/1913643.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523): 1094–1111, 2018.

Albert Lin and Somil Bansal. Verification of neural reachable tubes via scenario optimization and conformal prediction. In *6th Annual Learning for Dynamics & Control Conference*, pp. 719–731. PMLR, 2024.

Lars Lindemann, Yiqi Zhao, Xinyi Yu, George J. Pappas, and Jyotirmoy V. Deshmukh. Formal verification and control with conformal prediction. *arXiv preprint arXiv:2409.00536*, 2024. URL https://arxiv.org/abs/2409.00536.

Haitong Ma, Yang Guan, Shegnbo Eben Li, Xiangteng Zhang, Sifa Zheng, and Jianyu Chen. Feasible actor-critic: Constrained reinforcement learning for ensuring statewise safety. *arXiv preprint arXiv:2105.10682*, 2021.

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

Kostas Margellos and John Lygeros. Hamilton–jacobi formulation for reach–avoid differential games. *IEEE Transactions on automatic control*, 56(8):1849–1861, 2011.

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.

Yiming Meng and Jun Liu. Lyapunov-barrier characterization of robust reach–avoid–stay specifications for hybrid systems. *Nonlinear Analysis: Hybrid Systems*, 49:101340, 2023.

Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control*, 50(7):947–957, 2005.

Mahdi Nazeri, Thom Badings, Sadegh Soudjani, and Alessandro Abate. Data-driven yet formal policy synthesis for stochastic nonlinear dynamical systems. In *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*, volume 283 of *Proceedings of Machine Learning Research*, pp. 1550–1564. PMLR, 2025. URL https://proceedings.mlr.press/v283/nazeri25a.html.

Ameneh Nejati and Majid Zamani. Data-driven synthesis of safety controllers via multiple control barrier certificates. *IEEE Control Systems Letters*, 7:2497–2502, 2023.

Ameneh Nejati, Abolfazl Lavaei, Pushpak Jagtap, Sadegh Soudjani, and Majid Zamani. Formal verification of unknown discrete-and continuous-time systems: A data-driven approach. *IEEE Transactions on Automatic Control*, 68(5):3011–3024, 2023.

Tingting Ni and Maryam Kamgarpour. A learning-based approach to stochastic optimal control under reach-avoid constraint. In *Proceedings of the 28th ACM International Conference on Hybrid Systems: Computation and Control*, pp. 1–8, 2025.

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://arxiv.org/abs/2001.00106.

Stephen Prajna, Ali Jadbabaie, and George J Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8): 1415–1428, 2007.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. OpenAI, 2019. URL https://cdn.openai.com/safexp-short.pdf.

Spencer M Richards, Jean-Jacques Slotine, Navid Azizan, and Marco Pavone. Learning control-oriented dynamical structure from data. In *International Conference on Machine Learning*, pp. 29051–29062. PMLR, 2023.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on robot learning*, pp. 91–100. PMLR, 2022.

Ali Salamati and Majid Zamani. Safety verification of stochastic systems: A repetitive scenario approach. *IEEE Control Systems Letters*, 7:448–453, 2022.

Ali Salamati, Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Data-driven verification and synthesis of stochastic systems via barrier certificates. *Automatica*, 159:111323, 2024.

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

Sean Summers and John Lygeros. Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Automatica*, 46(12):1951–1961, 2010.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Joseph A. Vincent, Aaron O. Feldman, and Mac Schwager. Guarantees on robot system performance using stochastic simulation rollouts. *IEEE Transactions on Robotics*, 40, 2024.

Abraham P Vinod and Meeko MK Oishi. Stochastic reachability of a target tube: Theory and computation. *Automatica*, 125:109458, 2021.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Proceedings of the Asian Conference on Machine Learning*, volume 25 of *Proceedings of Machine Learning Research*, pp. 475–490, 2012.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Bai Xue. Finite-time safety and reach-avoid verification of stochastic discrete-time systems. *Information and Computation*, pp. 105368, 2025.

Bai Xue. Sufficient and necessary barrier-like conditions for safety and reach-avoid verification of stochastic discrete-time systems. *Automatica*, 187:112919, 2026.

Bai Xue, Renjue Li, Naijun Zhan, and Martin Fränzle. Reach-avoid analysis for stochastic discrete-time systems. In *2021 American Control Conference (ACC)*, pp. 4879–4885. IEEE, 2021.

Long Yang, Jiaming Ji, Juntao Dai, Yu Zhang, Pengfei Li, and Gang Pan. Cup: A conservative update policy algorithm for safe reinforcement learning. *arXiv preprint arXiv:2202.07565*, 2022.

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*, 2020.

Håkan L. S. Younes. Error control for probabilistic model checking. In *Verification, Model Checking, and Abstract Interpretation*, pp. 142–156. Springer, 2006.

Håkan L. S. Younes and Reid G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *Computer Aided Verification*, pp. 223–235. Springer, 2002.

Håkan L. S. Younes and Reid G. Simmons. Statistical probabilistic model checking with a focus on time-bounded properties. *Information and Computation*, 204(9):1368–1409, 2006. doi: 10.1016/j.ic.2006.05.002.

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

Ruikun Zhou, Thanin Quartz, Hans De Sterck, and Jun Liu. Neural lyapunov control of unknown nonlinear systems with stability guarantees. *Advances in Neural Information Processing Systems*, 35:29113–29125, 2022.

Đorđe Žikelić, Mathias Lechner, Thomas A Henzinger, and Krishnendu Chatterjee. Learning control policies for stochastic systems with reach-avoid guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11926–11935, 2023.

A PROOFS AND INTERPRETATION OF THE GUARANTEE

Table 3 lists the notation of the main text.

Table 3. Notation.

Symbol	Meaning
μ; $\mathcal{T}$, $\mathcal{F}$; H	initial-state distribution; target and failure sets; horizon
$p_H^\pi(x)$	probability of reaching $\mathcal{T}$ before $\mathcal{F}$ within H steps from x under π, eq. (1)
$q(x)$	candidate lower bound on $p_H^\pi(x)$
$V_H(\pi, q)$	violation mass $\Pr_{x\sim\mu}\{q(x) > p_H^\pi(x)\}$, eq. (2)
ϵ, δ	tolerated violation mass; total probability of a false return
N, M	starts per validation batch; rollouts per start
S_x	number of successes among the M rollouts from x
β, $L_\beta(s; M)$	error level of the local check; one-sided Clopper–Pearson lower limit, eq. (6)
a, K_N	$a = (1-\beta)\epsilon$; number of rejected starts in a batch
$B_{N,a}(k)$	binomial lower tail, eq. (8)
R, δ_t, k_*	maximum rounds; error level of round t; acceptance cutoff $\max\{k : B_{N,a}(k) \le \delta_t\}$
z, σ, $\widetilde{\rho}$	learned logit; logistic function; candidate denominator (Section 5.1)
n, α, c, $e_{(k)}$	shift-calibration states; tolerated local rejection rate; shift; k-th smallest residual
$\bar{q}$, c_{oracle}	mean of q over a validation batch; largest constant the same batch accepts
A_λ	region $\{x : q(x) \ge \lambda\}$ of starts with predicted bound at least λ

A.1 THE LOCAL CONFIDENCE LIMIT

For $s \in \{1, \dots, M\}$, define $T_s(u) = \Pr_{\text{Bin}(M,u)}(S \ge s)$. This function is continuous and strictly increasing from zero to one on $u \in [0, 1]$, so the root in (6) exists and is unique. Moreover, $L_\beta(s; M)$ is nondecreasing in s.

Proof of Lemma 1. Fix $p \in (0, 1)$. If there is no success count s with $L_\beta(s; M) > p$, the event has probability zero. Otherwise, let s_* be the smallest such count. It is at least one because $L_\beta(0; M) = 0$. By monotonicity, the event is $\{S \ge s_*\}$. Since $p < L_\beta(s_*; M)$,

$$\Pr_p\{L_\beta(S; M) > p\} = T_{s_*}(p) < T_{s_*}(L_\beta(s_*; M)) = \beta.$$

At $p = 0$, $S = 0$ almost surely; at $p = 1$, $L_\beta(S; M) \le 1$ always. Both endpoint cases have zero error. If $q > p$, the event $\{L_\beta(S; M) \ge q\}$ is contained in $\{L_\beta(S; M) > p\}$, which proves the false-support bound. □

For $s \ge 1$, $L_\beta(s; M)$ equals the β-quantile of a $\text{Beta}(s, M-s+1)$ distribution; this is a computational expression, and the guarantee is repeated-sampling coverage.

A.2 FROM REJECTION PROBABILITY TO VIOLATION MASS

For a fixed pair, let $\mathcal{B} = \{x : q(x) > p_H^\pi(x)\}$. At each $x \in \mathcal{B}$, Lemma 1 gives

$$\Pr\{Z = 1 \mid x\} = 1 - \Pr\{L_\beta(S_x; M) \ge q(x) \mid x\} \ge 1 - \beta.$$

Using nonnegativity of the rejection probability on $\mathcal{X} \setminus \mathcal{B}$,

$$r = \int_{\mathcal{X}} \Pr\{Z = 1 \mid x\}\mu(dx) \ge (1-\beta)\mu(\mathcal{B}) = (1-\beta)V_H(\pi, q).$$

For a fixed candidate, if Q_x^π denotes the law of one horizon-H rollout from x, the joint law of a validation scenario is

$$\mathbb{P}_{\text{scen}}(dx, d\mathcal{R}) = \mu(dx)\,(\mathsf{Q}_x^\pi)^{\otimes M}(d\mathcal{R}).$$

The conditional rollout law generally depends on x; it is not an independent product with an unconditional trajectory law. Although r is a probability under this joint law, $\mu(\mathcal{B})$ concerns only the initial state and its true success probability. In particular,

$$\mathbb{P}_{x\sim\mu}\{\mathbb{P}_x^\pi(\phi_H) \ge q(x)\} = 1 - V_H(\pi, q) \ge 1 - \frac{r}{1-\beta}.$$

Substituting $r \le a$ yields

$$r \le a \quad \Longrightarrow \quad \mathbb{P}_{x\sim\mu}\big\{\mathbb{P}^{\pi}_{x}(\phi_H) \ge q(x)\big\} \ge 1 - \frac{a}{1-\beta}. \tag{13}$$

Taking $a = (1-\beta)\epsilon$ gives (4). The confidence supplied by the statistical test concerns all validation data, as formalized below.

This implication is one-sided. A valid candidate can generate many rejections because its values are close to the true probabilities or its rollout budget is insufficient. Conversely, a local rejection does not identify a state as a member of $\mathcal{B}$.

A.3 One Round and Adaptive Repetition

For fixed N, a, δ_t, define

$$k_* = \max\{k \in \{0,\dots,N\} : B_{N,a}(k) \le \delta_t\},$$

with $k_* = -1$ if the set is empty. The acceptance event is exactly $\{K_N \le k_*\}$. If $k_* = -1$, acceptance is impossible. Otherwise, for any $r \ge a$, stochastic monotonicity of the binomial distribution gives

$$\Pr_{\mathrm{Bin}(N,r)}\{K_N \le k_*\} = B_{N,r}(k_*) \le B_{N,a}(k_*) \le \delta_t.$$

For example, the monotonicity follows by coupling Bernoulli draws as $\mathbf{1}\{U_i \le r\}$ with common independent $U_i \sim \mathrm{Unif}[0,1]$.

Proof of Theorem 1. Let $\mathcal{H}_{t-1}$ contain all information available before validation at round t, including learning randomness, all previous validation outcomes, the current candidate, and the selected sampling parameters. Conditional on $\mathcal{H}_{t-1}$ and on reaching that round, the candidate is fixed. By Assumption 1, the state-and-rollout blocks are independent and identically distributed. Their rejection indicators are therefore iid Bernoulli with some conditional parameter r_t.

The event $V_H(\pi_t, q_t) > \epsilon$ is determined by the candidate before the batch. On this event, $r_t \ge (1-\beta_t)V_H(\pi_t, q_t) > (1-\beta_t)\epsilon$. The preceding binomial argument bounds conditional acceptance probability by δ_t. If E_t denotes accepting an invalid pair at that round, including the requirement that the round is reached and its batch completed, then

$$\Pr(E_t \mid \mathcal{H}_{t-1}) \le \delta_t\, \mathbf{1}\{\text{round } t \text{ is reached}\}.$$

A union bound, the tower property, and the assumed total allocation give

$$\Pr\Big(\bigcup_t E_t\Big) \le \sum_t \mathbb{E}[\Pr(E_t \mid \mathcal{H}_{t-1})] \le \mathbb{E}\Big[\sum_t \delta_t\Big] \le \delta.$$

For countably many rounds the same argument follows by monotone convergence. Outside $\bigcup_t E_t$, every accepted pair has $V_H(\pi_t, q_t) \le \epsilon$. Since

$$1 - V_H(\pi_t, q_t) = \mathbb{P}_{x\sim\mu}\{\mathbb{P}^{\pi_t}_{x}(\phi_H) \ge q_t(x)\},$$

this is precisely (4), simultaneously for all accepted rounds on an event of probability at least $1-\delta$. Stopping after an acceptance or returning UNKNOWN can only reduce the error event. □

The theorem does not justify changing π_t, q_t, N_t, M_t, β_t, or the acceptance level in response to the current validation outcomes and then using the same fixed-batch test. It also does not bound $\Pr(V_H > \epsilon \mid \text{acceptance})$: conditioning on a potentially rare acceptance event changes the probability statement.

A.4 Calibration Controls the Local Rejection Rate

This section proves Lemma 2, conditional on the history before shift calibration. Fix the policy π, the finite measurable logit z, the measurable denominator $\widetilde{\rho} : \mathcal{X} \to (0,1]$, and the calibration parameters. For the corollary also fix N, ϵ, δ_t at this stage. Calibration blocks are iid conditional on this history, and validation uses a separate fresh batch. For a state–count pair (x, S) define the residual

$$e(x,S) = z(x) - \mathrm{logit}\,\big(\widetilde{\rho}(x) L_\beta(S; M)\big),$$

with $\mathrm{logit}(0) = -\infty$, so that $e = +\infty$ when $S = 0$. Since $L_\beta(S; M) \le \beta^{1/M} < 1$, the argument of the logit lies in $[0, 1)$.

Lemma 3 (Rejection is a residual exceedance)**.** *For every $c \in (-\infty, \infty]$ and every pair (x, S), with q_c as in* (11)*, $L_\beta(S; M) < q_c(x)$ holds if and only if $e(x, S) > c$.*

Proof. If $c = \infty$, then $q_c \equiv 0$ and both events are false. For finite c, write $L = L_\beta(S; M)$. If $L = 0$, then $e = \infty$ and $q_c(x) > 0$, so both events hold. Otherwise $0 < L < 1$, and

$$L < \min\{\sigma(z(x) - c)/\widetilde{\rho}(x), 1\} \iff \widetilde{\rho}(x)L < \sigma(z(x) - c) \iff e(x, S) > c,$$

by strict monotonicity of the logit on $(0, 1)$. □

Proof of Lemma 2(i). If $k > n$, then $c = \infty$, $q_c \equiv 0$, and $r_c = 0$, proving the claims. Suppose $k \le n$. Conditional on the pre-calibration history, the iid pairs in the lemma give exchangeable extended-real residuals $e_1, \ldots, e_n, e$. Let $c = e_{(k)}$. Break ties uniformly at random; the rank of e among all $n + 1$ residuals is then uniform on $\{1, \ldots, n + 1\}$, and $e > e_{(k)}$ requires this rank to exceed k. Hence

$$\Pr\{e > c\} \le 1 - \frac{k}{n+1} \le \alpha,$$

the split-conformal bound (Vovk et al., 2005; Lei et al., 2018). Lemma 3 identifies $\{e > c\}$ with the local rejection event, which proves (12). Conditional on the calibration pairs, c is fixed and $r_c = \Pr\{e > c \mid c\}$; the tower property gives $\mathbb{E}[r_c] \le \alpha$. If the residual distribution function F is continuous, then $r_c = 1 - F(e_{(k)})$, and $F(e_{(k)})$ is the k-th order statistic of n uniform variables, so $r_c \sim \mathrm{Beta}(n + 1 - k, k)$ (Vovk, 2012). A larger shift $c' \ge c$ only removes rejections, so the rejection-probability bound also holds for any shift that dominates the conformal quantile. □

Combining (12) with (7) gives $\mathbb{E}[V_H(\pi, q_c)] \le \mathbb{E}[r_c]/(1-\beta) \le \alpha/(1-\beta)$, an average statement over the calibration draw. This marginal bound alone does not give the stated procedure-level confidence. That confidence follows either from fresh validation (Theorem 1) or from the training-conditional tail bound below, with a confidence budget across rounds.

Proof of Lemma 2(ii). Conditional on the calibration pairs, (π, q_c) is fixed before the validation batch, so Assumption 1 makes the N rejection indicators iid Bernoulli with parameter r_c and $K_N \sim \mathrm{Bin}(N, r_c)$. Passing the complete batch is $\{K_N \le k_*\}$, so its conditional probability is $B_{N,r_c}(k_*)$, and averaging over calibration gives the stated expectation. If $k > n$, then $r_c = W = 0$, proving the bound, including the case $k_* = -1$ where passing is impossible.

