SNAPSHOT ENTROPY: EXPLORATION WITH COVERAGE AND CERTAINTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Maximum state-entropy exploration seeks broad state visitation, but the standard visitation distribution pools variation across both time and independent executions. Consequently, the same pooled coverage can arise either from a reproducible, time-structured route or from dispersion across runs. We make this distinction explicit by sampling a time T according to the visitation weights, independently of a run, and letting X denote the agent's state at that time. Then $H(X)$ is the *pooled state entropy*, while the chain rule $H(X) = H(X \mid T) + I(T; X)$ separates it into two components. We call $H(X \mid T)$ the *snapshot entropy*: it measures the uncertainty across executions that remains once elapsed time is known.

We study *exploration with certainty*, which maximizes pooled state entropy subject to a snapshot-entropy budget. This budget limits time-aligned disagreement across runs and guarantees that expected within-run visitation entropy is at least the pooled state entropy minus the budget. We derive an exact policy-gradient identity for this objective and obtain Certainty Policy Gradient (Certainty PG), a model-free method that estimates time-indexed state distributions from synchronized rollouts.

We characterize the structure and limits of exploration with certainty. Deterministic policies suffice to minimize snapshot entropy; selecting a high-coverage certain route is NP-hard; and stochastic transitions impose an unavoidable uncertainty floor computable by dynamic programming. On Ant, a continuous-control benchmark used in prior maximum-entropy exploration work, synchronized rollouts expose same-time dispersion hidden by pooled occupancy: Certainty PG reduces this dispersion, though at a cost in coverage. On clean Rooms-8, where broad reproducible routes exist, Certainty PG retains 98.0% of Pooled PG's pooled entropy while increasing single-run all-state coverage from 31.5% to 99.8%. A trajectory-entropy objective achieves similarly high coverage but substantially greater time-aligned uncertainty, demonstrating that single-run coverage and repeatability are distinct objectives. On noisy Rooms-8, it approaches the uncertainty floor, while navigation stress tests show that excessive concentration can produce incomplete routes.

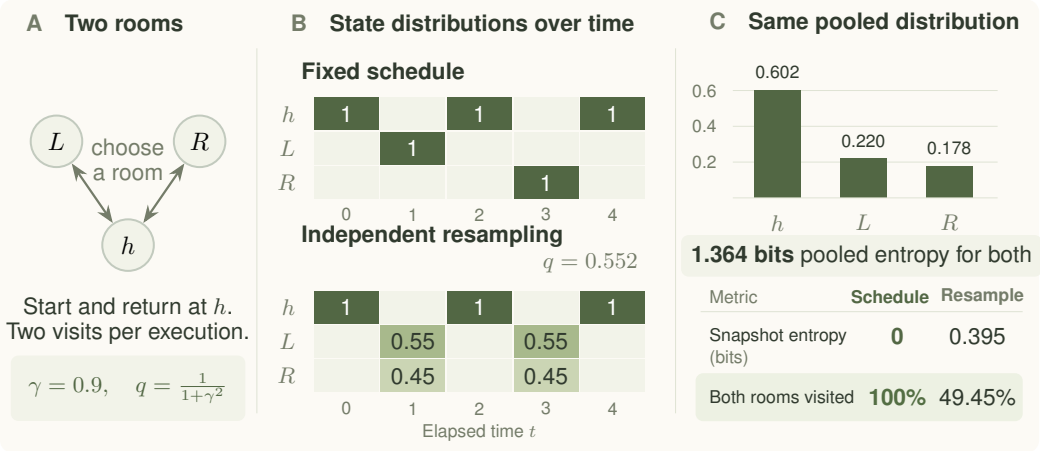
1 INTRODUCTION

Reward-free exploration asks an agent to collect useful experience before a downstream task is known (Jin et al., 2020). Without a task reward, a natural objective is *coverage*: visit the state space broadly enough to support a range of future tasks. Maximum state-entropy exploration formalizes this principle by maximizing the entropy of the state-visitation distribution (Hazan et al., 2019; Seo et al., 2021; Mutti et al., 2021). This distribution, however, averages over both time and trajectory randomness. We call its entropy *pooled state entropy*. High pooled entropy therefore does not reveal whether an agent follows a broad, reproducible route on each run or whether different runs disperse to different parts of the state space, a distinction that matters when useful coverage must be obtained within only a few executions (Mutti et al., 2022).