For $k \le n$, couple $e_i = F^{-1}(U_i)$ with U_i iid uniform on $(0, 1)$, where F is the residual distribution function, extended by $F(\infty) = 1$, and F^{-1} is its generalized inverse (equal to ∞ when no finite quantile exists). Then $c = F^{-1}(U_{(k)})$ and

$$r_c = 1 - F(F^{-1}(U_{(k)})) \le 1 - U_{(k)} =: W, \qquad W \sim \mathrm{Beta}(n + 1 - k, k).$$

The inequality follows from $F(F^{-1}(u)) \ge u$ and permits atoms, including at ∞. Since N and k_* were fixed before calibration and $B_{N,r}(k_*)$ is nonincreasing in r, taking expectations proves the lower bound. For continuous residuals, $r_c = W$ under this coupling. A computation that stops before completing the batch need not return a certificate with this probability. □

Calibration alone as a certificate. The same coupling yields a training-conditional bound (Vovk, 2012, Proposition 2b). For $k \le n$ and $a = (1 - \beta)\epsilon$, the risk link and $r_c \le 1 - U_{(k)}$ give

$$\Pr\{V_H(\pi, q_c) > \epsilon\} \le \Pr\{r_c > a\} \le \Pr\{U_{(k)} < 1 - a\} = B_{n,a}(n - k).$$

The equality counts the uniform draws above $1 - a$; continuity of the uniforms, rather than of the residuals, justifies the binomial identity. If $k > n$, then $V_H(\pi, q_c) = 0$ and the bound is zero by $B_{n,a}(-1) = 0$. For $n = 2048$, $k = 2039$, and $a = 0.00999$, the bound is $0.0037 < \delta_t = 0.005$. Across adaptive rounds, choose the candidate learner, calibration parameters, and error allocation before each fresh calibration batch. If each tail bound is at most its allocation and the allocations sum to at most δ, the conditional union-bound argument of Theorem 1 applies. Without target clipping, the implemented interpolation dominates rank 2037, giving the weaker upper bound $B_{2048,a}(11) = 0.0166$. Target clipping breaks Lemma 3; independent validation instead certifies the resulting candidate at the cost of an additional NM rollouts per round.

Empirical comparison of the two routes. Table 4 pairs the two rules on every accepted Feedback (C) run. The evaluation draws its fitting and shift states from μ and uses compact rollout batches for all four tasks, including PointGoal, whose synthesis runs use scans (Appendix D.9). The exact shift is never smaller than the implemented one and the two differ by at most 0.1 in logit space. Empirically, the mean calibrated bounds differ by at most 0.003 per task (0.008 on a single run); the exact rule produces 4–8 fewer local rejections per $10{,}000$ states on the fresh batches. On these policies no conformal state had a confidence limit below 10^{-3}, so neither the target clipping nor the $+\infty$ residual was active and the two rules differ only in the rank of the quantile. In general, the exact shift is infinite if and only if at least $n - k + 1$ calibration blocks have zero successes; it then returns $q_c \equiv 0$. Here this requires at least ten such blocks. Evaluated post hoc on the stored validation batches, which were selected by first acceptance and therefore do not estimate an acceptance rate, the two recalibrated candidates pass in 35 and 36 of 40 cases. On one fresh validation batch per run, drawn after both calibrations, both pass in 37 of 40 (0.925); the exact rule has the acceptance lower bound 0.92 of Lemma 2(ii). Per round, calibration uses $8192 \times 64 + 2048 \times 1000 = 2.6 \times 10^6$ rollouts against $NM = 10^7$ for validation, a ratio of 3.9 on every task; policy training is common to both routes.

Table 4. Direct PAC calibration versus calibrate-then-verify, paired on the accepted Feedback (C) runs (ten seeds per task). Each frozen policy and candidate is recalibrated once on fresh calibration data; from the same fitted network we take the implemented shift (interpolated quantile, clipped targets) and the exact shift of Lemma 2 (rank $\lceil (n+1)(1-\alpha) \rceil$, unclipped CP limit, residual $+\infty$ at zero-success states), and evaluate both on one fresh validation batch drawn after both calibrations. K is the number of local rejections out of $10{,}000$ ($k_* = 74$); Acc. counts accepted runs. Transitions are per round: fresh calibration versus the validation batch of the accepted round.

	Implemented shift			**Exact shift (Lemma 2)**			**Transitions per round**	
Task	$\bar{q}$	K	Acc.	$\bar{q}$	K	Acc.	Calibration	Validation
PointGoal	0.945	61.2	8/10	0.942	54.5	8/10	1.9×10^7	7.5×10^7
Hopper	0.952	44.6	10/10	0.952	39.9	10/10	2.1×10^8	8.3×10^8
Linear	0.991	46.1	9/10	0.991	39.7	9/10	1.0×10^7	4.0×10^7
Planar	0.988	55.8	10/10	0.988	48.1	10/10	8.7×10^6	3.4×10^7

Implementation deviations. The implementation takes the level-$(1-\alpha)$ empirical quantile with linear interpolation and applies its positive part (Appendix C.3). With $u = (n-1)(1-\alpha)$, the interpolated value lies between ranks $\lfloor u \rfloor + 1$ and $\lfloor u \rfloor + 2$. For $n = 2048$ and $\alpha = 0.005$ these are ranks 2037 and 2038, whereas the exact rule uses 2039. Applied to unclipped residuals, a shift at least as large as rank 2037 gives $\mathbb{E}[r_c] \le 1 - 2037/2049 \simeq 0.0059$. Taking the positive part can only reduce rejections. At $N = 10^4$, the Beta-mixture acceptance lower bounds for ranks 2039 and 2037 are respectively 0.924 and 0.814 at cutoff 74, and 0.966 and 0.903 at cutoff 83. For Frozen Lake, the exact rank 1000 gives 0.9995 ($n = 1024$, $\alpha = 0.025$, cutoff 443).

These calculations concern unclipped residuals. The implementation also clips calibration targets at 10^{-3}, making a zero-success residual finite rather than $+\infty$. Its candidate then stays positive where $L = 0$ and is rejected there. Consequently, the foregoing calibration-based bounds do not certify this implementation; its certificate comes from fresh validation. The observed acceptance counts in Appendix D.6 (78/80 on PointGoal and 45/45 on Frozen Lake) are empirical results. For Hopper seed 2, many zero-success validation blocks explain the rejection of the calibration-only synthesis attempt (Section 6.1); whether exact calibration returns the zero function depends on the calibration counts through the criterion above.

A.5 Informativeness and Distribution Changes

To establish (5), observe that $q - p_H^\pi \le 0$ outside $\mathcal{B}$, and $q - p_H^\pi \le 1$ on $\mathcal{B}$. Integrating yields $\mathbb{E}_\mu[q] - \mathbb{E}_\mu[p_H^\pi] \le \mu(\mathcal{B}) \le \epsilon$. The nonnegative lower bound follows because $p_H^\pi \ge 0$.

A violation-mass guarantee need not remain equally informative under conditioning. For a measurable set A of positive μ mass,

$$\Pr_{x \sim \mu(\cdot|A)} \{q(x) > p_H^\pi(x)\} = \frac{\mu(\mathcal{B} \cap A)}{\mu(A)} \le \min\{1, \epsilon/\mu(A)\}.$$

For $A_\lambda = \{x : q(x) \geq \lambda\}$, every state outside $\mathcal{B}$ in this region has $p_H^\pi(x) \geq q(x) \geq \lambda$. Consequently,

$$\Pr_{x \sim \mu(\cdot|A_\lambda)}\{p_H^\pi(x) \geq q(x) \geq \lambda\} \geq \max\{0, 1 - \epsilon/\mu(A_\lambda)\}. \tag{14}$$

Integrating $q - p_H^\pi \leq \mathbf{1}_\mathcal{B}$ on A also gives

$$\mathbb{E}[p_H^\pi \mid A] \geq \max\{0, \mathbb{E}[q \mid A] - \epsilon/\mu(A)\}. \tag{15}$$

These population inequalities hold simultaneously for every measurable A of positive mass on the single event $V_H \leq \epsilon$. Likewise, for every $\lambda \in [0, 1]$,

$$\mu\{x : p_H^\pi(x) \geq \lambda\} \geq \mu\{x : q(x) \geq \lambda\} - \epsilon,$$

since $\{q \geq \lambda\} \setminus \mathcal{B} \subseteq \{p_H^\pi \geq \lambda\}$. The finite-sample estimates below require separate confidence control: we use fixed-threshold bounds for average success and a uniform mass band for coverage across thresholds. A small selected region may consist entirely of violating states.

Estimating a gated bound from fresh states. Fix an accepted pair and a threshold λ before drawing J fresh iid states $X_1, \ldots, X_J \sim \mu$, independently of the learning and validation data. Set $A = \{x : q(x) \geq \lambda\}$, $m = \sum_{i=1}^J \mathbf{1}_A(X_i)$, $\widehat{\mu} = m/J$, and, for $m > 0$, $\bar{q}_A = m^{-1} \sum_{i:X_i \in A} q(X_i)$. For $\eta_\mu, \eta_q \in (0, 1)$, define

$$\mu_{\mathrm{lo}} = \max\left\{0, \widehat{\mu} - \sqrt{\frac{\log(1/\eta_\mu)}{2J}}\right\}, \qquad \bar{q}_{\mathrm{lo}} = \max\left\{0, \bar{q}_A - \sqrt{\frac{\log(1/\eta_q)}{2m}}\right\},$$

with $\bar{q}_{\mathrm{lo}} = 0$ when $m = 0$. Hoeffding's inequality gives $\Pr\{\mu_{\mathrm{lo}} > \mu(A)\} \leq \eta_\mu$. For $\mu(A) > 0$, conditional on any $m > 0$, the retained states are iid with law $\mu(\cdot \mid A)$; applying Hoeffding to their bounded values and averaging over m gives $\Pr\{\bar{q}_{\mathrm{lo}} > \mathbb{E}[q \mid A]\} \leq \eta_q$. Thus, with probability at least $1 - \eta_\mu - \eta_q$, both lower limits hold. Combining them with (15), whenever $V_H \leq \epsilon$ and $\mu_{\mathrm{lo}} > 0$, yields

$$\mathbb{E}[p_H^\pi \mid A] \geq \max\{0, \bar{q}_{\mathrm{lo}} - \epsilon/\mu_{\mathrm{lo}}\}.$$

If $\mu_{\mathrm{lo}} = 0$, report only the trivial bound zero; if $\mu(A) = 0$, no conditional claim is made. Taking $\eta_\mu = \eta_q = 0.025$ gives the fresh-sample confidence 0.95 used in Table 2. A union bound with the procedure's error event gives total error at most $\delta + 0.05$ for this accepted output and threshold. The reported PointGoal averages summarize ten per-seed bounds; these confidence levels apply per policy and fixed threshold, not simultaneously across table entries or to thresholds chosen using the estimation sample.

Estimating coverage simultaneously across thresholds. Fix the returned policy and function before drawing the fresh iid state sample, but allow the threshold to be selected afterward. For $\eta_\mu \in (0, 1/2]$, define

$$\widehat{\mu}(\lambda) = \frac{1}{J}\sum_{i=1}^J \mathbf{1}\{q(X_i) \geq \lambda\}, \qquad \mu_{\mathrm{lo}}(\lambda) = \max\left\{0, \widehat{\mu}(\lambda) - \sqrt{\frac{\log(1/\eta_\mu)}{2J}}\right\}.$$

Conditionally on the history before this sample, the one-sided DKW–Massart inequality (Massart, 1990) gives

$$\Pr\{\exists \lambda \in [0, 1] : \mu_{\mathrm{lo}}(\lambda) > \mu(A_\lambda)\} \leq \eta_\mu.$$

Indeed, apply the inequality to the empirical distribution function of $-q(X_i)$ at $-\lambda$; its value is $\widehat{\mu}(\lambda)$, including when the score distribution has atoms. The condition $\eta_\mu \leq 1/2$ ensures the stated one-sided Massart bound applies. The numerical lower limit at each threshold is unchanged from the preceding Hoeffding formula.

On this uniform mass event and the function certificate's validity event, (14) yields coverage at least $\max\{0, 1 - \epsilon/\mu_{\mathrm{lo}}(\lambda)\}$ simultaneously at every threshold with $\mu_{\mathrm{lo}}(\lambda) > 0$. Thus the probability of any false coverage report across thresholds is at most $\delta + \eta_\mu$. This controls erroneous reports across the complete procedure, including runs returning UNKNOWN; it is not a conditional-on-acceptance error bound. The uniform statement concerns thresholds of this fixed function. No estimate of $\mathbb{E}[q \mid A]$ is needed, and the average-success estimates above retain their fixed-threshold interpretation.

For Hopper seed 2 at $\lambda = 0.8$, the stored fresh-state sample has $J = 100{,}000$ and $m = 69{,}235$. At $\eta_\mu = 0.025$, $\mu_{\mathrm{lo}}(0.8) = 0.6880553\ldots$ and $1 - \epsilon/\mu_{\mathrm{lo}}(0.8) = 0.9854662\ldots$. The 98.5% coverage

statement is therefore one member of a family of coverage bounds with simultaneous procedure-level confidence at least 0.925 across thresholds for this fixed policy and function. This is coverage over a new random start in the selected region; it does not identify the exceptional starts or certify every specified start individually.

The original guarantee likewise says nothing about a different transition kernel. Extending it to another initial distribution or a physical system requires additional, justified assumptions relating those distributions or kernels.

B Finite-Horizon Learning and Comparison of Guarantees

B.1 The Ratio-TD Candidate

Discounted value learning (Sutton et al., 1998) weights both success and the time taken to succeed. Fix policy π, discount $\gamma \in (0,1)$, and remaining horizon $h \in \{0,\dots,H\}$, with $\gamma^{\infty} = 0$. Let $A_h = \{\tau_{\mathcal{T}} \le h, \tau_{\mathcal{T}} < \tau_{\mathcal{F}}\}$ denote success, and define

$$v_h^{\pi}(x) = \mathbb{E}_x^{\pi}[\gamma^{\tau_{\mathcal{T}}} \mathbf{1}_{A_h}], \qquad \rho_h^{\pi}(x) = \begin{cases} \mathbb{E}_x^{\pi}[\gamma^{\tau_{\mathcal{T}}} \mid A_h], & p_h^{\pi}(x) > 0, \\ 0, & p_h^{\pi}(x) = 0. \end{cases} \tag{16}$$

Thus v_h^{π} averages discounted successes, and ρ_h^{π} averages the discount conditional on success. With $\mathcal{D} = \mathcal{X} \setminus (\mathcal{T} \cup \mathcal{F})$, the value recursion is

$$v_0^{\pi} = \mathbf{1}_{\mathcal{T}}, \qquad v_h^{\pi}(x) = \mathbf{1}_{\mathcal{T}}(x) + \gamma \mathbf{1}_{\mathcal{D}}(x) \int v_{h-1}^{\pi}(x') \mathsf{P}^{\pi}(dx' \mid x). \tag{17}$$

Proposition 2 (Discounted hitting-time decomposition)**.** *For each h, x, $v_h^{\pi}(x) = \rho_h^{\pi}(x) p_h^{\pi}(x)$. If $p_h^{\pi}(x) > 0$, then $\gamma^h \le \rho_h^{\pi}(x) \le 1$ and $p_h^{\pi}(x) = v_h^{\pi}(x)/\rho_h^{\pi}(x)$. Otherwise, $v_h^{\pi}(x) = \rho_h^{\pi}(x) = 0$. The terminal values and* (17) *uniquely specify v_h^{π} by backward induction.*

When both components are exact, dividing v by ρ removes the time weighting and recovers success probability. Probability-TD instead sets the discount to one in (17). Both learned versions have approximation errors, so both need calibration and verification.

Implemented candidate. We fit the value logit of $v_\eta(x,h)$ by cross-entropy against the bootstrapped TD target of (17), computed from a Polyak-averaged target network with no gradients through it. The horizon is an input; terminal values are fixed. Successful trajectories also label visited states with the remaining time $D \le h$ to reach the target. Regressing γ^D with squared loss would estimate the conditional mean ρ_h^{π}. The implemented variant instead predicts the 10th percentile of successful hitting times with pinball loss (Koenker & Bassett, 1978) at $\tau = 0.1$, and forms

$$\widehat{\rho}_\tau(x,h) = \gamma^{T_{\xi,\tau}(x,h)}, \qquad q_h(x) = \mathrm{clip}_{[0,1]} \frac{v_\eta(x,h)}{\max\{\widehat{\rho}_\tau(x,h), \rho_{\min}\}}, \quad q(x) = q_H(x). \tag{18}$$

A shorter predicted time increases the denominator and lowers the candidate. This motivates a low quantile, but does not prove conservatism: even an exact quantile can give $q_h > p_h^{\pi}$ (Appendix B.4). Few successes leave little data for the time predictor, and small denominators amplify errors (Appendix B.3). A trajectory suffix is used only if its recording window covers the required horizon.

B.2 Proof of the Discounted Decomposition

Proof of Proposition 2. For $p_h^{\pi}(x) > 0$, the definition of conditional expectation gives

$$\mathbb{E}_x^{\pi}[\gamma^{\tau_{\mathcal{T}}} \mathbf{1}_{A_h}] = \Pr_x^{\pi}(A_h)\, \mathbb{E}_x^{\pi}[\gamma^{\tau_{\mathcal{T}}} \mid A_h] = p_h^{\pi}(x) \rho_h^{\pi}(x).$$

On A_h, $0 \le \tau_{\mathcal{T}} \le h$, hence $\gamma^h \le \gamma^{\tau_{\mathcal{T}}} \le 1$ and the same bounds hold for the conditional expectation. If $p_h^{\pi}(x) = 0$, the integrand is zero almost surely, and the defined value $\rho_h^{\pi}(x) = 0$ gives $v_h^{\pi}(x) = 0$ without dividing by zero.

For $h = 0$, success occurs exactly on $\mathcal{T}$. For $h \ge 1$, a start in $\mathcal{T}$ has value one and a start in $\mathcal{F}$ has value zero. At other states, conditioning on the first transition consumes one step of the horizon and

one discount factor. The Markov property gives (17). Its initial and terminal values determine every subsequent v_h^π by induction. □

B.3 Approximation Error and Rare Success

The sensitivity of the ratio can be bounded in terms of errors in its two components. At a state with $p = p_h^\pi(x) > 0$, abbreviate $v = v_h^\pi(x)$ and $\rho = \rho_h^\pi(x)$. Suppose that

$$\rho \geq c \geq \rho_{\min} > 0, \qquad |\widehat{v} - v| \leq e_v, \qquad |\widehat{\rho} - \rho| \leq e_\rho < c.$$

Let $\widetilde{\rho} = \max\{\widehat{\rho}, \rho_{\min}\}$. Since $\rho \geq \rho_{\min}$, flooring does not increase the denominator's error, so $|\widetilde{\rho} - \rho| \leq e_\rho$ and $\widetilde{\rho} \geq c - e_\rho$. Consequently,

$$\left| \mathrm{clip}_{[0,1]} \left(\frac{\widehat{v}}{\widetilde{\rho}} \right) - p \right| \leq \frac{e_v + p e_\rho}{c - e_\rho}. \tag{19}$$

Indeed, before clipping the numerator of the difference is $\widehat{v} - p\widetilde{\rho} = (\widehat{v} - v) + p(\rho - \widetilde{\rho})$; clipping cannot increase distance to $p \in [0, 1]$.