A simple example in Figure 1 makes the ambiguity precise. An agent starts at a hub and has time for two room visits, returning to the hub after each. One policy follows a fixed schedule: left room, then right. Another independently chooses a room at each visit. Weighting the state at time t proportionally to γ^t, choosing the left room with probability $q = 1/(1+\gamma^2)$ makes the two policies induce the same discounted state-visitation distribution. At $\gamma = 0.9$, both assign probabilities $(0.602, 0.220, 0.178)$

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: The fixed schedule visits both rooms on every execution, whereas independent resampling does so with probability $2q(1-q) \approx 49.45\%$. With normalized weights $w_t = \gamma^t / \sum_{j=0}^{4} \gamma^j$, $\gamma = 0.9$, and $q = 1/(1+\gamma^2)$, the two policies induce the same pooled state distribution and hence the same pooled state entropy, despite their different behavior across executions. Snapshot entropy distinguishes this time-aligned uncertainty.



to the hub, left room, and right room, and hence have entropy 1.364 bits. Yet the fixed schedule visits both rooms on every run, whereas independent resampling does so only about half the time.

To expose what this aggregation hides, let us keep track of elapsed time. Let S_t be the agent's state at time t, and let $d_t(s) = \mathbb{P}(S_t = s)$ be its *state law*, the distribution of states across independent runs at that time. Given normalized visitation weights w_t, define the *pooled state distribution* as $\bar{d}(s) = \sum_t w_t d_t(s)$ and its entropy as the *pooled state entropy*. Now let T be drawn according to the same weights, independently of the run, and let $X = S_T$. Then X has law $\bar{d}$, and Shannon's chain rule gives $H(X) = H(X \mid T) + I(T; X)$. We call $H(X \mid T)$ the *snapshot entropy*: it measures the uncertainty across executions that remains once elapsed time is known. Thus pooled entropy has two sources: time-aligned variation across executions, measured by $H(X \mid T)$, and systematic change over time, measured by $I(T; X)$. In Figure 1, the scheduled policy has zero snapshot entropy, while independent resampling has snapshot entropy 0.395 bits.

This distinction goes beyond the example. Maximum-entropy methods can produce dispersed state laws, and mixture-based planners can add run-to-run variation when different components occupy different regions at the same elapsed time (Hazan et al., 2019). On Ant, pooled spatial occupancy can appear broad even when synchronized executions remain substantially dispersed. At the other extreme, minimizing snapshot entropy alone is insufficient: an agent can be perfectly predictable simply by staying in one place.

Contributions. Our contributions range from defining the objective, to learning it without a model, to understanding its theoretical foundations, and finally to testing when it helps in practice.

C1. Objective and interpretation. We introduce *exploration with certainty*, which seeks broad pooled coverage while bounding time-aligned uncertainty:

$$\max_{\pi} H(X) \qquad \text{subject to} \qquad H(X \mid T) \leq \varepsilon. \tag{1}$$

The objective neither prescribes a route nor supplies a task reward. The snapshot budget controls disagreement between independent executions at the same sampled time and guarantees that, if m_τ is the visitation distribution of run τ, then $\mathbb{E}\tau[H(m_\tau)] \geq H(X) - H(X \mid T)$. Thus pooled entropy C and snapshot entropy at most ε imply expected within-run entropy at least $C - \varepsilon$.

C2. Model-free optimization. We derive an exact policy-gradient identity for the Lagrangian $H(X) - \lambda H(X \mid T)$. States rare in the pooled distribution receive a coverage bonus, while states unexpected across executions at the same time receive an uncertainty penalty. Estimating pooled and time-indexed state laws from synchronized rollouts yields *Certainty Policy Gradient* (Certainty PG), a model-free learner.

C3. Foundations of exploration with certainty. We characterize structure and limits of the objective. Snapshot entropy is concave in the occupancy measure, making the constraint reverse-convex. Deterministic policies suffice to minimize it, yet finding a high-coverage route under a small budget is NP-hard even in deterministic MDPs. For $\lambda > 1$, every deterministic route is locally optimal with respect to induced state laws, including incomplete routes. Under stochastic transitions, we derive a dynamic-programming lower bound on the smallest achievable snapshot entropy.

C4. Empirical evaluations. We test whether the distinction appears on controlled exploration tasks and a standard continuous-control benchmark, whether broad repeatable routes can be learned when they exist, and where enforcing certainty fails. On Ant, synchronized executions expose same-time dispersion hidden by pooled occupancy. On clean Rooms-8, Certainty PG retains 98.0% of Pooled PG's pooled entropy while increasing single-run all-state coverage from 31.5% to 99.8%. Comparisons with trajectory entropy, transition noise, and navigation stress tests expose both the benefits and limits of enforcing repeatability. Additional structural results, long-horizon guarantees, proofs, ablations, estimator checks, and extended experiments appear in the appendix.

2 Related Work

Entropy-based exploration. Maximum state-entropy exploration seeks broad visitation by maximizing the entropy of an aggregate state distribution (Hazan et al., 2019; Seo et al., 2021; Mutti et al., 2021). Our starting point is that this aggregate distribution pools variation across time and across executions. Snapshot entropy instead isolates the uncertainty that remains across runs once elapsed time is known, allowing pooled coverage and repeatability to be controlled separately. Count-based exploration provides another notion of coverage through visitation novelty (Bellemare et al., 2016), but does not directly distinguish temporal structure from run-to-run dispersion.

Trajectory-level coverage and predictability. Trajectory-level entropy objectives directly encourage broad visitation within individual runs (Mutti et al., 2022). Our objective is complementary: high within-run coverage does not require independent executions to follow the same time-structured behavior, whereas exploration with certainty explicitly bounds such time-aligned uncertainty. At the opposite extreme, surprise-minimization methods such as SMiRL encourage predictable state distributions without a coverage objective (Berseth et al., 2021). Exploration with certainty combines these two desiderata by maximizing pooled coverage subject to a bound on snapshot entropy.

3 Problem Definition

For a finite set $\mathcal{Y}$, let $\Delta(\mathcal{Y})$ denote distributions on $\mathcal{Y}$, δ_y the point mass at y, and $\mathrm{supp}(d)$ the support of d. We write $\mathbb{P}$ and $\mathbb{E}$ for probability and expectation, and $\mathbf{1}\{E\}$ for the indicator of E. All logarithms are base 2 unless stated otherwise, so entropies are measured in bits. For $d \in \Delta(\mathcal{Y})$,

$$H(d) = -\sum_{y \in \mathcal{Y}} d(y) \log_2 d(y),$$

and for a random variable $Y \sim d$ we write $H(Y)=H(d)$. We use the usual conditional entropy $H(Y|Z)$ and mutual information $I(Y;Z) = H(Y)-H(Y \mid Z)$; see Section B.1 for standard facts.

A finite Markov decision process (MDP) is $\mathcal{M} = (\mathcal{S}, (\mathcal{A}(s))_{s\in\mathcal{S}}, P, \rho)$, with finite state space $\mathcal{S}$, $n = |\mathcal{S}|$, enabled actions $\mathcal{A}(s)$, transition law $P(\cdot \mid s, a)$, and initial distribution ρ. We fix a horizon of $K \geq 2$ states, indexed by $t = 0, \ldots, K-1$. A run is $\tau = s_0 a_0 s_1 \cdots a_{K-2} s_{K-1}$. A time-dependent Markov policy is a sequence $\pi = (\pi_t)_{t=0}^{K-2}$ with $\pi_t(\cdot \mid s) \in \Delta(\mathcal{A}(s))$; we write Π_K for this class. A policy is deterministic if every $\pi_t(\cdot \mid s)$ is a point mass, and stationary if it does not depend on t. The MDP is deterministic if ρ and every $P(\cdot \mid s, a)$ are point masses.

Under π, let S_t and A_t denote the state and action at time t, with

$$S_0 \sim \rho, \qquad A_t \sim \pi_t(\cdot \mid S_t), \qquad S_{t+1} \sim P(\cdot \mid S_t, A_t).$$

We write

$$d_t^\pi(s) = \mathbb{P}_\pi(S_t = s), \qquad \mu_t^\pi(s,a) = \mathbb{P}_\pi(S_t = s, A_t = a)$$

for the *state law* and *occupancy measure* at time t. General history-dependent strategies and the corresponding path measure are recalled in Section A.