Equation (19) describes how errors are amplified when the denominator is small. It is conditional on the component error bounds; these are not supplied by the training losses. The exact denominator bound γ^h decreases with the horizon, and setting $\rho_{\min} > \rho$ introduces flooring bias. Rare success presents a separate statistical difficulty by reducing the number of auxiliary regression labels, even when the conditional discount factor is moderate.

B.4 Quantile-Based Auxiliary Candidates

For a successful suffix with relative hitting time D and covariates (x, h), quantile regression minimizes

$$\mathbb{E}[\ell_\tau(D - T_\xi(x, h)) \mid A_h], \qquad \ell_\tau(u) = u(\tau - \mathbf{1}\{u < 0\}).$$

At the population optimum it identifies a conditional τ-quantile. In the implemented variant, T_ξ has a sigmoid output scaled by H and $\tau = 0.1$. This auxiliary target differs from the conditional expectation ρ_h^π. Equation (19) applies only if the error in the resulting discount factor relative to the exact ρ_h^π is bounded; a quantile loss alone supplies no such bound.

To see why a low quantile is insufficient for conservatism, consider a state from which success has probability $p = 1/2$. Conditional on success, let $D = 1$ with probability 0.05 and $D = 100$ with probability 0.95, with $H = 100$ and $\gamma = 0.95$. The 0.1-quantile is 100, but

$$\rho = 0.05\gamma + 0.95\gamma^{100} > \gamma^{100} = \widehat{\rho}_{0.1}.$$

Even using the exact discounted value $v = p\rho$ and a floor $\rho_{\min} \leq \gamma^{100}$, the clipped ratio $\min\{v/\widehat{\rho}_{0.1}, 1\}$ equals one, exceeding p. A Markov process with a probabilistic initial branch and deterministic paths of lengths one and 100 realizes this example. Jensen's inequality instead gives $\gamma^{\mathbb{E}[D|A_H]} \leq \mathbb{E}[\gamma^D \mid A_H]$, so replacing the mean discount by the discount of the mean time can also produce optimism. Reducing the fitted time lowers the candidate, but does not prove that it is a valid lower bound.

A sufficient deterministic condition would be $\widehat{v} \leq v$ and $\widetilde{\rho} \geq \rho > 0$, which implies $\widehat{v}/\widetilde{\rho} \leq p$. The experiments do not establish these componentwise inequalities. Instead, independent validation tests the final calibrated candidate directly, covering auxiliary approximation, flooring, and calibration errors through the same distributional statement.

B.5 Relation to Infinite-Horizon Reach-Avoid

For a fixed policy, define $p_\infty^\pi(x) = \Pr_x^\pi\{\tau_{\mathcal{T}} < \infty, \tau_{\mathcal{T}} < \tau_{\mathcal{F}}\}$. The finite-horizon events increase to this event, so

$$p_H^\pi(x) \uparrow p_\infty^\pi(x), \qquad p_\infty^\pi(x) - p_H^\pi(x) = \Pr_x^\pi\{H < \tau_{\mathcal{T}} < \infty, \tau_{\mathcal{T}} < \tau_{\mathcal{F}}\}.$$

It follows immediately that

$$\{x : q(x) > p_\infty^\pi(x)\} \subseteq \{x : q(x) > p_H^\pi(x)\}.$$

Thus a valid distributional lower bound for the horizon-H event is also a distributional lower bound for eventual reach-avoid under the same policy and initial distribution. Its tightness for that eventual event is a different question.

Without assumptions on hitting-time tails there is no generally useful uniform truncation-error rate. One elementary bound is

$$0 \le p^{\pi}_{\infty}(x) - p^{\pi}_{H}(x) \le \Pr^{\pi}_{x}\{\min(\tau_{\mathcal{T}}, \tau_{\mathcal{F}}) > H\}.$$

The right-hand side is the probability that neither terminal set has been reached by H. A quantitative truncation bound therefore requires hitting-time information for the same fixed policy.

B.6 Statistical Model Checking and Function-Level Certificates

SMC certifies probability properties using simulation and statistical tests (Younes & Simmons, 2002; 2006). Nested probabilistic model checking accounts for errors in inner tests when conducting an outer test. Our risk link has this same error-propagation structure: the inner check tests a learned, state-dependent threshold $q(x)$, and the outer test controls the initial-state mass where that threshold exceeds the true probability. Repeating learning and verification is handled by a shared error budget. The CP false-support bound holds for every positive amount of overestimation; the closeness of q to the true probability affects power. As in Younes (2006), undecided outcomes let the test control error without an indifference region.

Smoothed model checking learns a satisfaction function over model parameters through Gaussian-process inference (Bortolussi et al., 2016). Here we certify a candidate function of the initial state through repeated-sampling error control, without assuming its smoothness or a probabilistic prior over functions. Both approaches learn probability functions; the guarantee and its assumptions distinguish them.

For a fixed starting distribution or a fixed selection rule, ordinary Monte Carlo can directly certify average success. A certified q instead supplies a state-dependent score at new starts without new trajectories, together with the population implications in Appendix A.5. Different thresholds can also reuse a common Monte Carlo dataset when their estimation error is controlled uniformly. For one selected region and threshold, direct Monte Carlo on $\mu(\cdot \mid A)$ is cheaper (Section 6.2); the certified function adds per-start bounds and simultaneous validity over every region defined by q.

The distinction concerns the certified quantity. For example, $p(x) \equiv 0.9$ and a probability function equal to one on 90% of starts and zero elsewhere have the same mean success. Yet the bound $q(x) \equiv 0.8$ has violation mass zero in the first case and 0.1 in the second. Knowing the mean alone cannot distinguish their coverage. In the second case, at $\epsilon = 0.01$, the largest constant satisfying $V_H \le \epsilon$ is zero: every positive constant overstates the probability on 10% of starts. A state-dependent function can instead equal 0.99 on the successful 90% and zero elsewhere, with violation mass zero. Repeated-rollout Monte Carlo at a specified start can also bound its success probability; evaluating the certified function supplies a prediction without such new rollouts, with the distributional coverage of (4).

B.7 RASM and LogRASM

RASM and LogRASM establish reach-avoid probability bounds by verifying inequalities for a certificate function over a state space (Žikelić et al., 2023; Badings et al., 2025). With their respective hypotheses satisfied, these bounds apply to every state in the specified initial set and concern eventual target hitting before failure.

A logarithmic representation writes a positive certificate as $U = \exp(V)$, with a real-valued function V. Its log-moment expression is $\log \mathbb{E}[U(X')] = \log \mathbb{E}[\exp(V(X'))]$. The complete hypotheses and verification conditions for the respective certificates are given in the cited papers.

Our test instead uses independent, restartable trajectory access to control the initial-state mass of violations for a finite-horizon probability bound. It allows an exceptional set and attaches confidence to the sampling procedure. These differences in horizon, coverage, and information requirements determine how the benchmark results in Section 6 can be compared.

Their verifiers use transition and disturbance descriptions to bound expectations and certificate variation between grid points. Our verifier instead uses binomial confidence limits on iid state–rollout blocks, with total error controlled across adaptive rounds. Failure feedback also differs: a violated certificate inequality identifies a region needing refinement, whereas a rejected rollout block only indicates insufficient local statistical support and does not prove that its state is unsafe.

Both approaches separate learning a candidate from verifying it. Their verification burdens differ: certificate methods must bound inequalities beyond sampled points, whereas our method obtains a distributional statement by sampling and explicitly allows exceptional states. A simulator that supports resets and rollouts may satisfy our access requirement without exposing the regularity information needed by those certificate verifiers. Conversely, a sound uniform certificate can establish a stronger initial-set claim than the distributional test. The finite-horizon lower bound also applies to eventual reach-avoid by Appendix B.5, but does not estimate the truncation gap.

B.8 Data-Driven Policy Synthesis and Repeated Verification

Data-driven policy synthesis and repeated statistical verification have existing guarantees. Table 5 identifies the output and conditions of selected nearby approaches. In particular, Ni & Kamgarpour (2025) protect a reach-avoid constraint during policy optimization, whereas our theorem certifies accepted outputs. Their safe initialization and optimization assumptions support a guarantee on learning iterates that we do not provide. Our acceptance rule needs no feasible initial policy, but does not guarantee convergence or an informative accepted output.

Table 5. Selected data-driven approaches and this work. Entries describe the cited formulations, rather than all variants of each method. These are comparisons of guarantees and access conditions, not empirical performance rankings.

Approach	Certified object	Construction and conditions
Ni & Kamgarpour (2025)	Reach-avoid chance constraint at a given start, maintained during policy optimization	Trajectory access; strictly feasible initial policy; policy smoothness, constraint qualification, and exploration assumptions
Nazeri et al. (2025)	Policy satisfaction probability through a sound interval MDP abstraction	Additive noise; separate samples of nominal dynamics and noise; known derivative or Lipschitz bounds; a finite partition
Salamati & Zamani (2022)	A lower bound on safety probability for an unknown stochastic system	Barrier construction through repeated scenario convex programs, fresh feasibility checks, and a final safety condition
This work	Accepted state-dependent bound, valid outside mass at most ϵ with confidence $1-\delta$ over the full procedure	Independent resets and rollouts under each frozen policy; arbitrary preceding learning and updates; fresh validation and a shared confidence budget

The interval abstraction of Nazeri et al. (2025) also uses CP bounds, there to enclose transition probabilities before policy synthesis. We apply local confidence limits directly to trajectory success under a proposed policy, then control the mass of initial states with overstated bounds. This avoids constructing a transition abstraction at the cost of a distributional initial-state claim and repeated rollout validation. Likewise, independent validation and repetition are established design choices (Salamati & Zamani, 2022); our analysis specifies their role for the learned policy–probability-bound pair. All these methods require data consistent with the system to which their guarantee is applied.

B.9 Direct Monte Carlo Alternatives

Certifying average policy success. For a fixed policy, draw independent $X_i \sim \mu$ and one independent horizon-H trajectory from each X_i. Its success indicator Y_i is Bernoulli with parameter $\bar{p} = \mathbb{E}_\mu[p_H^\pi(X)]$. A binomial confidence bound (Clopper & Pearson, 1934) on $\sum_i Y_i$ therefore certifies average success without learning a probability function. Fresh batches and confidence allocation can also cover repeated tests after policy updates. If a scalar average guarantee is the objective, this is a valid, simpler alternative. Our objective additionally represents variation in conditional success across initial states through $q(x)$. By Markov's inequality, an average lower bound b also gives the coarse tail bound $\mu\{p_H^\pi \geq \lambda\} \geq \max\{0, (b-\lambda)/(1-\lambda)\}$ for any fixed $\lambda < 1$; it does not imply (4) for a proposed function q.

Learning the state-dependent candidate with MC. MC regression can learn such a function directly from binary trajectory outcomes. Repeated rollouts at every training state are not required to define this regression problem. Under the same calibration and validation protocol, an accepted MC candidate receives exactly the guarantee of Theorem 1. MC regression is therefore a complete alternative learner under the same certificate; the fixed-policy comparisons isolate candidate construction. The observed post-calibration mean bounds support MC as an effective choice, and the calibration sweep does not establish a lower minimum certification cost for Ratio-TD. A learned function permits evaluation at new states without additional rollouts, but its distributional certificate cannot authorize any individually selected state without further justification (Appendix A.5).

B.10 Further Related Work and Deployment Scope

Constrained policy learning. Safe RL includes several ways to improve return while controlling costs. CPO uses a trust-region update (Achiam et al., 2017); PCPO separates reward improvement from projection onto a constraint set (Yang et al., 2020); FOCOPS uses first-order optimization in policy space (Zhang et al., 2020); and CUP develops conservative policy updates (Yang et al., 2022). RCPO uses a penalty signal and multiple update timescales (Tessler et al., 2018), while PID Lagrangian methods adjust the constraint multiplier using feedback (Stooke et al., 2020). Lyapunov-based safe RL constructs local constraints to control cumulative costs (Chow et al., 2018); feasible actor-critic instead explicitly addresses statewise safety (Ma et al., 2021). These methods address different notions of safety, from cumulative costs to statewise constraints, with guarantees tied to their optimization formulations. Our verifier instead controls false acceptance of a learned reach-avoid probability bound using finite samples. Our verifier can follow different policy learners, but does not by itself keep their training iterates safe.

Model-based reachability and robust specifications. Hamilton–Jacobi methods characterize reachable sets for continuous dynamic games (Mitchell et al., 2005), including reach-avoid games with state constraints (Margellos & Lygeros, 2011); Bansal et al. (2017) review their numerical methods and dimensionality challenges. Lyapunov-barrier characterizations also address robust reach-avoid-stay specifications for hybrid systems (Meng & Liu, 2023). Distributionally robust synthesis optimizes safety and reach-avoid satisfaction over an ambiguity set of disturbance distributions (Chen et al., 2025). These formulations explicitly address classes of disturbances or model uncertainty. Our theorem concerns the particular transition law generating the validation rollouts; it does not supply robustness to a different law.

Stochastic reach-avoid certificates. Barrier-like functions provide conditions for reach-avoid probability bounds in stochastic discrete-time systems (Xue et al., 2021). Related work develops finite-time bounds (Xue, 2025) and sufficient and necessary conditions under stated assumptions (Xue, 2026). Cao et al. (2025) compare several infinite-horizon barrier-like conditions and their computational properties. These results provide context for RASM and LogRASM, but their functional verification conditions differ from our direct statistical test of the proposed probability bound. The finite-horizon discounted recursion in Appendix B.1 constructs a candidate; its neural approximation need not satisfy a barrier inequality for the acceptance theorem to hold.

Learning certificates and dynamics from data. Data-driven formal verification covers unknown deterministic systems (Nejati et al., 2023), and multiple control barrier certificates support controller synthesis with finite input sets (Nejati & Zamani, 2023). For stochastic systems, Salamati et al. (2024) use sampled constraints to construct barrier certificates and synthesize controllers. Another route learns a dynamics model together with a neural Lyapunov function and stabilizing controller (Zhou et al., 2022), or learns dynamical structure for feedback controller design (Richards et al., 2023). These approaches establish different properties under their own structural and approximation assumptions. Our verifier directly samples trajectory outcomes and does not require a learned dynamics model or a certificate inequality between sampled states.

Neural reachability and calibration. Lin & Bansal (2024) verify neural reachable tubes using robust scenario optimization and conformal prediction, allowing a tradeoff between exceptional states and probabilistic safety guarantees. This is closely related to our use of distributional coverage. The verified object here is instead a lower bound on the conditional success probability of a stochastic

policy; each sampled state therefore has a local statistical test based on repeated trajectories. Repeated rollouts supply an observable local rejection risk that controls violations of the unknown probability bound. This supports either a fresh nested test or exact calibration with a training-conditional guarantee, with confidence allocation across rounds (Section 5.2).

Simulation and transfer to physical systems. MuJoCo (Todorov et al., 2012), Isaac Gym (Makoviychuk et al., 2021), CARLA (Dosovitskiy et al., 2017), and Brax (Freeman et al., 2021) illustrate simulation environments for robotics and driving. Simulation-based training has enabled transfer through domain randomization (Tobin et al., 2017), dexterous manipulation (Akkaya et al., 2019), massively parallel locomotion training (Rudin et al., 2022), and rapid motor adaptation (Kumar et al., 2021). These results motivate learning with simulators, but do not transfer our statistical certificate to hardware. Such a transfer would require a justified connection between the validation dynamics and deployment dynamics, together with the relevant initial-state distribution. Assumption 1 also requires independent restarts from queried full states, an access condition that may be difficult to realize on a physical platform.

C Statistical Implementation and Experimental Protocol

C.1 Computing the CP Test

For $q = 0$, local support is automatic since $L_\beta \geq 0$. For $q > 0$ and $s = 0$, support fails. For $q > 0$ and $s \geq 1$, inversion of (6) gives

$$L_\beta(s; M) \geq q \quad \Longleftrightarrow \quad \Pr_{\mathrm{Bin}(M,q)}(S \geq s) \leq \beta. \tag{20}$$

A survival-function routine defined as $\Pr(S > k)$ is therefore evaluated at $k = s - 1$. The outer test uses the lower tail $B_{N,(1-\beta)\epsilon}(K)$. The experiment implementation includes checks of both tails, $s = 0$, $s = M$, $q = 0$, $q = 1$, and exact enumeration of the local false-support probability. Stable tail evaluation avoids subtracting a cumulative probability close to one.

With all-success data, $L_\beta(M; M) = \beta^{1/M}$. This is the largest value that rollout-only local evidence can support, rather than an upper bound on every state of an accepted function. For $M = 1000$ and $\beta = 10^{-3}$ it is approximately 0.99312. One failure lowers it to approximately 0.99080. Thus a near-perfect policy and an accurate estimate close to one can still produce local rejections. At known target states the exact probability is one, but a rollout-only CP check cannot support $q = 1$ with finite data. Exact terminal predicates can be handled analytically if their use preserves the false-support bound.

Conditional on the pre-validation history, independent random streams and independent restored simulator states generate iid state–rollout blocks. Learning, fitting, shift calibration, reference evaluation, and validation have distinct roles. The candidate and sample sizes must be frozen before the validation batch. Prior rejections can guide the next candidate, but the next batch must use fresh samples. An incomplete batch does not yield a certificate.