3.1 EXPLORATION WITH CERTAINTY

We now define the two quantities from the introduction for a fixed horizon of K time steps. As in discounted reinforcement learning, earlier times receive more weight: for a discount $\gamma \in (0, 1)$ let

$$w_t = \frac{(1-\gamma)\gamma^t}{1-\gamma^K}, \qquad 0 \le t < K. \tag{2}$$

The weights are positive and sum to one, so $w = (w_t)_{t<K}$ is a distribution over times: the geometric distribution with parameter γ conditioned on being below K. We write $\mathbb{P}(T = t) = w_t$ and define

$$\text{pooled state distribution:} \qquad \bar{d}^{\pi}_{\gamma,K} := \sum_{t=0}^{K-1} w_t d^{\pi}_t, \tag{3}$$

$$\text{pooled coverage:} \qquad C_{\gamma,K}(\pi) := H\big(\bar{d}^{\pi}_{\gamma,K}\big), \tag{4}$$

$$\text{snapshot entropy:} \qquad U_{\gamma,K}(\pi) := \sum_{t=0}^{K-1} w_t H(d^{\pi}_t). \tag{5}$$

The coverage $C_{\gamma,K}(\pi)$ is the entropy of the state distribution once time is forgotten, while the snapshot entropy $U_{\gamma,K}(\pi)$ is the uncertainty that remains once time is known. We omit (γ, K) when these parameters are fixed. The two quantities are related by a simple decomposition: pooled coverage consists of snapshot entropy plus the information that the state carries about time.

Proposition 1. *Let $T \sim w$ be drawn independently of a run of π, and set $X = S_T$. Then*

$$C_{\gamma,K}(\pi) = U_{\gamma,K}(\pi) + I_{\pi}(T; X), \qquad 0 \le I_{\pi}(T; X) \le \min\{\log_2 n,\, H(T)\}. \tag{6}$$

The proof is Shannon's chain rule (Cover & Thomas, 2006, Ch. 2): X has law $\bar{d}^{\pi}_{\gamma,K}$, while snapshot entropy is the weighted sum in (5) (Section B.2). Mutual information $I(T; X)$ is high for an orderly tour through different regions and low when the same broad distribution appears at every time.

Shannon entropy has a direct operational meaning: two independent draws from a distribution of entropy H bits coincide with probability at least 2^{-H}. Applied to the state laws, this quantifies how often two independent runs of the same policy occupy the same state at the same time.

Proposition 2 (Meaning of the uncertainty budget)**.** *Run a policy π twice independently, with states S_t and S'_t, and draw $T \sim w$ independently. Then*

$$\mathbb{P}(S_T \neq S'_T) = \sum_t w_t \Big(1 - \sum_s d_t(s)^2\Big) \le 1 - 2^{-\sum_t w_t H(d_t)}. \tag{7}$$

The same bound holds for the error probability of predicting a most likely state at time T.

Since $\sum_t w_t H(d_t) = U_{\gamma,K}(\pi)$, a budget $U_{\gamma,K}(\pi) \le \varepsilon$ bounds disagreement by $1 - 2^{-\varepsilon}$. The proof uses Jensen's inequality (Section B.3). The bound is averaged over $T \sim w$, not uniform over time.

Agreement between runs is not our ultimate goal: we want each run to cover the state space. For a run τ, let m_τ be its weighted visitation distribution and $J(\pi) := \mathbb{E}_\tau[H(m_\tau)]$ its expected within-run entropy of π. The next theorem shows that small snapshot entropy keeps J close to pooled coverage.

Theorem 3 (Certainty certifies single-run coverage)**.** *For a run $\tau = (S_0, \ldots, S_{K-1})$ of π, let $m_\tau(s) = \sum_t w_t \mathbf{1}\{S_t = s\}$. Let $|\{S_0, \ldots, S_{K-1}\}|$ denote number of distinct states in τ, then*

$$\begin{aligned} C_{\gamma,K}(\pi) - U_{\gamma,K}(\pi) \le \mathbb{E}_\tau[H(m_\tau)] \le C_{\gamma,K}(\pi), \\ \mathbb{E}_\tau\big|\{S_0, \ldots, S_{K-1}\}\big| \ge 2^{\,C_{\gamma,K}(\pi) - U_{\gamma,K}(\pi)}. \end{aligned} \tag{8}$$

As in Proposition 1, let $T \sim w$ be independent of the run and $X = S_T$. Conditional on τ, X has law m_τ, so $\mathbb{E}_\tau H(m_\tau) = I_\pi(T; X \mid \tau) \ge I_\pi(T; X)$. The upper bound follows from concavity of entropy, and the distinct-state bound from $|\mathrm{supp}(m_\tau)| \ge 2^{H(m_\tau)}$ and Jensen's inequality (Section B.4). These are expectation guarantees, not complete-coverage guarantees for every run. Here, J is the geometrically weighted counterpart of the single-trajectory criterion of Mutti et al. (2022).

These guarantees motivate the central optimization problem of this paper:

Exploration with certainty. For an MDP $\mathcal{M}$, horizon K, discount γ, and budget $\varepsilon \geq 0$,

$$\max_{\pi \in \Pi_K} C_{\gamma,K}(\pi) \qquad \text{s.t.} \qquad U_{\gamma,K}(\pi) \leq \varepsilon. \tag{9}$$

Why time-dependent Markov policies. Coverage and snapshot entropy depend on a strategy only through its state laws. For every history-dependent strategy σ, there is a policy $\pi \in \Pi_K$ with the same occupancy measures, and hence the same state laws: take $\pi_t(a \mid s) = \mu_t^\sigma(s,a)/d_t^\sigma(s)$ whenever $d_t^\sigma(s) > 0$ (Lemma 11). Thus optimizing over Π_K in (9) loses nothing.

The time index, however, is essential. In the two-room example, the scheduled policy goes left on the first visit to the hub and right on the second. A stationary policy must behave identically on both visits, so it cannot realize this schedule.

4 Certainty PG: Learning from Time-Aligned Rollouts

Parameterized policies. Let π_θ, $\theta \in \mathbb{R}^p$, be a differentiable policy, and abbreviate $C(\theta) = C_{\gamma,K}(\pi_\theta)$, $U(\theta) = U_{\gamma,K}(\pi_\theta)$, and $d_t^\theta = d_t^{\pi_\theta}$.

Lagrangian and per-state score. We optimize (9) through

$$\mathcal{L}(\theta, \lambda) = C(\theta) - \lambda\big(U(\theta) - \varepsilon\big), \qquad \lambda \geq 0, \tag{10}$$

using ascent in θ and projected descent in λ. The corresponding per-state score is

$$r_t^\lambda(s) = \underbrace{-\log_2 \bar{d}_{\gamma,K}(s)}_{\text{coverage bonus}} - \lambda \underbrace{\big[-\log_2 d_t(s)\big]}_{\text{same-time surprise}}. \tag{11}$$

Thus globally rare states receive a coverage bonus, while states rare among runs at the same time incur an uncertainty penalty. At $\lambda{=}0$ this recovers pooled-entropy maximization; at $\lambda{=}1$ the objective is $C{-}U{+}\varepsilon$, which by Theorem 3 lower-bounds expected within-run entropy up to the additive constant.

Snapshot policy gradient theorem. Although entropy is nonlinear in the state laws, its gradient still admits a likelihood-ratio form.

Theorem 4 (Policy Gradient)**.** *Assume P and ρ are independent of θ, and π_θ is differentiable with every enabled action assigned positive probability. For a run $\tau = (S_0, A_0, \ldots, S_{K-1})$ under π_θ,*

$$\nabla_\theta U(\theta) = \mathbb{E}_\tau\left[\sum_{k=0}^{K-2} \nabla_\theta \ln \pi_\theta(A_k \mid S_k, k) \sum_{t=k+1}^{K-1} w_t\big[-\log_2 d_t^\theta(S_t)\big]\right]. \tag{12}$$

The same identity holds for C with $\bar{d}_{\gamma,K}$ in place of d_t; hence $\nabla_\theta \mathcal{L} = \nabla_\theta C - \lambda \nabla_\theta U$.

The constant term in the derivative of $-p\log_2 p$ cancels because probabilities sum to one, so the densities inside the logarithms may be treated as fixed when differentiating (Section B.5).