C.2 Confidence Levels and Exact Cutoffs

The maximal acceptable rejection count is

$$k_* = \max\{k : B_{N,(1-\beta)\epsilon}(k) \leq \delta_t\},$$

with $k_* = -1$ when even zero rejections are insufficient. Table 6 lists the exact integer cutoffs for the experimental settings, computed from the binomial CDF.

The Frozen Lake known-truth and candidate-learning studies use $\delta_t = 0.005$, $\epsilon = 0.05$, and $\beta = 10^{-3}$. The resulting exact maximal cutoffs are 75 for $N = 2000$ and 443 for $N = 10000$. For example, $B_{2000,0.04995}(75) \simeq 0.004709 \leq 0.005$, whereas $B_{2000,0.04995}(76) \simeq 0.006482 > 0.005$; similarly, the CDF at 443 is approximately 0.004508 and at 444 approximately 0.005177.

The PointGoal, Hopper, Linear, and Planar fixed-policy comparisons use $\delta_t = \delta = 0.05$, giving cutoff 83 at $N = 10000$ and $\epsilon = 0.01$. Complete control synthesis, including the RAPC runs in Table 13, instead uses $\delta_t = \delta/R = 0.005$ for at most ten rounds, giving cutoff 74. Each fixed-policy

Table 6. Binomial thresholds with $\beta = 10^{-3}$. These are maximal cutoffs under the stated parameters, computed independently of empirical acceptance counts.

N	ϵ	δ_t	k_*	k_*/N
500	0.01	0.005	-1	—
2000	0.01	0.005	9	0.0045
2000	0.05	0.005	75	0.0375
10000	0.01	0.005	74	0.0074
10000	0.01	0.05	83	0.0083
10000	0.05	0.005	443	0.0443

confidence level applies to an individual test: 0.005 for Frozen Lake and 0.05 for the continuous-state tasks. Comparing many rows does not provide the same confidence simultaneously for every accepted entry, nor permit selecting a deployment claim without accounting for that selection. The complete synthesis procedure retains its allocated total error budget $\delta = 0.05$.

C.3 Learning Data and Split Calibration

The candidate models take the remaining horizon as an input. Value targets obey the finite-horizon boundary conditions. For MC, a binary label says whether the suffix reaches the target before failure within its available horizon. A suffix is eligible only when its recording window covers that horizon; retaining only shorter windows that happened to terminate would select outcomes. For Ratio-TD, successful eligible suffixes additionally supply hitting-time labels for the conditional quantile. Failure and timeout labels are not assigned artificial hitting times.

Candidate losses and calibration fit. The Probability-TD, MC, and discounted-value candidates use sigmoid outputs and cross-entropy with soft targets clipped to $[10^{-3}, 1 - 10^{-3}]$. Ratio-TD uses the same logit-space cross-entropy for its value branch, with the bootstrapped TD target in place of the label, and pinball loss for the time branch. Calibration freezes the time branch and operates on the value logit, avoiding the flat gradient of a ratio clipped at one. Control tasks use 3000 optimization steps and weight four on positive fitting residuals; Frozen Lake uses 4000 steps and weight one. Both use learning rate 10^{-2} and anchor weight 0.1. The anchor penalizes movement from the pre-calibration candidate; it does not impose a validity condition. Learned-candidate calibration uses the CP target in (9) with $M = 1000$; the constant comparator uses margin 0.1.

For Ratio-TD, write $\widetilde{\rho}(x) = \max\{\widehat{\rho}_\tau(x, H), \rho_{\min}\}$ for the fixed denominator and $z(x)$ for the value logit. The fit targets $\mathrm{logit}(\widetilde{\rho}(x)y(x))$, with clipping at numerical endpoints and y from (9). For direct candidates set $\widetilde{\rho} = 1$. The regression penalizes both positive and negative residuals, so it can raise low predictions. When $m_{\mathrm{fit}} \neq M$, the target y is not a confidence limit based on the m_{fit} observed trials.

Independent shift calibration. On a disjoint set, collect M rollouts per state and define the logit residual

$$e_i = z(x_i) - \mathrm{logit}(\widetilde{\rho}(x_i)y_i), \qquad y_i = L_\beta(S_i^{\mathrm{shift}}; M),$$

again clipping numerical endpoints. The implementation computes `np.quantile` at level $1 - \alpha$, using NumPy's default linear interpolation. For sorted residuals $e_{(1)} \leq \cdots \leq e_{(n)}$, let $u = (n-1)(1-\alpha)$, $j = \lfloor u \rfloor$, and $\theta = u - j$. It returns

$$c = (1 - \theta)e_{(j+1)} + \theta e_{(\min\{j+2,n\})}.$$

The scalar shift is subtracted before forming the calibrated candidate q_c of (11); increasing c lowers the function pointwise. The shift uses an interpolated empirical quantile, not a finite-sample conformal rank correction. It is chosen on data independent of the fitted network and precedes the final test. The interpolation and clipping prevent applying the exact calibration guarantee directly to this implementation, so fresh validation tests the final q_c. The exact-rank alternative retains the noisy CP limits without clipping and uses the risk link in (7) to certify the unknown success probability (Appendix A.4).

The principal experiments use $\alpha = \epsilon/2$: 0.005 for control and 0.025 for Frozen Lake. PointGoal, Hopper, Linear, and Planar each use 8192 fitting states with 64 trajectories each, plus 2048 independent shift states with 1000 trajectories each. Frozen Lake uses 2048 fitting states with 256 trajectories each and 1024 shift states with 1000 trajectories each. Candidate-learning budgets are additional to these calibration samples. In the parameter sweep, α is set to $\epsilon/4$ for $N \leq 2000$, 0.35ϵ for $2000 < N \leq 5000$, and $\epsilon/2$ otherwise. These are empirical design choices that reserve room below the outer cutoff, not requirements or sufficient conditions of the theorem.

In all fixed-policy comparisons, predictions below 0.02 are set to zero in the evaluated candidate. Lowering such predictions cannot increase the true violation mass. The complete PointGoal synthesis uses no such threshold. A constant comparator uses the ϵ-quantile of the sampled empirical success estimates minus a margin of 0.1, lowered to the smallest calibration target where necessary, followed by the experiment's calibration step; it is designed to give low predictions, but still requires validation. A distributional constant optimized to discard an ϵ fraction of states could be less conservative, so the gap to this comparator does not establish superiority over all constant certificates.

C.4 Known-Truth Experiments

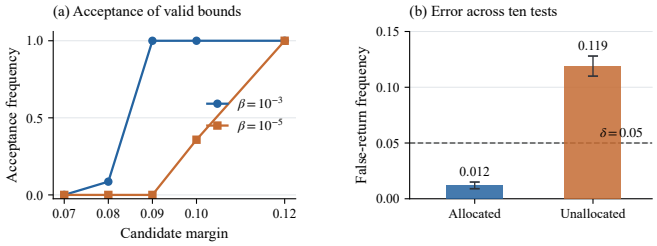

Figure 3. Known-truth tests, 5000 repetitions per condition. Left: larger margins improve acceptance of valid bounds. Right: allocating the budget across ten rounds keeps false returns below $\delta = 0.05$ (dashed). Error bars: 95% CP intervals.

Figure 3 summarizes the acceptance and repeated-testing experiments. The complete settings and counts follow.

For finite states with known probabilities p_j, weights w_j, and candidate values q_j, the true violation mass and local rejection probability can be evaluated without learning:

$$V_H = \sum_j w_j \mathbf{1}\{q_j > p_j\}, \qquad r = \sum_j w_j \sum_{s=0}^{M} \binom{M}{s} p_j^s (1-p_j)^{M-s} \mathbf{1}\{L_\beta(s;M) < q_j\}.$$

The acceptance probability of a fresh batch with cutoff k_* is $B_{N,r}(k_*)$. These formulas provide the analytic references for the synthetic and finite-state tests; they require neither an approximation to p_j nor a posterior distribution over it.

The synthetic power study uses 40 states spanning probabilities from zero to one. It sets $\epsilon = 0.01$, $\delta = 0.05$, $R = 10$, $N = 2000$, $M = 1000$, and cutoff 9, with 5000 repetitions per condition. The best-case state-sample requirement is $N \geq 528$; at $\beta = 10^{-3}$, locally supporting 0.9 requires $M \geq 66$ and supporting 0.99 requires $M \geq 688$. These necessities do not ensure high power.

Table 7. Acceptance frequency for valid synthetic candidates $q = [p - m]_+$. All entries use 5000 trials.

Local error	$m = 0.07$	0.08	0.09	0.10	0.12
$\beta = 10^{-3}$	0	0.086	0.9998	1	1
$\beta = 10^{-5}$	0	0	0	0.358	1

To probe violation mass, the synthetic study raises selected predictions above the true probability by 0.01. At $\beta = 10^{-3}$, acceptance frequencies for masses 0.5ϵ, 0.9ϵ, 1.1ϵ, and 2ϵ are 0.414, 0.010, 0.0004, and zero. The first two masses are allowed by the distributional specification even though some states are overestimated. With zero events among 5000 repetitions, the two-sided 95% CP upper endpoint is about 0.000738; zero observations are not a proof of zero error. Lemma 1 is uniform over the amount of overestimation, but actual detection probability can depend on that amount.

The known-truth Frozen Lake check uses an 8×8 slippery grid, $H = 100$, and a policy with exploration probability 0.1 around the horizon-optimal DP policy. Initial sampling is uniform over the 53 nonterminal cells, excluding the ten holes and the goal. Overstating one, two, or three such cells gives violation masses $1/53 \simeq 0.01887$, $2/53 \simeq 0.03774$, and $3/53 \simeq 0.05660$. At $N = 2000$, $\delta_t = 0.005$, and violation tolerance 0.05, the exact cutoff is 75. A valid margin-0.08 candidate accepts in all repetitions; the three overstatement cases have observed acceptance frequencies 1, 0.48, and zero, respectively. This test uses a different policy from the candidate-learning study in Section 6.3.

Table 8. Repeated synthetic procedures, 5000 independent repetitions and at most ten tests each. Intervals are reported 95% CP intervals for returning an invalid candidate. The adaptive rule lowers predictions at rejected states.

Candidate rule	Per-round budget	Invalid returns	Frequency	95% interval
Fixed, $V_H = 1.1\epsilon$	δ/R	60	0.012	$[0.009, 0.015]$
Fixed, $V_H = 1.1\epsilon$	δ	593	0.119	$[0.110, 0.128]$
Adaptive	δ/R	9	0.0018	$[0.0008, 0.0034]$
Adaptive	δ	57	0.011	$[0.009, 0.015]$

For a valid margin-0.08 candidate, allocation accepts $2980/5000$ complete procedures, compared with $4853/5000$ without allocation. Mean first-acceptance rounds among successful procedures are 4.7 and 2.9, respectively. Thus allocation has a visible power cost in this difficult setting, even though the PointGoal sensitivity sweep detects little change when varying δ_t over $[0.001, 0.01]$.

D Experimental Settings and Additional Results

Environment and learning settings appear in Sections D.1–D.3. Section D.6 collects full comparisons and failed runs; Section D.7 examines calibration budgets. Sections D.8 and D.9 report parameter sensitivity and computational cost, followed by the scope of the evidence in Section D.10.

The experiments are implemented in JAX (Bradbury et al., 2021) and run as single-GPU jobs on an NVIDIA H20 fleet. The software environment pins JAX 0.6.2, Brax 0.14.1, MuJoCo and MJX 3.3.3, Flax 0.10.6, Optax 0.2.5, Distrax 0.1.5, and Gymnax 1.0.0. The full synthesis study contains ten seeds per task, including unsuccessful policy-learning runs; the eight-seed Hopper subset in Table 17 is identified separately. Comparisons among modes share each seed's initial policy checkpoint, while subsequent learning, calibration, and validation have their own random draws. Means and standard deviations describe seed variation; they are not confidence intervals for a statewise success probability. The code is available at the anonymous repository linked in the abstract.

D.1 Control Environments

The policy and bound use the observation $o(x)$, with $q(x) = q_\theta(o(x), H)$ and $\pi(du \mid x) = \pi_\phi(du \mid o(x))$. For the robotic tasks, the interface supplies observations and rollout access. PointGoal's policy and bound see sensor and lidar readings rather than the internal position coordinates; Hopper's observation contains its generalized coordinates and velocities. Table 9 lists the state and input dimensions and the coordinates in which each validation distribution μ varies. Simulator state restoration repeats trajectories from the same initial condition, as required by Assumption 1.

Table 9. State and input dimensions. The last column lists the coordinates that vary under the validation distribution μ; the remaining coordinates are fixed or carry only the Brax reset noise of ± 0.005.

Task	State	Policy/bound input	μ varies in
Linear	2	2	both coordinates (two squares)
Planar	3	3	all three coordinates
Hopper	12	13 (12 observed + unsafe indicator)	head $x \in [0.2, 0.6]$ (wider in Appendix D.5)
PointGoal	6	44 (37 non-constant)	planar position and yaw; zero velocities

The event is checked at time zero and after transitions. With signed target and failure functions g, h, entry is detected by $g < 0$ and $h \geq 0$, respectively. The implementation gives target detection



(a) Hopper **(b)** PointGoal

Figure 4. Hopper and PointGoal reach-avoid environments. Table 11 reports certified synthesis results.

priority if numerical predicates overlap; equivalently, the mathematical failure set excludes the target. Arrival at either terminal set ends the trajectory, and failure to reach the target within the specified horizon receives label zero. A reset restores the full simulator state even when the neural input is a lower-dimensional representation.

Linear system. The two-dimensional system, adapted from Žikelić et al. (2023), has dynamics

$$\begin{aligned} x(l+1) &= 0.8x(l) + 0.05y(l) + 0.01\omega_1(l) + 0.1a_1, \\ y(l+1) &= 0.8y(l) + 0.005\omega_2(l) + 0.1a_2, \end{aligned} \tag{21}$$

with $\mathcal{X} = [-1,1]^2$, $a_1, a_2 \in [-1,1]$, and independent disturbance components with triangular density $1 - |\omega|$ on $[-1,1]$. The sets are

$$\begin{aligned} \mathcal{X}_0 &= [-0.8,-0.7]^2 \cup [0.7,0.8]^2, \qquad \mathcal{T} = [-0.1,0.1]^2, \\ \mathcal{F} &= ([-0.9,-0.7] \cup [0.7,0.9]) \times [-0.2,0.2]. \end{aligned}$$

The initial distribution chooses either square with probability one half and is uniform within the chosen square. There is no additional action perturbation.

Planar robot. The three-dimensional state and two-dimensional action satisfy $\mathcal{X} = [-1,1]^3$, $\mathcal{U} = [-1,1]^2$, and

$$f(x,u,\omega) = 0.7x + \Delta \begin{pmatrix} (x_2 + 2\Delta u_1)\cos(\pi u_2) \\ (x_2 + 2\Delta u_1)\sin(\pi u_2) \\ u_1 \end{pmatrix} + 0.01(\omega_1, \omega_2, 0)^\top, \qquad \Delta = 0.2. \tag{22}$$

The disturbance law is the same independent triangular law. Initial states are uniform on $\mathcal{X}_0 = [0.4,0.6] \times [-0.8,-0.6] \times [-0.1,0.1]$; the target is $\mathcal{T} = [-0.1,0.1]^2 \times [-1,1]$ and

$$\begin{aligned} \mathcal{F} = {} & ([-1,-0.8] \times [-1,0] \times [-1,1]) \cup ([-0.1,1] \times [0.8,1] \times [-1,1]) \\ & \cup ([0.8,1] \times [0,0.8] \times [-1,1]) \cup ([-0.3,-0.2] \times [-0.3,-0.2] \times [-1,1]). \end{aligned}$$

There is no additional action perturbation.

Hopper. The Brax-based Hopper task (Freeman et al., 2021) has a 12-dimensional state (six generalized coordinates and their velocities). The policy and bound receive the 12-dimensional Brax observation, which here includes the root x position, augmented with an indicator of the unsafe region, giving 13 inputs. A head position within distance 0.1 of $(2.0, 1.4)$ is the target. The unsafe ceiling is detected when the head has $y \geq 1.3$ and $x \in [0.95, 1.05]$. Falling is not a separate termination condition. An action is repeated twice. Validation and diagnostic starts have head x uniform on $[0.2, 0.6]$, with the remaining coordinates following the Brax reset. The inherited `reset_noise_scale` is 5×10^{-3}: the reset perturbs positions and velocities by uniform noise on $[-0.005, 0.005]$. Policy-learning resets use head $x \in [0.2, 2.0]$ to permit exploration; this broader learning distribution is not substituted for the validation distribution. A separate study (Appendix D.5) uses this range without terminal starts as μ.

PointGoal. The JAX PointGoal task is adapted from Safety Gym (Ray et al., 2019), with a 44-dimensional observation: accelerometer, velocimeter, gyroscope, and magnetometer readings (three each) and 16 lidar bins each for the goal and the hazards. In this planar simulation the accelerometer, the vertical velocimeter and magnetometer components, and the two in-plane gyroscope components

are identically zero, leaving 37 informative coordinates. The underlying physical state is six-dimensional: planar position, yaw, and their velocities. The goal is the disk of radius 0.2 at the origin. Eight fixed hazards have radius 0.1, and the agent radius is 0.1. Their centers are

$$(-0.8, -1.1),\ (1.2, -0.9),\ (-1.1, 0.7),\ (0.9, 1.3),$$
$$(0.1, -1.2),\ (-1.3, -0.1),\ (0.8, 0.1),\ (-0.2, 1.0).$$

Leaving the arena, $|x| > 1.8$ or $|y| > 1.8$, counts as failure. Initial positions are uniform on $[-0.2, 0.2] \times [-0.5, -0.2]$, orientation is uniform, and initial velocities are zero. Starts are checked against the terminal predicates; an initially unsafe state is not allowed to escape before its label is assigned. One environment step draws the action noise once, then holds the perturbed action for twenty MuJoCo substeps of 0.005 s, so a step is 0.1 s and $H = 400$ spans 40 s of simulated time. The goal, hazard, and boundary predicates are evaluated once per step on the state after the twenty substeps; a hazard crossed and left within one step is not detected, and the certified event is defined with respect to this step-level check. The noise key is split from the environment key of the rollout, independently of the policy's sampling key; hazard and goal positions are fixed constants.

Robotic action noise. Both robotic tasks apply $\tanh$, independent multiplicative noise, and clipping to an unbounded control vector b:

$$u = \mathrm{clip}_{[-1,1]}\big(\tanh(b) \odot (1 + \zeta)\big), \qquad \zeta_j \sim \mathrm{Unif}[-0.05, 0.05],$$

with a fresh noise draw once per environment step. In Hopper, $b = a \in \mathbb{R}^3$ is the policy output. In PointGoal, the policy outputs $a = (a_f, a_r) \in \mathbb{R}^2$, and the control vector is formed in the world frame from the current yaw ψ before the nonlinearity, $b = (a_f \cos\psi,\ a_f \sin\psi,\ a_r)$, so the noise has three components. These dynamics, action-repeat settings, and reset laws define the tested simulator.

D.2 Frozen Lake and Exact Reference Probabilities

The discrete task is the slippery 8×8 Frozen Lake grid in Figure 5, with target at the bottom-right corner, ten holes as failure states, and horizon $H = 100$ (Brockman et al., 2016). Initial sampling is uniform over the 53 nonterminal cells, excluding both the holes and the goal. Given a selected direction, the transition attempts that direction or either perpendicular direction with probability $1/3$ each. Dynamic programming under the fixed experimental policy gives p_h^π by the probability recursion and v_h^π by (17). Exact V_H is the uniform weighted sum over this initial-state support, not an estimate from validation counts.

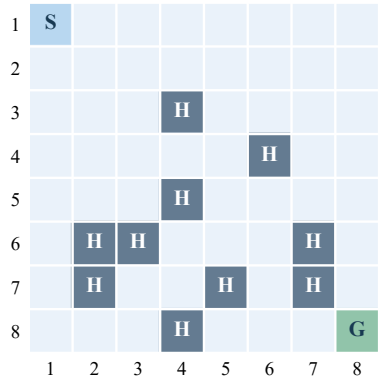

Figure 5. The Frozen Lake map. Light cells denote ice, `H` a hole, and `G` the goal. `S` marks the conventional start cell; experimental initial states are uniform over the 53 nonterminal cells (`S` and `F`).

For the candidate-learning comparison, a tabular softmax policy is trained for 2000 Adam steps at learning rate 0.05, maximizing the μ-weighted exact finite-horizon success probability plus an entropy term with weight 0.02. The final iterate is used for candidate learning and evaluation. Its exact mean success is 0.5597 (reported as 0.560), with minimum 0.062 over nonterminal cells. This policy differs from the horizon-optimal DP policy with 0.1-soft exploration used in the known-probability verifier study; results for the two policies are not pooled.

The learning discount is 0.95 and the violation tolerance is 0.05. Learning budgets are fractions $0.1, 0.25, 0.5, 1$ of 4096 candidate trajectories with recording horizon $H = 100$. The same policy

and budget convention apply to all five candidate methods within each seed. Calibration data are additional and fixed across these fractions. Figure 6 shows that calibration accounts for much of the final bound height; additional candidate data mostly improve Ratio-TD within the tested range. A mean local lower-limit reference of about 0.52 indicates the scale of finite-sample conservatism, but is not an optimal achievable distributional certificate.

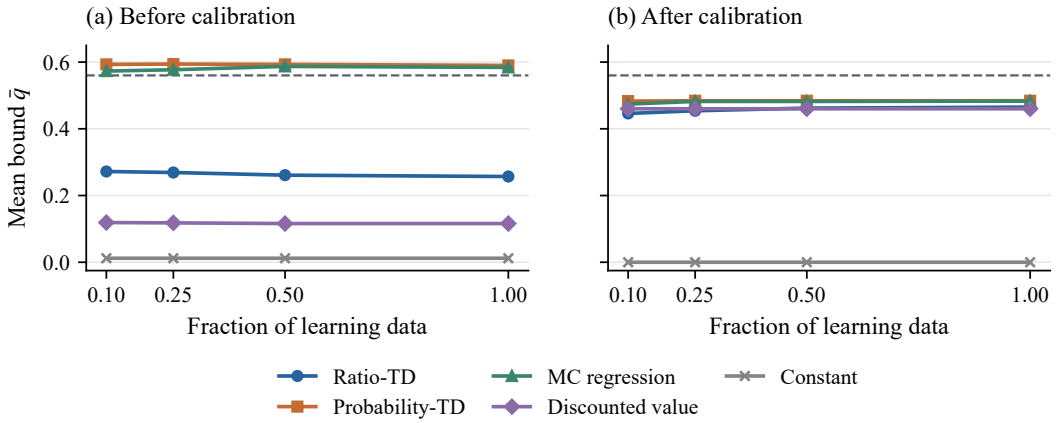

Figure 6. Frozen Lake candidate means across ten seeds, before and after calibration. The dashed line is the exact mean success 0.560. Curves are reported aggregate values, not statewise maps. Calibrated learned candidates have exact $V_H = 0$ on the initial distribution.

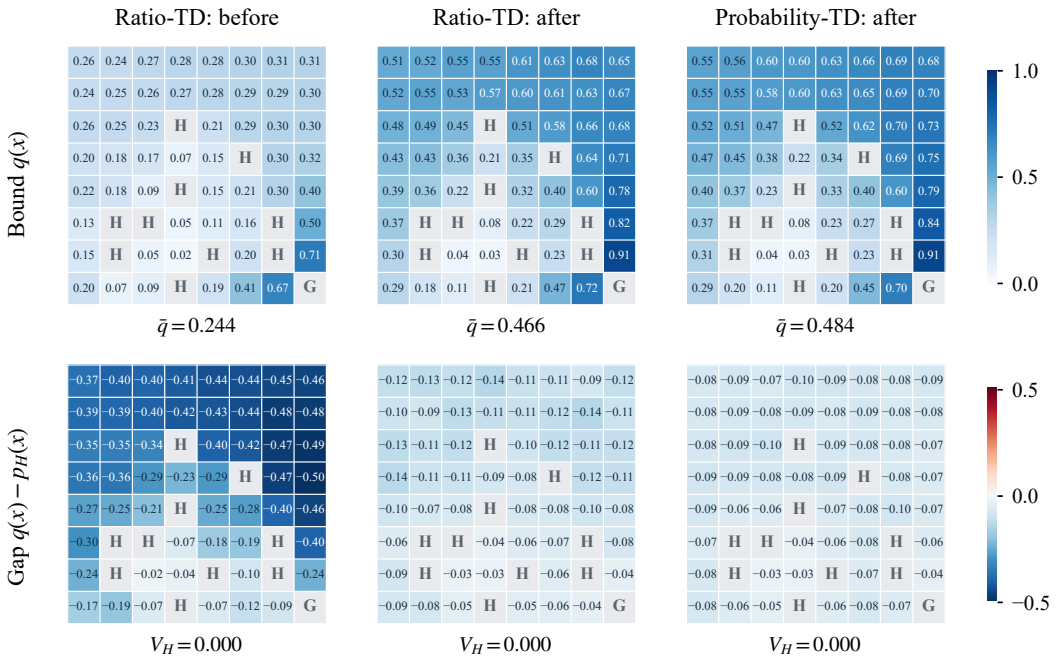

Figure 7. Frozen Lake statewise comparison, seed 0 at full candidate-learning budget. Top: candidate values; bottom: differences from exact p_H, with a common color scale within each row. Gray terminal cells are excluded from μ. Calibration raises Ratio-TD's mean from 0.244 to 0.466, while calibrated Probability-TD reaches 0.484. All three displayed candidates have exact $V_H = 0$; these single-seed maps complement the ten-seed summaries.

D.3 Training and Data Budgets

All control tasks use a JAX implementation of RESPO (Ganai et al., 2023). The reward is 20 on first target entry plus progress shaping $(g(x_t) - g(x_{t+1}))/100$; the cost is 20 on first failure entry. No energy penalty is included. Target and failure labels follow the same event convention as validation, although the reward objective need not equal the verified probability.

The value and auxiliary time networks each have two hidden layers of 128 units with ELU activations. Inputs concatenate the observation and normalized remaining horizon h/H. The value head is sigmoid; the auxiliary sigmoid head is scaled by H to predict a hitting-time quantile, from which $\widehat{\rho}_\tau = \gamma^{T_{\xi,\tau}}$ is formed. The ratio uses $\rho_{\min} = 10^{-3}$. Formal candidate fitting uses 4000 optimization steps, batch size 4096, learning rate 10^{-3}, and Polyak target-update coefficient 0.02. The auxiliary loss uses $\tau = 0.1$. The separate calibration protocol uses 3000 steps for control tasks and 4000 for Frozen Lake (Appendix C.3).

Table 10. Reported training and validation configuration. Paired calibration settings are control / Frozen Lake. PointGoal's actor learning rate differs from the common default. The auxiliary discount for RESPO is separate from the horizon of the verified event.

Parameter	Setting
Control horizon; learning discount	$H = 400$; $\gamma = 0.99$
Frozen Lake horizon; discount	$H = 100$; $\gamma = 0.95$
Control validation	$N = 10000$, $M = 1000$, $\beta = 10^{-3}$
Synthesis tolerance; total confidence budget	$\epsilon = 0.01$; $\delta = 0.05$
Maximum synthesis rounds; round allocation	$R = 10$; $\delta_t = 0.005$
RESPO reachability discount	0.99
RESPO multiplier	Initial value 1, softplus parameterization
Multiplier / reachability-network learning rates	5×10^{-5} / 10^{-4}
Actor learning rate	10^{-3}, linearly decayed; PointGoal 5×10^{-4}
Critic learning rate	10^{-3}, linearly decayed
Entropy coefficient	10^{-2}, linearly decayed
PPO clip; GAE; target KL	0.2; 0.95; 0.02
Policy log standard deviation	Clipped to $[-5, 0.5]$
Ratio-TD auxiliary mode	Hitting-time quantile, $\tau = 0.1$
Candidate hidden layers; activation	2×128; ELU
Candidate input; denominator floor	$(\mathrm{obs}, h/H)$; $\rho_{\min} = 10^{-3}$
Candidate steps; batch size	4000; 4096
Candidate learning rate; Polyak coefficient	10^{-3}; 0.02
Calibration steps; target	3000 / 4000; CP lower confidence limit
Calibration learning rate; anchor weight	10^{-2}; 0.1
Configured margin; local-target sample size	0.1; 1000
Overprediction residual weight	4 / 1
Control / Frozen Lake shift-tail fraction α	0.005 / 0.025

Linear, Planar, and Hopper use 50 policy-pretraining iterations. PointGoal uses 800 iterations; from iteration 25 onward, each multiple of 25 is evaluated on 256 independent starts with 200 trajectories each. The checkpoint with the highest diagnostic success is selected before certificate learning and validation. Its full optimization state, including multiplier and optimizer/schedule states, is restored for continued training. Checkpoint diagnostics are charged to policy learning.

After a rejected round, modes A and C continue policy learning for ten iterations. In complete-feedback mode C, rejected states supply 50% of policy-learning starts, with the rest from the task's ordinary learning reset distribution. New-policy trajectories are used to refit the bound. Mode B fixes the policy and recalibrates. Rejected states also supply additional calibration targets: 64 rollouts per state, using at most 512 rejected states. These samples are learning data, not the next validation batch.

For each control task, the full candidate-learning budget is 2048 trajectories with recording horizon $H = 400$; the fixed-policy comparisons use fractions 0.25 and 1 of this budget. Each is followed by 8192 fitting states with 64 trajectories each, 2048 shift states with 1000 trajectories each, and a fresh $N \times M$ verification batch. The separate PointGoal reference evaluation uses 128 starts with 2000 trajectories each and has mean success 0.996 across seeds. Agreement with this finite reference is descriptive; a reference overestimation fraction $\widehat{V}$ need not equal true V_H. All shared calibration costs must be included when interpreting apparent learning efficiency.

D.4 Policy Refinement and Baseline Comparisons

From a common checkpoint per seed, Train (A) continues policy training, Calibrate (B) fixes the policy and recalibrates, and Feedback (C) trains with rejected-state starts. All modes calibrate before verification (Table 11).

Table 11. Robotic policy synthesis, all ten seeds per task and mode. Train (A), Calibrate (B), and Feedback (C) differ in their response to rejection; all calibrate before validation. Rounds, empirical success, and mean bound $\bar{q}$ summarize accepted seeds ($\pm$ seed s.d.), with success and $\bar{q}$ computed on the accepted validation batch; transitions average all attempts, including exhausted ones.

Task	Mode	Accept	Rounds	Success	$\bar{q}$	Transitions
Hopper	Train	10/10	1.30	0.973 ± 0.056	0.941 ± 0.097	2.12×10^{9}
	Calibrate	9/10	1.11	0.987 ± 0.038	0.967 ± 0.057	6.18×10^{9}
	Feedback	10/10	1.30	0.976 ± 0.049	0.945 ± 0.088	2.07×10^{9}
PointGoal	Train	10/10	1.20	0.995 ± 0.002	0.939 ± 0.010	4.61×10^{8}
	Calibrate	10/10	1.10	0.995 ± 0.002	0.940 ± 0.011	4.50×10^{8}
	Feedback	10/10	1.10	0.995 ± 0.002	0.940 ± 0.011	4.50×10^{8}

Recovery and refinement. Both training modes recover Hopper seed 2, which has no successes in its initial diagnostic sample, and pass in round three. Fixed-policy calibration returns UNKNOWN after ten rounds; Appendix A.4 explains the effect of target clipping. All thirty PointGoal attempts with strong initial policies pass in the first or second round. To study refinement over several rounds, we also initialize weak PointGoal policies and require both validation acceptance and a batch mean bound of at least 0.9. This additional height requirement preserves Algorithm 1's error control by restricting its return events.

From success 0.75–0.93 and calibrated bounds near 0.6, Feedback reaches the target within ten rounds in seven seeds and Train in five (Table 12); fixed-policy calibration stays near 0.63. Feedback is more variable: three runs collapse to success 0.51–0.64, their calibrated means fall to 0.04–0.24 and stay below the return threshold, two recover in the next round, and one collapses in the final round (Appendix A.4 details the rejections). Train never collapses, but five seeds stall at bounds 0.83–0.89.

Table 12. Refinement from weak PointGoal policies (ten seeds, RESPO stopped after 60 iterations). A passed round is returned only if the batch mean of q reaches 0.9; otherwise the policy is refined for 50 iterations (A, C) or the bound is recalibrated (B) and re-verified with a fresh batch, for at most ten rounds. Rounds, final success, and final $\bar{q}$ average the seeds that reached the target; transitions average all seeds. Calibrate (B) ran on three seeds.

Mode	Reached 0.9	Rounds	Initial succ.	Final succ.	Final $\bar{q}$	Transitions
Train (A)	5/10	6.8 ± 2.8	0.836 ± 0.047	0.987 ± 0.004	0.912 ± 0.007	1.38×10^{9}
Calibrate (B)	0/3	–	0.836 ± 0.026	–	–	2.31×10^{9}
Feedback (C)	7/10	7.6 ± 2.1	0.836 ± 0.047	0.988 ± 0.004	0.918 ± 0.017	1.39×10^{9}

Comparison with RASM and LogRASM. Table 13 compares uniform initial-set certificates from the authors' RASM and LogRASM implementations (Žikelić et al., 2023; Badings et al., 2025) with RAPC's distributional PAC certificates on Linear and Planar. The baselines were not run on Hopper and PointGoal: they need an analytical or Lipschitz-bounded model of the dynamics and a discretization of the state space, whereas these tasks provide only a simulator (Appendix D.11). RAPC reports sample extrema of accepted bound functions, with violation mass at most ϵ (Appendix B.7). Its full learning, calibration, and verification pipeline takes about two minutes per seed (at most 3.2) on one H20. Under a 30-minute budget on the same hardware, LogRASM certifies 0.9999 on Linear, exceeding the local rollout evidence ceiling $\beta^{1/M} = 0.9931$ at $M = 1000$; on Planar, at least six of the ten runs of each baseline time out even for a target of 0.1. Appendix D.11 specifies the protocol.

Table 13. Comparison on Linear and Planar (%, ten seeds). Baselines: 30 minutes per run on an H20 machine; RAPC: about two minutes per seed on one H20. RAPC low/high are the minima/maxima of calibrated q over 10,000 validation states at first CP acceptance, mean $\pm$ s.d. across seeds; $\beta^{1/M} = 0.9931$ is the largest value a single local check can support. "Timed out": at least six of ten runs did not finish verifying a 0.1 target within 30 minutes (Appendix D.11).

Method	Reported quantity	Linear	Planar
RASM	Uniform verified bound	81.25	Timed out
LogRASM	Uniform verified bound	99.99	Timed out
RAPC	Candidate low	98.96 ± 0.23	98.18 ± 1.19
	Candidate high	99.31 ± 0.06	99.09 ± 0.21

D.5 Hopper with Wide Initial States

Setup. Under the original Hopper distribution, head $x \in [0.2, 0.6]$, every start is far from the ceiling block. Eight of the ten accepted policies succeed from almost every start, and their learned bounds and best constants coincide near the local ceiling (Table 1). To test state dependence where success varies across starts, we rerun complete Feedback synthesis with μ replaced by the policy-learning start range $[0.2, 2.0]$ without terminal starts: head x uniform on $[0.2, 1.9] \setminus [0.95, 1.05]$, with the other coordinates as in the Brax reset. Starts with head $x \in [0.95, 1.05]$ lie inside the ceiling block and hence in $\mathcal{F}$; their mass of 5.9% exceeds ϵ and would force every valid constant to zero. Starts with head $x > 1.9$ lie in or next to the goal disc. None of 10^5 sampled starts lies in $\mathcal{T}$ or $\mathcal{F}$. Each seed starts from the initial pair of its main run, with the same learning settings, ϵ, δ, N, M, β, and ten-round budget.

Results with Ratio-TD. All ten runs return certified pairs, after 2–4 rounds (Table 14). Starts next to the ceiling, with head $x \in [0.9, 0.95) \cup [1.05, 1.1)$, are harder; elsewhere the returned policies succeed almost surely, except that the Ratio-TD policy of seed 2 also fails from the farthest starts (Figure 8). The share of validation starts with empirical success below 0.9 ranges from 0.3% to 11.3%. When it clearly exceeds ϵ, the best constant must cover the hard starts: in seeds 2, 3, 5, and 6 it falls to 0.01–0.24, while the learned mean bound is 0.58–0.89. Averaged over seeds, the learned bound is 0.82 ± 0.11 and the best constant 0.51 ± 0.38. In seeds 1 and 9 the learned bound is 0.04 below the constant; in seed 4, whose policy succeeds almost everywhere, the two nearly coincide.

Role of the learner. The Ratio-TD bound decreases with the distance from the goal even where success is constant (Figure 8a). Its candidate is $q = \min\{\sigma(z - c)/\widetilde{\rho}, 1\}$, with $\widetilde{\rho}(x)$ a discount estimate that shrinks with the time needed to reach the target. Where $\sigma(z)$ is small, subtracting the calibration shift c lowers q roughly by the factor e^{-c}; near the goal, where $\widetilde{\rho}$ and $\sigma(z)$ approach one, it lowers q much less. Hitting times vary little under the original μ but span the whole corridor here. After these results, we ran the same procedure with the MC learner, whose denominator is one. Its bound stays between 0.95 and 0.99 away from the ceiling, except at the farthest starts of seeds 2 and 7, where success itself is lower, and drops near the ceiling. Nine of the ten MC runs return certified pairs, after 2–4 rounds; seed 1 returns Unknown (below). Over these nine seeds the learned mean bound is 0.97 ± 0.01 (0.956–0.977) and the best constant 0.61 ± 0.26; the learned bound exceeds the constant in every seed, by more than 0.1 in seven.

Starting regions. Table 15 evaluates $A_{0.8} = \{q \geq 0.8\}$ on 10^5 fresh starts per seed. With MC, $A_{0.8}$ keeps 94–100% of the starts in every certified seed, with coverage lower bounds of at least 98.9%. In the seven seeds where more than 1% of validation starts have empirical success below 0.9 (1.1–5.3%), at most 0.43% of the starts in $A_{0.8}$ do; in seeds 4 and 7 the region contains almost every start. The lower Ratio-TD bound gives smaller regions, 19–100% of starts, with coverage lower bounds of at least 94.6%; outside seeds 4 and 7, none of its validation starts in $A_{0.8}$ has empirical success below 0.9.

Feedback instability. Feedback training can degrade a nearly accepted policy. Ratio-TD seeds 1 and 9 had 77 and 79 rejections in round 2 (cutoff 74), fell from success 0.996 and 0.997 to 0.935 and 0.944 in round 3, and passed in round 4. MC seed 1 had 76 rejections in round 2, fell from success 0.996 to 0.927 in round 3, and stayed at 0.93–0.94 through round 10, returning Unknown.

Table 14. Wide-start Hopper, per seed: complete Feedback synthesis from the main runs' initial pairs with μ widened to head $x \in [0.2, 1.9] \setminus [0.95, 1.05]$. Rounds until acceptance, learned mean bound $\bar{q}$ and best accepted constant c_{oracle} on the first accepted validation batch (as in Table 1), and the percentage of its starts with empirical success below 0.9. Ratio-TD is the learner of all other synthesis runs; the MC arm was run after the Ratio-TD results. Original μ: the main Hopper runs of the same seeds. Dashes: MC seed 1 returned UNKNOWN after ten rounds. Means are over certified seeds; standard deviations are in the text.

	Ratio-TD, wide μ				MC, wide μ				Ratio-TD, original μ	
Seed	rounds	$\bar{q}$	c_{oracle}	% < 0.9	rounds	$\bar{q}$	c_{oracle}	% < 0.9	$\bar{q}$	c_{oracle}
0	2	0.911	0.774	1.5	2	0.973	0.699	1.5	0.989	0.991
1	4	0.667	0.707	1.2	–	–	–	–	0.990	0.991
2	3	0.583	0.012	11.3	4	0.957	0.466	5.2	0.747	0.165
3	3	0.819	0.038	2.0	3	0.975	0.365	1.8	0.987	0.991
4	2	0.936	0.930	0.3	2	0.968	0.931	0.4	0.816	0.709
5	3	0.854	0.023	3.5	2	0.972	0.614	1.6	0.984	0.989
6	3	0.887	0.238	3.6	3	0.956	0.114	3.4	0.981	0.980
7	2	0.917	0.767	1.2	3	0.956	0.883	0.5	0.987	0.987
8	2	0.810	0.760	1.3	2	0.977	0.786	1.1	0.990	0.991
9	4	0.821	0.856	0.8	2	0.964	0.634	2.7	0.983	0.987
mean	2.8	0.821	0.511	2.7	2.6	0.966	0.610	2.0	0.945	0.878

Table 15. Wide-start Hopper regions $A_{0.8} = \{q \geq 0.8\}$. $\widehat{\mu}(A_{0.8})$ is the region mass on 10^5 fresh initial states (no rollouts); the coverage lower bound $1 - \epsilon/\mu_{\text{lo}}$ uses the one-sided DKW–Massart limit $\mu_{\text{lo}} = \widehat{\mu} - 0.0043$ ($\eta = 0.025$, uniform over thresholds) and holds with confidence at least 0.925 together with the certificate. The last column of each arm is the percentage of validation-batch starts with empirical success below 0.9, over all starts $\rightarrow$ within $A_{0.8}$.

	Ratio-TD			MC		
Seed	$\widehat{\mu}(A_{0.8})$	coverage $\geq$ (%)	% < 0.9	$\widehat{\mu}(A_{0.8})$	coverage $\geq$ (%)	% < 0.9
0	0.924	98.91	1.46 $\rightarrow$ 0.00	0.975	98.97	1.53 $\rightarrow$ 0.00
1	0.189	94.59	1.24 $\rightarrow$ 0.00	–	–	–
2	0.207	95.07	11.33 $\rightarrow$ 0.00	0.940	98.93	5.25 $\rightarrow$ 0.43
3	0.615	98.36	1.99 $\rightarrow$ 0.00	0.981	98.98	1.84 $\rightarrow$ 0.08
4	1.000	99.00	0.29 $\rightarrow$ 0.29	1.000	99.00	0.35 $\rightarrow$ 0.35
5	0.948	98.94	3.47 $\rightarrow$ 0.00	0.984	98.98	1.61 $\rightarrow$ 0.27
6	0.947	98.94	3.62 $\rightarrow$ 0.00	0.965	98.96	3.38 $\rightarrow$ 0.01
7	0.965	98.96	1.16 $\rightarrow$ 0.02	0.999	98.99	0.53 $\rightarrow$ 0.53
8	0.446	97.73	1.31 $\rightarrow$ 0.00	0.986	98.98	1.13 $\rightarrow$ 0.05
9	0.447	97.74	0.79 $\rightarrow$ 0.00	0.971	98.97	2.67 $\rightarrow$ 0.10

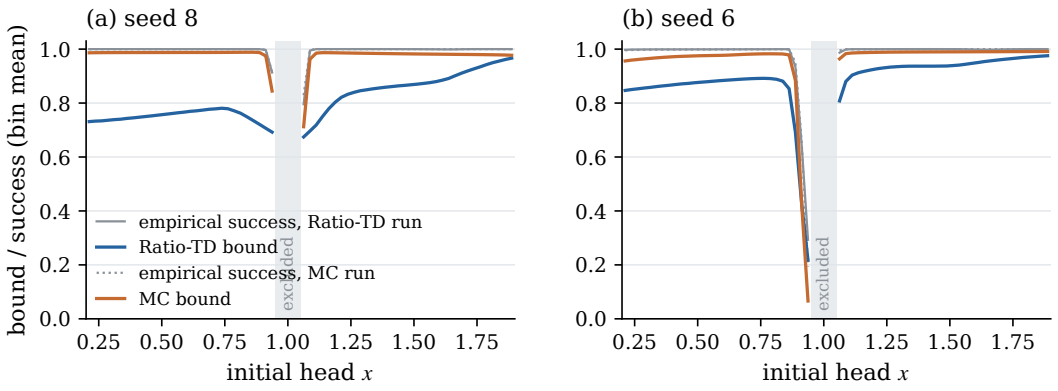

Figure 8. Wide-start Hopper: bin means over the initial head x of the learned bound and of the empirical success S_x/M, on the first accepted validation batch of each arm, for seeds 8 and 6. The shaded interval (starts inside the ceiling block) is excluded from μ. The two arms start from the same initial pair but return different policies.

D.6 Complete Candidate Comparisons and Failed Runs

Table 16 retains the full synthesis comparison across refinement modes, with all ten attempts per task and mode.

Table 17 additionally reports candidate low/high for the two robotic tasks. For each accepted feedback run, these are the minimum and maximum calibrated $q(x)$ over the 10,000 initial states in its first accepted validation batch; the table averages each statistic across its reported seeds and gives sample standard deviations. Hopper excludes seeds 2 and 4 after inspection of their low initial policy success, leaving eight seeds; PointGoal retains all ten. The full ten-seed Hopper low/high summaries are 0.866 ± 0.278 and 0.968 ± 0.047. These extrema describe predictions on sampled states, not the full initial set. Hopper's seed 2 and seed 4 have sample minima 0.125 and 0.675, while the other eight exceed 0.974. PointGoal's sample minima range from 0.280 to 0.825. This variation would be hidden by reporting only a mean candidate value.

All ten feedback seeds pass on Linear and Planar, with mean bounds 0.991 and 0.988. These near-perfect policies mainly test calibration near the local evidence ceiling.

Table 16. Complete synthesis, ten seeds per task and mode. Train (A), Calibrate (B), and Feedback (C) differ in their response to rejection; all calibrate before validation. Success and mean bound $\bar{q}$ use the first accepted validation batch; rounds summarize accepted seeds; $\pm$ denotes seed standard deviation. First counts round-one acceptances. Transitions average all ten attempts, including exhaustion. Acceptance also permits a zero bound, so bound height must be read alongside acceptance.

Task	Mode	Accept	First	Rounds	Success	$\bar{q}$	Transitions
Linear	Train	10/10	10	1.00	1.000 ± 0.000	0.991 ± 0.001	6.22×10^{7}
	Calibrate	10/10	9	1.10	1.000 ± 0.000	0.991 ± 0.001	6.73×10^{7}
	Feedback	10/10	9	1.10	1.000 ± 0.000	0.991 ± 0.001	6.76×10^{7}
Planar	Train	10/10	10	1.00	0.999 ± 0.001	0.988 ± 0.004	5.39×10^{7}
	Calibrate	10/10	10	1.00	0.999 ± 0.001	0.988 ± 0.004	5.39×10^{7}
	Feedback	10/10	10	1.00	0.999 ± 0.001	0.988 ± 0.004	5.39×10^{7}
Hopper	Train	10/10	8	1.30	0.973 ± 0.056	0.941 ± 0.097	2.12×10^{9}
	Calibrate	9/10	8	1.11	0.987 ± 0.038	0.967 ± 0.057	6.18×10^{9}
	Feedback	10/10	8	1.30	0.976 ± 0.049	0.945 ± 0.088	2.07×10^{9}
PointGoal	Train	10/10	8	1.20	0.995 ± 0.002	0.939 ± 0.010	4.61×10^{8}
	Calibrate	10/10	9	1.10	0.995 ± 0.002	0.940 ± 0.011	4.50×10^{8}
	Feedback	10/10	9	1.10	0.995 ± 0.002	0.940 ± 0.011	4.50×10^{8}

Table 17. Feedback synthesis. Hopper reports eight seeds after excluding seeds 2 and 4 for low initial policy success; PointGoal reports all ten. Candidate low/high are per-seed minima/maxima of calibrated $q(x)$ over 10,000 states at first acceptance (mean $\pm$ sample s.d. across reported seeds). These are sample extrema, not global bounds. Success is the empirical rate on the same first accepted validation batch (mean $\pm$ seed s.d.); rounds and transitions are mean rounds to first acceptance and mean total cost. Table 16 retains all ten seeds per task.

Task	Accepted	Rounds	Success	Candidate low	Candidate high	Transitions
Hopper	8/8	1.13	1.000 ± 0.001	0.982 ± 0.005	0.990 ± 0.003	9.67×10^{8}
PointGoal	10/10	1.10	0.995 ± 0.002	0.625 ± 0.170	0.984 ± 0.005	4.50×10^{8}

Tables 18–23 include both candidate-learning budgets for continuous tasks and all four budgets for Frozen Lake, before and after calibration. Each mean includes rejected candidates. Within each seed, $\bar{q}$ is the sample mean across evaluation states and $q_{.05}$ is the empirical fifth percentile. Reported means and standard deviations summarize these quantities across seeds. A high mean combined with failed verification is an unaccepted prediction, not a certified bound.

The incomplete Hopper seed. Hopper fixed-policy comparisons completed for nine seeds. No successful trajectories were observed in the initial diagnostic sample for the tenth policy (seed 2); its

Table 18. Frozen Lake: before calibration, ten seeds. Exact probability mean is 0.560. V_H is computed using dynamic programming. Means include rejected candidates. Each test uses $\delta_t = 0.005$ and exact cutoff $K \leq 443$.

Candidate	**Budget**	$\bar{q}$	V_H	**Pass**	Mean K
Ratio-TD	0.1	0.272 ± 0.030	0.045 ± 0.031	2/10	636.1
	0.25	0.269 ± 0.012	0.028 ± 0.018	6/10	410.4
	0.5	0.261 ± 0.011	0.015 ± 0.008	10/10	292.5
	1	0.257 ± 0.011	0.011 ± 0.013	10/10	203.3
Probability-TD	0.1	0.593 ± 0.019	0.849 ± 0.159	0/10	9771.0
	0.25	0.594 ± 0.010	0.923 ± 0.095	0/10	9940.3
	0.5	0.593 ± 0.007	0.957 ± 0.025	0/10	9972.3
	1	0.590 ± 0.004	0.964 ± 0.023	0/10	9998.4
MC regression	0.1	0.573 ± 0.034	0.560 ± 0.181	0/10	7949.7
	0.25	0.577 ± 0.011	0.666 ± 0.092	0/10	8798.3
	0.5	0.587 ± 0.015	0.730 ± 0.142	0/10	9368.6
	1	0.584 ± 0.013	0.743 ± 0.100	0/10	9541.5
Discounted value	0.1	0.119 ± 0.007	0.006 ± 0.013	9/10	103.7
	0.25	0.118 ± 0.005	0.000 ± 0.000	10/10	19.6
	0.5	0.116 ± 0.003	0.000 ± 0.000	10/10	18.0
	1	0.116 ± 0.002	0.000 ± 0.000	10/10	19.7
Constant	0.1	0.012 ± 0.013	0.000 ± 0.000	10/10	1.0
	0.25	0.012 ± 0.013	0.000 ± 0.000	10/10	0.7
	0.5	0.012 ± 0.013	0.000 ± 0.000	10/10	1.2
	1	0.012 ± 0.013	0.000 ± 0.000	10/10	1.5

Table 19. Frozen Lake: after calibration, ten seeds. Exact probability mean is 0.560. V_H is computed using dynamic programming. Means include rejected candidates. Each test uses $\delta_t = 0.005$ and exact cutoff $K \leq 443$.

Candidate	**Budget**	$\bar{q}$	V_H	**Pass**	Mean K
Ratio-TD	0.1	0.446 ± 0.019	0.000 ± 0.000	10/10	278.8
	0.25	0.454 ± 0.013	0.000 ± 0.000	10/10	276.1
	0.5	0.462 ± 0.008	0.000 ± 0.000	10/10	302.5
	1	0.465 ± 0.006	0.000 ± 0.000	10/10	310.8
Probability-TD	0.1	0.483 ± 0.002	0.000 ± 0.000	10/10	269.4
	0.25	0.484 ± 0.002	0.000 ± 0.000	10/10	273.0
	0.5	0.484 ± 0.002	0.000 ± 0.000	10/10	272.8
	1	0.484 ± 0.002	0.000 ± 0.000	10/10	278.6
MC regression	0.1	0.474 ± 0.018	0.000 ± 0.000	10/10	258.5
	0.25	0.482 ± 0.002	0.000 ± 0.000	10/10	253.0
	0.5	0.482 ± 0.003	0.000 ± 0.000	10/10	274.1
	1	0.483 ± 0.002	0.000 ± 0.000	10/10	274.8
Discounted value	0.1	0.460 ± 0.003	0.000 ± 0.000	10/10	260.5
	0.25	0.460 ± 0.002	0.000 ± 0.000	10/10	248.7
	0.5	0.460 ± 0.003	0.000 ± 0.000	10/10	258.4
	1	0.460 ± 0.003	0.000 ± 0.000	10/10	237.8
Constant	0.1	0.000 ± 0.000	0.000 ± 0.000	10/10	0.0
	0.25	0.000 ± 0.000	0.000 ± 0.000	10/10	0.0
	0.5	0.000 ± 0.000	0.000 ± 0.000	10/10	0.0
	1	0.000 ± 0.000	0.000 ± 0.000	10/10	0.0

Table 20. Linear: fixed-policy candidate comparison, ten seeds. Pre/post denote before/after calibration. $\widehat{V}$ compares predictions with finite reference estimates and is not the true violation mass. Values include rejected candidates. Each test uses $K \leq 83$.

Candidate	**Budget**	**Stage**	$\bar{q}$	$q_{.05}$	$\widehat{V}$	**Pass**	Mean K
Ratio-TD	0.25	pre	0.987 ± 0.010	0.981	0.000	2/10	4546.3
	0.25	post	0.992 ± 0.000	0.992	0.000	10/10	41.6
	1	pre	0.992 ± 0.009	0.986	0.000	1/10	6989.6
	1	post	0.992 ± 0.000	0.992	0.000	10/10	30.5
Probability-TD	0.25	pre	0.999 ± 0.000	0.999	0.000	0/10	10000.0
	0.25	post	0.993 ± 0.000	0.993	0.000	10/10	49.2
	1	pre	0.999 ± 0.000	0.999	0.000	0/10	10000.0
	1	post	0.993 ± 0.000	0.993	0.000	10/10	42.5
MC regression	0.25	pre	0.999 ± 0.000	0.999	0.000	0/10	10000.0
	0.25	post	0.993 ± 0.000	0.993	0.000	10/10	3.1
	1	pre	0.999 ± 0.000	0.999	0.000	0/10	10000.0
	1	post	0.993 ± 0.000	0.993	0.000	10/10	9.2
Discounted value	0.25	pre	0.963 ± 0.001	0.960	0.000	10/10	0.0
	0.25	post	0.992 ± 0.000	0.992	0.000	10/10	0.0
	1	pre	0.962 ± 0.001	0.960	0.000	10/10	0.0
	1	post	0.992 ± 0.000	0.992	0.000	10/10	0.0
Constant	0.25	pre	0.900 ± 0.000	0.900	0.000	10/10	0.0
	0.25	post	0.900 ± 0.000	0.900	0.000	10/10	0.0
	1	pre	0.900 ± 0.000	0.900	0.000	10/10	0.0
	1	post	0.900 ± 0.000	0.900	0.000	10/10	0.0

Table 21. Planar: fixed-policy candidate comparison, ten seeds. Pre/post denote before/after calibration. $\widehat{V}$ compares predictions with finite reference estimates and is not the true violation mass. Values include rejected candidates. Each test uses $K \leq 83$.

Candidate	**Budget**	**Stage**	$\bar{q}$	$q_{.05}$	$\widehat{V}$	**Pass**	Mean K
Ratio-TD	0.25	pre	0.981 ± 0.015	0.979	0.012	4/10	4324.0
	0.25	post	0.988 ± 0.004	0.985	0.000	8/10	62.0
	1	pre	0.976 ± 0.012	0.972	0.000	6/10	772.0
	1	post	0.988 ± 0.004	0.985	0.000	7/10	71.0
Probability-TD	0.25	pre	0.999 ± 0.000	0.999	0.128	0/10	10000.0
	0.25	post	0.988 ± 0.004	0.985	0.000	8/10	56.4
	1	pre	0.999 ± 0.000	0.999	0.180	0/10	10000.0
	1	post	0.989 ± 0.004	0.985	0.000	8/10	62.4
MC regression	0.25	pre	0.998 ± 0.002	0.996	0.066	0/10	9731.0
	0.25	post	0.988 ± 0.004	0.985	0.000	9/10	55.3
	1	pre	0.998 ± 0.001	0.998	0.066	0/10	10000.0
	1	post	0.988 ± 0.004	0.985	0.000	8/10	51.6
Discounted value	0.25	pre	0.970 ± 0.004	0.967	0.000	10/10	4.1
	0.25	post	0.988 ± 0.004	0.985	0.000	9/10	41.5
	1	pre	0.969 ± 0.004	0.967	0.000	10/10	2.6
	1	post	0.988 ± 0.004	0.985	0.000	9/10	39.9
Constant	0.25	pre	0.889 ± 0.011	0.889	0.000	10/10	0.0
	0.25	post	0.889 ± 0.011	0.889	0.000	10/10	0.0
	1	pre	0.889 ± 0.011	0.889	0.000	10/10	0.0
	1	post	0.889 ± 0.011	0.889	0.000	10/10	0.0

Table 22. Hopper: fixed-policy candidate comparison, nine completed seeds; the interrupted seed with no initial diagnostic successes is described in the text. Pre/post denote before/after calibration. $\widehat{V}$ compares predictions with finite reference estimates and is not the true violation mass. Values include rejected candidates. Each test uses $K \leq 83$.

Candidate	**Budget**	**Stage**	$\bar{q}$	$q_{.05}$	$\widehat{V}$	**Pass**	Mean K
Ratio-TD	0.25	pre	0.936 ± 0.075	0.921	0.010	5/9	476.0
	0.25	post	0.966 ± 0.057	0.953	0.000	9/9	36.8
	1	pre	0.932 ± 0.077	0.926	0.014	6/9	1082.4
	1	post	0.906 ± 0.165	0.887	0.000	9/9	28.7
Probability-TD	0.25	pre	0.915 ± 0.151	0.903	0.000	9/9	0.1
	0.25	post	0.968 ± 0.058	0.954	0.000	9/9	48.4
	1	pre	0.910 ± 0.151	0.901	0.000	9/9	0.0
	1	post	0.969 ± 0.057	0.955	0.000	9/9	40.1
MC regression	0.25	pre	0.982 ± 0.032	0.978	0.051	1/9	7101.2
	0.25	post	0.969 ± 0.055	0.956	0.000	9/9	51.0
	1	pre	0.979 ± 0.039	0.976	0.034	0/9	4601.7
	1	post	0.969 ± 0.055	0.956	0.000	9/9	55.3
Discounted value	0.25	pre	0.484 ± 0.100	0.451	0.000	9/9	0.0
	0.25	post	0.967 ± 0.061	0.953	0.000	9/9	34.3
	1	pre	0.475 ± 0.099	0.445	0.000	9/9	0.0
	1	post	0.967 ± 0.062	0.952	0.000	9/9	34.4
Constant	0.25	pre	0.860 ± 0.091	0.860	0.000	9/9	0.0
	0.25	post	0.854 ± 0.110	0.854	0.000	9/9	0.0
	1	pre	0.860 ± 0.091	0.860	0.000	9/9	0.0
	1	post	0.854 ± 0.110	0.854	0.000	9/9	0.0

Table 23. PointGoal: fixed-policy candidate comparison, ten seeds. Pre/post denote before/after calibration. $\widehat{V}$ compares predictions with finite reference estimates and is not the true violation mass. Values include rejected candidates. Each test uses $K \leq 83$.

Candidate	**Budget**	**Stage**	$\bar{q}$	$q_{.05}$	$\widehat{V}$	**Pass**	Mean K
Ratio-TD	0.25	pre	0.958 ± 0.011	0.889	0.041	0/10	2001.8
	0.25	post	0.940 ± 0.017	0.885	0.002	10/10	52.1
	1	pre	0.953 ± 0.008	0.894	0.013	0/10	928.2
	1	post	0.939 ± 0.019	0.889	0.002	10/10	49.4
Probability-TD	0.25	pre	0.990 ± 0.006	0.958	0.213	0/10	8099.4
	0.25	post	0.944 ± 0.015	0.861	0.002	10/10	50.6
	1	pre	0.992 ± 0.004	0.977	0.180	0/10	7904.4
	1	post	0.947 ± 0.012	0.871	0.003	10/10	56.7
MC regression	0.25	pre	0.991 ± 0.007	0.979	0.381	0/10	9333.1
	0.25	post	0.943 ± 0.017	0.862	0.002	9/10	51.8
	1	pre	0.991 ± 0.007	0.964	0.280	0/10	8583.2
	1	post	0.947 ± 0.016	0.865	0.002	10/10	56.3
Discounted value	0.25	pre	0.928 ± 0.013	0.861	0.005	4/10	120.2
	0.25	post	0.947 ± 0.013	0.876	0.002	10/10	51.3
	1	pre	0.922 ± 0.009	0.863	0.002	5/10	78.2
	1	post	0.949 ± 0.013	0.875	0.001	9/10	54.8
Constant	0.25	pre	0.836 ± 0.019	0.836	0.000	10/10	16.6
	0.25	post	0.420 ± 0.241	0.420	0.000	10/10	0.6
	1	pre	0.836 ± 0.019	0.836	0.000	10/10	16.0
	1	post	0.420 ± 0.241	0.420	0.000	10/10	0.5

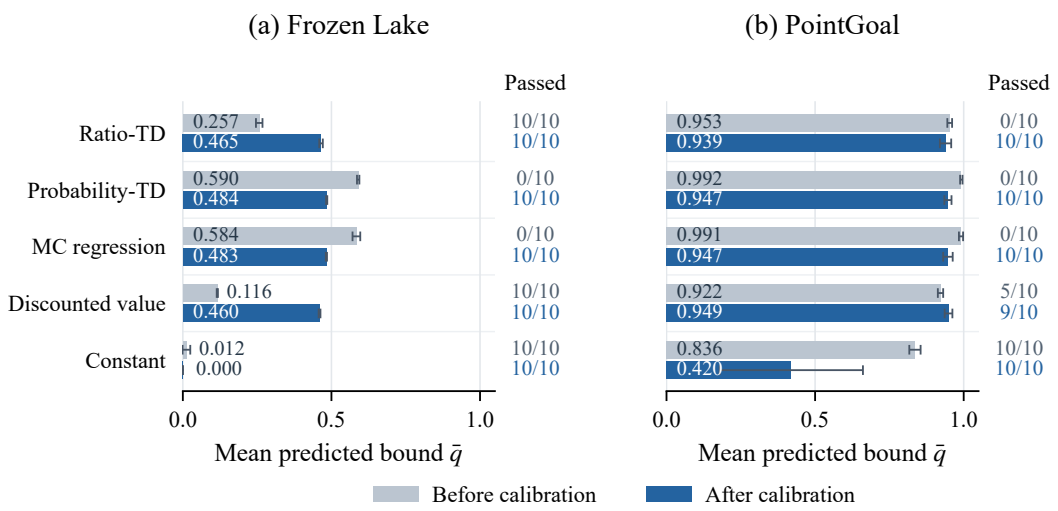

Figure 9. Calibration changes bound height and verification outcome. Bars show mean predictions at full learning budget, including rejected runs; whiskers show seed standard deviations. *Passed* counts accepted runs out of ten. Higher predictions need not pass. Appendix D.6 gives full results.

comparison was stopped after twelve rows had completed, partway through the full-budget Ratio-TD run. Completed learned candidates before calibration had means between 0.005 and 0.65 and all failed verification. Their completed calibrated versions collapsed to zero and passed with $K = 0$, as did the zero constant; the fixed-policy comparisons set predictions below 0.02 to zero (Appendix C.3), whereas the synthesis runs use no such floor, which is why the same seed's calibration-only synthesis attempt below is rejected instead. Because the full comparison was interrupted, these partial rows are not pooled with the nine-seed table.

The complete synthesis comparison retains all ten Hopper seeds. In seed 2, calibration-only attempts reject all 10000 states in each of ten rounds: a strictly positive sigmoid-based candidate lacks support wherever the observed success count is zero. The procedure returns UNKNOWN. Both policy-training modes recover by round three. The remaining partially successful initial seed (seed 4) has diagnostic success 0.887; it is also retained.

States with low candidate values. Post hoc analysis of the accepted validation batches associates low predictions with lower empirical success. Correlations between $q(x)$ and success frequencies from the same 1000 rollouts per state range from 0.5 to 0.7 on PointGoal and from 0.97 to 1.0 on Hopper. In the weaker Hopper runs, low predictions occur near the left edge of the initial region, where reaching the target requires a longer path beneath the ceiling. PointGoal stores observations rather than positions in these records; lidar-derived distance proxies associate low predictions with greater distance to the goal. For PointGoal seed 2, the lowest-prediction 1% of states have mean $q = 0.70$ and empirical success 0.88. The global calibration shift can further lower predictions at these difficult starts. These associations are measured on the accepted validation batches; their certificate remains the distributional statement of Theorem 1.

Near-perfect policies do not guarantee test acceptance. Linear's calibrated comparisons all pass, while Planar's learned candidates pass in seven to nine of ten seeds depending on method and budget. Across Planar's eight learner–budget configurations, $66/80$ tests pass; calibration data are shared within each seed, and 12 of the 14 failures occur in seeds 1 and 5. The Planar probability estimates are near one, and small changes in the number of failures move the discrete local confidence limit across a tight candidate. A finite reference set can show no observed overestimation while the verifier still rejects. This is consistent with the distinction between V_H and r in (7).

PointGoal seed variation. Table 24 reports all thirty synthesis attempts and Figure 10 separates policy success, average bound, and its fifth percentile. These statistics describe variability over initial states and seeds. The table and figure use diagnostic evaluations of the accepted outputs; Tables 11 and 16 use the accepted validation batches.

In the fixed-policy comparison, define $\Delta_s = q_{.05}^{\mathrm{Ratio},s} - \max_{m\in\{\mathrm{Probability,MC,Discounted}\}} q_{.05}^{m,s}$ after calibration. At quarter candidate budget, $\Delta_s > 0$ in 9/10 seeds, with positive differences 0.001–0.029; seed 2 has $\Delta_2 = -0.032$. At full budget, 8/10 differences are positive; the exceptions are seed 2 (-0.017) and seed 5 (-0.007). Mean fifth percentiles for Ratio-TD, Probability-TD, MC, and discounted value are respectively $(0.885, 0.861, 0.862, 0.876)$ at quarter budget and $(0.889, 0.871, 0.865, 0.875)$ at full budget. These comparisons include rejected candidates (quarter-budget MC seed 5 and full-budget discounted-value seed 5); fifth percentiles summarize the learned bound distribution.

Table 24. PointGoal synthesis, all thirty attempts. Success and bound statistics use the separate diagnostic evaluation of each accepted output; aggregate validation-batch results appear in Table 16. The fifth percentile is computed within a seed across states. K_1 is the first validation count; acceptance uses $K \leq 74$. Each mode starts from its seed's common policy checkpoint, but its rollout estimates and learning randomness can differ.

Seed	Mode	Round	Success	$\bar{q}$	$q_{.05}$	K_1	Transitions
0	A	1	0.993	0.922	0.862	37	4.55×10^8
0	B	1	0.992	0.917	0.859	19	4.54×10^8
0	C	1	0.996	0.922	0.864	27	4.54×10^8
1	A	1	0.995	0.936	0.877	61	4.36×10^8
1	B	1	0.995	0.941	0.892	42	4.36×10^8
1	C	1	0.997	0.940	0.899	45	4.36×10^8
2	A	2	0.998	0.946	0.904	92	5.42×10^8
2	B	2	0.996	0.945	0.868	88	5.40×10^8
2	C	2	0.993	0.942	0.864	91	5.45×10^8
3	A	1	0.994	0.935	0.862	42	4.47×10^8
3	B	1	0.995	0.938	0.885	41	4.46×10^8
3	C	1	0.994	0.935	0.852	49	4.46×10^8
4	A	2	0.994	0.943	0.878	75	5.60×10^8
4	B	1	0.995	0.953	0.915	53	4.51×10^8
4	C	1	0.994	0.952	0.910	66	4.52×10^8
5	A	1	0.998	0.956	0.930	42	4.35×10^8
5	B	1	0.998	0.955	0.929	51	4.35×10^8
5	C	1	0.997	0.953	0.925	44	4.34×10^8
6	A	1	0.995	0.932	0.888	52	4.34×10^8
6	B	1	0.996	0.939	0.897	55	4.34×10^8
6	C	1	0.997	0.938	0.902	49	4.33×10^8
7	A	1	0.993	0.953	0.922	28	4.32×10^8
7	B	1	0.998	0.953	0.917	33	4.32×10^8
7	C	1	0.999	0.953	0.920	31	4.32×10^8
8	A	1	0.995	0.925	0.844	63	4.37×10^8
8	B	1	0.994	0.925	0.853	54	4.37×10^8
8	C	1	0.995	0.928	0.840	49	4.37×10^8
9	A	1	0.996	0.940	0.882	51	4.33×10^8
9	B	1	0.996	0.938	0.879	62	4.34×10^8
9	C	1	0.996	0.937	0.894	67	4.34×10^8

D.7 Calibration Budget and Seed Variation

We vary fitting and independent shift data over five budgets on Frozen Lake, using seeds 0–2 and Ratio-TD, Probability-TD, and MC. Within each seed the policy and full candidate-trajectory budget are held fixed. Candidate fitting is repeated across budget configurations; the pre-calibration weights are not held identical. Fitting uses the Frozen Lake protocol in Appendix C.3, with 1000 rollouts per shift state. All configurations use fresh validation with $N = 10000$, $M = 1000$, $\beta = 10^{-3}$, $\epsilon = 0.05$, $\delta_t = 0.005$, and cutoff 443.

At the smallest budget (256 fitting states with 64 rollouts each and 256 shift states), fitting and shift calibration cost 1.32×10^7 active transitions on average. Ratio-TD yields 0.425 ± 0.019, Probability-TD 0.403 ± 0.075, and MC 0.407 ± 0.076. The paired values are $(0.425, 0.407, 0.444)$ for Ratio-TD,

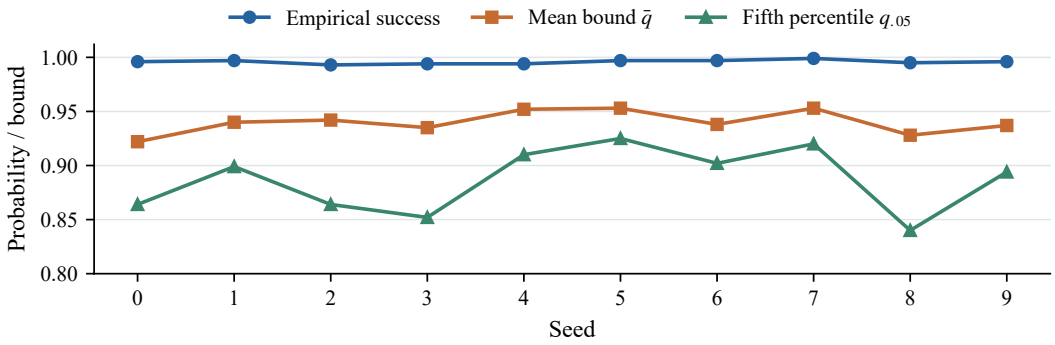

Figure 10. PointGoal complete-feedback outputs, using the diagnostic evaluation summarized in Table 24. Every seed is accepted; seed 2 requires two rounds. The gap between the mean and fifth percentile of q shows that the learned function gives different values across starting states.

Table 25. Frozen Lake calibration budgets, seeds 0–2. Entries are mean calibrated bound ± sample standard deviation. Costs count fitting and shift-calibration transitions. Every post-calibration run passes (45/45), with exact $V_H = 0$. The validation budget is fixed at $N = 10000$, $M = 1000$, $\delta_t = 0.005$.

Fitting + shift data	Transitions	Ratio-TD	Probability-TD	MC
$256\times64 + 256\times1000$	1.32×10^7	0.425 ± 0.019	0.403 ± 0.075	0.407 ± 0.076
$512\times128 + 512\times1000$	2.80×10^7	0.441 ± 0.010	0.467 ± 0.004	0.464 ± 0.002
$1024\times256 + 1024\times1000$	6.27×10^7	0.462 ± 0.006	0.482 ± 0.003	0.483 ± 0.002
$2048\times256 + 1024\times1000$	7.51×10^7	0.466 ± 0.005	0.484 ± 0.001	0.483 ± 0.001
$4096\times256 + 2048\times1000$	14.95×10^7	0.464 ± 0.002	0.480 ± 0.004	0.481 ± 0.004

$(0.435, 0.317, 0.457)$ for Probability-TD, and $(0.435, 0.320, 0.464)$ for MC. Thus Ratio-TD exceeds either direct learner in only one of three paired seeds: its aggregate advantage comes from avoiding the pronounced decrease in seed 1. This three-seed observation is evidence of smaller variation in this setting, not a general stability result.

From the next tier onward, both direct learners have higher means. At the nominal budget of 2048 fitting states with 256 rollouts and 1024 shift states, means are 0.466, 0.484, and 0.483; increasing to the largest budget does not improve them. All 45 post-calibration runs have exact $V_H = 0$ and rejection counts between 100 and 423. Before calibration, Ratio-TD passes all 15 runs, whereas both direct learners fail all 15. These observations separate initial conservatism, calibrated bound height, and acceptance.

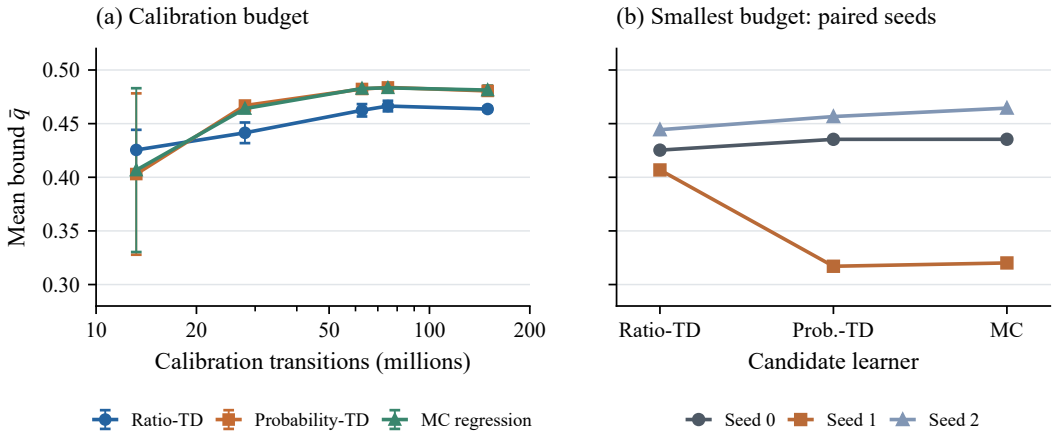

Figure 11. Frozen Lake calibration-budget study. Left: mean bound with sample-standard-deviation bars across three seeds; the horizontal axis is logarithmic. Right: paired results at the smallest budget, exposing the direct learners' lower bounds in seed 1. All 45 calibrated candidates pass and have exact $V_H = 0$. Error bars describe seed variation, not confidence intervals.

Calibration costs span 1.32×10^7–1.49×10^8 transitions, approximately 3%–31% of the roughly 4.8×10^8–4.9×10^8 validation cost. The nominal budget costs 7.51×10^7 transitions, about 16% of validation. These are calibration costs alone; candidate learning and validation are additional. Since all methods pass at the smallest tested budget, this sweep does not establish a lower minimum certification cost for Ratio-TD.

D.8 Sensitivity Analyses

Table 26 keeps the accepted PointGoal predictions and rollout counts fixed, then recomputes the decision under one changed parameter. Its N sweep subsamples without replacement from the original accepted batches. These are dependent, selected-batch diagnostics, not fresh draws estimating an unconditional acceptance probability. In particular, the full-N row passes by construction because the source batches were accepted.

Table 26. PointGoal: offline re-evaluation of fixed accepted batches from mode A. Only the test parameter changes. For N, each of ten seeds contributes twenty subsamples without replacement; these 200 outcomes are dependent and are not new certification attempts. The full-batch row is consequently repeated twenty times per seed. All other settings are held at $N = 10{,}000$, $M = 1000$, $\beta = 10^{-3}$, $\epsilon = 0.01$, $\delta_t = 0.005$.

Parameter	**Value**	Mean $K \pm$ s.d.	k_*	k_*/N	**Accept**
β	1×10^{-5}	70.0 ± 12.9	74	0.74%	6/10
	1×10^{-4}	56.4 ± 12.1	74	0.74%	10/10
	1×10^{-3}	47.1 ± 10.2	74	0.74%	10/10
	1×10^{-2}	37.7 ± 9.2	73	0.73%	10/10
ϵ	0.005	47.1 ± 10.2	32	0.32%	1/10
	0.01	47.1 ± 10.2	74	0.74%	10/10
	0.02	47.1 ± 10.2	164	1.64%	10/10
	0.05	47.1 ± 10.2	443	4.43%	10/10
δ_t	1×10^{-20}	47.1 ± 10.2	22	0.22%	0/10
	1×10^{-3}	47.1 ± 10.2	70	0.70%	10/10
	0.005	47.1 ± 10.2	74	0.74%	10/10
	0.01	47.1 ± 10.2	77	0.77%	10/10
	0.05	47.1 ± 10.2	83	0.83%	10/10
N	500	2.4 ± 1.7	-1	—	0/200
	1000	4.8 ± 2.3	2	0.20%	27/200
	2000	9.6 ± 3.4	9	0.45%	101/200
	5000	23.6 ± 6.1	32	0.64%	188/200
	10000	47.1 ± 10.2	74	0.74%	200/200

Changing N changes the outer rejection-count cutoff, while changing M changes the local confidence limits. Reducing β makes local lower limits smaller and changes the outer threshold through $(1-\beta)\epsilon$; its effect is not confined to the local count. Changing ϵ changes the claim being certified. Changing δ_t changes confidence, not the true statewise probability.

Table 27 instead refits the calibration targets and shift under each setting and performs new validation on three fixed policy seeds. The values at $N = 2000, 5000, 10000, 20000$ are $0.909, 0.925, 0.932, 0.932$. This sweep also changes α for the two smaller sample sizes, so the gain cannot be attributed to N alone. All these configurations pass in three of three seeds.

At $M = 200, 500, 1000, 2000$, the best-case local limits $\beta^{1/M}$ are approximately $0.966, 0.986, 0.993, 0.997$, while calibrated means are $0.884, 0.922, 0.932, 0.940$. The remaining gap reflects the learned function and the global shift. Tightening β from 10^{-3} to 10^{-5} with recalibration reduces the mean from 0.932 to 0.924, while leaving all three tests accepted. Tightening ϵ to 0.005 yields 0.908 and two acceptances; the rejected seed has $K = 34$ against cutoff 32. Values $\epsilon = 0.02$ and 0.05 give means 0.951 and 0.966 with three acceptances each. The mean is unchanged between $\delta_t = 0.001$ and 0.01 on these three seeds.

Lower success and smaller conditional discounts. Additional single-seed Frozen Lake experiments use an earlier calibration configuration and are exploratory. A lower-success policy has exact

Table 27. PointGoal: separate recalibration and validation under each setting, using three fixed policy seeds and Ratio-TD at full learning budget. Means include all three seeds, including rejected candidates. Unlisted parameters equal the reference setting in Table 26. α is a calibration design parameter; the theorem does not require $\alpha = \epsilon/2$. Each row is an individual configuration, with no joint confidence claim across this sweep.

Changed setting	α	$\bar{q}$	$q_{.05}$	K (three seeds)	k_*	**Pass**
Reference	0.005	0.932 ± 0.007	0.867	32/59/42	74	3/3
$N = 2000$	0.0025	0.909 ± 0.012	0.829	5/6/2	9	3/3
$N = 5000$	0.0035	0.925 ± 0.007	0.855	14/20/23	32	3/3
$N = 20000$	0.005	0.932 ± 0.008	0.868	61/107/89	164	3/3
$M = 200$	0.005	0.884 ± 0.003	0.803	28/52/43	74	3/3
$M = 500$	0.005	0.922 ± 0.006	0.856	29/62/53	74	3/3
$M = 2000$	0.005	0.940 ± 0.008	0.877	30/60/48	74	3/3
$\beta = 10^{-5}$	0.005	0.924 ± 0.008	0.858	44/58/45	74	3/3
$\beta = 10^{-4}$	0.005	0.926 ± 0.004	0.860	34/49/52	74	3/3
$\beta = 10^{-2}$	0.005	0.937 ± 0.007	0.875	33/60/46	73	3/3
$\epsilon = 0.005$	0.0025	0.908 ± 0.006	0.828	21/34/30	32	2/3
$\epsilon = 0.02$	0.01	0.951 ± 0.003	0.897	65/84/109	164	3/3
$\epsilon = 0.05$	0.025	0.966 ± 0.002	0.924	223/198/235	443	3/3
$\delta_t = 0.001$	0.005	0.931 ± 0.010	0.862	42/53/40	70	3/3
$\delta_t = 0.01$	0.005	0.931 ± 0.007	0.859	34/56/44	77	3/3

mean $p_H = 0.306$; calibrated means are approximately 0.12 for Ratio-TD, 0.13 for Probability-TD, 0.15 for MC, and 0.13–0.14 for discounted value. All reported candidates have exact $V_H = 0$ and pass.

With discount reduced from 0.95 to 0.9, calibrated Ratio-TD averages about 0.37 at quarter budget and 0.44 at full budget, while the direct learners remain around 0.48. The observed larger loss at low budget is consistent with sensitivity to the auxiliary denominator.

D.9 Transition Accounting and Wall-Clock Measurements

Each attempt is charged for policy learning, checkpoint selection, candidate learning, calibration, validation, and diagnostics through its stopping point. Costs count active environment transitions through target/failure termination; padding of terminated trajectories is not counted as additional interaction. The theoretical worst-case verifier cost is NMH per round. Table 28 provides a representative PointGoal decomposition, while Table 16 reports all-seed average totals.

Table 28. PointGoal mode A, seed 0, first-round acceptance: transition accounting. Components are rounded and approximately sum to 4.55×10^8. Checkpoint-selection diagnostics are included in the policy-learning entry.

Stage	Transitions
Policy pretraining and checkpoint selection	3.45×10^8
Candidate learning	8.2×10^5
Calibration (fitting and shift)	2.2×10^7
One verification batch	8.7×10^7
Additional diagnostics	1.2×10^6
Total	4.55×10^8

All experiments run as single-GPU jobs on NVIDIA H20 GPUs (96 GB), one GPU per run. Table 29 gives end-to-end wall-clock times per seed and mode for the complete synthesis runs, including the shared initialization (policy pretraining, candidate learning, calibration) whose cost every mode inherits.

Hopper validation costs about 6.5×10^8 transitions for the near-perfect policies; averaging the accepted feedback batches across all ten seeds, including the weaker policies, gives 8.3×10^8. First-round accepted attempts typically total 8.4–8.9×10^8 transitions. The calibration-only attempt for

Table 29. Wall-clock time per seed and mode on one H20 (minutes), complete synthesis runs of Table 16, including initialization. Ranges are over ten seeds; the Hopper maximum is seed 2, which passes in round three. Seed-0 stage times are the run's cumulative policy training / calibration / validation; Hopper seed 0 passes in round two, so its calibration and validation entries cover two rounds.

Task	Rollouts	Train (A)	Feedback (C)	Seed-0 stages
Linear	compact	1.5–1.8	1.5–3.2	0.3 / 0.5 / 0.2
Planar	compact	1.5–2.0	1.5–2.0	0.4 / 0.5 / 0.2
Hopper	compact	15–62	15–62	1.8 / 15.1 / 6.0
PointGoal	scan	48–77	48–77	19.8 / 11.1 / 15.8

seed 2, whose initial diagnostic sample contains no successful trajectories, exhausts ten rounds and costs approximately 5.2×10^{10} transitions.

The principal Hopper, Linear, and Planar results use compact rollout batches that remove terminated trajectories every ten steps and pool active tails. PointGoal synthesis and Frozen Lake use fixed-length scans. PointGoal fixed-policy comparisons use scans for seeds 0–8 and compaction for seed 9. A separate implementation benchmark reduced a PointGoal validation batch from 941 seconds to about 70 seconds; compact Hopper validation takes about 240 seconds in the reported benchmark. These are implementation timings on H20 hardware, not end-to-end synthesis times.

Compaction changes only execution scheduling. In both modes every rollout owns its own random key: one key per start state is split into M rollout keys, and each step splits a lane's key into a new key, a policy-sampling key, and an environment key; in compact mode the key travels with the lane when terminated lanes are dropped and active tails are re-batched, so it never depends on the lane's position in a batch. The policy network is a per-sample MLP and the environment step is vectorized over independent states, with no reduction across lanes, so the law of a trajectory conditional on its key is the same in both modes. Remaining differences are floating-point: XLA and MJX select different kernels for different batch shapes. Counts are identical on Linear and Planar; in the PointGoal benchmark (512 states, 1000 rollouts each), 237 states differ by at most 11 successes, with a net difference of -55 successes out of 5.1×10^5 and a sign that varies across batch sizes. The implementation preserves independent random streams in both modes; the formal guarantee assumes the iid rollout law of Assumption 1, and the comparison above measures the numerical differences between execution modes. Each synthesis run uses one mode throughout; the paired calibration evaluation of Appendix A.4 uses compaction for all tasks.

D.10 Scope of the Empirical Evidence

The ten-seed studies evaluate certificate informativeness on the specified tasks, initial distributions, budgets, and one policy optimizer; refinement benefits vary across seeds. The strong PointGoal checkpoints leave little room to compare refinement mechanisms. The low-dimensional tasks have limited variation in success across states; Frozen Lake supplies the exact-reference evidence for heterogeneous probabilities. The PointGoal synthesis configuration (validation batches of $N = 10^4$, 800 pretraining iterations) was settled after development runs on the same seeds with smaller validation batches and shorter pretraining, whose certificates were valid but lower; the acceptance rates reported here are for the final protocol. The wide-start Hopper study (Appendix D.5) was added after the main results to test state dependence under heterogeneous success. Its μ was fixed from the existing learning-start range before any run; the MC arm was run after the Ratio-TD results, and all runs of both arms are reported.

The RAPC results in Table 13 use the CP verifier, with the benchmark protocol and guarantee semantics specified in Appendix D.11. The RAPC entries come from the same single-H20 Linear and Planar runs as Table 16. No physical-robot or controlled simulator-to-real transfer experiment is reported.

D.11 RASM/LogRASM Benchmark

Table 13 compares certificate-based policy synthesis with RAPC outputs accepted by the CP verifier of Section 4.1. The RASM and LogRASM runs were conducted on the same H20 hardware with a

30-minute budget per run. The RAPC runs are the single-H20 jobs of Appendix D.9, which finish in 1.5–3.2 minutes per seed.

RAPC uses the same verification parameters as the complete control-synthesis experiments: $N = 10{,}000$, $M = 1000$, $H = 400$, $\epsilon = 0.01$, $\beta = 10^{-3}$, and total error budget $\delta = 0.05$. Each seed permits at most $R = 10$ rounds, allocating $\delta_t = \delta/R = 0.005$ to each round. Acceptance requires a complete validation batch with $K \leq 74$ (Appendix C.2).

RASM and LogRASM used their authors' official implementations with default hyperparameters. On Linear, their target probability started at 0.8 and was increased until verification could no longer be completed within the per-run budget; the table reports the highest verified value. On Planar, the target was lowered to 0.1, and "Timed out" means that at least six of ten runs could not verify even this target within the budget. It is a time-budget outcome, not a proof that no certificate exists. Only Linear and Planar were tested with these baselines. Both use the LogRASM code base; RASM is obtained by disabling its logarithmic certificate. Verification starts from a mesh of 0.01 on the state box $[-1, 1]^d$, i.e., 200 intervals per dimension, and refines it by counterexample-guided iterations without a cap on their number; a run fails only by the 30-minute timeout. On Planar, the timed-out runs were still refining; for seed 1, at timeout RASM was checking the expected-decrease condition on 8.25×10^8 grid samples and LogRASM on 3.30×10^8, against 8×10^6 cells in the initial mesh. The robotic tasks were not attempted with these baselines: they require an analytical or Lipschitz-bounded model of the dynamics and a discretization of the state space, neither of which the rollout interface of Section D.1 supplies. RAPC operates through sampled state–rollout blocks with at most NMH transitions per round at fixed N, M, H; the cost of each transition and the difficulty of learning depend on the system.

For each RAPC seed, low/high are the minimum and maximum calibrated $q(x)$ over the 10,000 initial states in its first accepted validation batch. Table 13 reports the mean and sample standard deviation of each statistic across ten seeds. These are sample extrema of CP-accepted bound functions, not lower and upper confidence limits for the success probability or verified extrema over a continuous initial set. CP acceptance certifies the function under the distributional guarantee, whereas RASM and LogRASM establish uniform initial-set bounds (Appendix B.7). Linear and Planar use the dynamics and sets given in Section D.1. The robotic renderings in Figure 4 illustrate the task types; the precise dynamics and reset laws are specified in Section D.1.