Time-aligned estimation. Estimating d_t requires several runs observed at the same elapsed time. We therefore collect B independent runs under frozen θ, align them by time, and estimate each d_t from column t of the batch. Let $N_t(s)$ be the number of runs in state s at time t and $m_i(s) = \sum_t w_t \mathbf{1}\{S_t^i = s\}$. For run i we use leave-one-out estimates

$$\widehat{d}_t^{\,-i}(s) = \frac{N_t(s) - \mathbf{1}\{S_t^i = s\} + \alpha}{B - 1 + \alpha n}, \qquad \widehat{\bar{d}}_{\gamma,K}^{\,-i}(s) = \frac{\sum_t w_t N_t(s) - m_i(s) + \alpha}{B - 1 + \alpha n}, \tag{13}$$

where $\alpha{>}0$ keeps logarithms finite. Leaving run i out prevents its own visit from inflating the density used to score it. The estimates are consistent as $B{\to}\infty$ but biased for finite B (Paninski, 2003).

Algorithm 1 combines these ingredients: each iteration collects time-aligned rollouts, forms leave-one-out scores and likelihood-ratio returns, updates θ, and adjusts λ according to the estimated snapshot entropy. A time-dependent baseline reduces variance without changing the expected gradient (Section B.5).

Algorithm 1 Certainty policy gradient (Certainty PG)

Require: Reset access to ρ and samples from unknown P; policy π_θ; $K, B \geq 2$; γ; budget ε; pseudocount α; rates η_λ, β; cap $\lambda_{\max}$

Compute w_t by (2); $\lambda \leftarrow 0$; $b_k \leftarrow 0$
repeat
  Run B independent runs of K states under frozen θ; compute $N_t(s)$ and $m_i(s)$
  for each run i **do**
    Score states by (11) using (13); $\widehat{G}^\lambda_{i,k} \leftarrow \sum_{t>k} w_t \widehat{r}^\lambda_{i,t}$
  end for
  $\widehat{g} \leftarrow B^{-1} \sum_i \sum_{k=0}^{K-2} \nabla_\theta \ln \pi_\theta(A^i_k \mid S^i_k, k)(\widehat{G}^\lambda_{i,k} - b_k)$; $\theta \leftarrow \mathsf{Ascent}(\theta, \widehat{g})$
  $b_k \leftarrow (1-\beta) b_k + \beta B^{-1} \sum_i \widehat{G}^\lambda_{i,k}$
  $\lambda \leftarrow \Pi_{[0,\lambda_{\max}]}[\lambda + \eta_\lambda (\sum_t w_t H(N_t/B) - \varepsilon)]$
until the interaction limit is reached
return π_θ

5 Structure and Limits of Certainty

Before turning to experiments, we ask what the certainty constraint does to the optimization problem: its geometry, computational difficulty, local landscape, and feasibility. Proofs are in Section B.

Occupancy geometry. Policy parameters are not a convenient space for studying convexity, since state laws depend nonlinearly on the decision rules. Instead, let $\mu^\pi_t(s,a) = \mathbb{P}_\pi(S_t = s, A_t = a)$ and let $\mathcal{Q}$ be the set of occupancy sequences. Each state law is linear in μ, so coverage and snapshot entropy define functions $C(\mu)$ and $U(\mu)$ on $\mathcal{Q}$. Mixtures of policies are central here. The *mixture* of policies $\pi_1, \ldots, \pi_k$ with weights α draws i with probability α_i at the start of each run and then follows π_i; $C(\mathrm{mix})$ and $U(\mathrm{mix})$ denote its coverage and snapshot entropy.

Proposition 5 (Occupancy geometry)**.** *(a) $\mathcal{Q}$ is the convex hull of deterministic time-dependent occupancies, and C and U are concave on $\mathcal{Q}$.*

(b) $\min_{\pi \in \Pi_K} U_{\gamma,K}(\pi)$ is attained by a deterministic time-dependent policy.

(c) If a mixture selects π_i with probability α_i, then

$$\sum_i \alpha_i U(\pi_i) \leq U(\mathrm{mix}) \leq \sum_i \alpha_i U(\pi_i) + H(\alpha), \qquad C(\mathrm{mix}) \geq \sum_i \alpha_i C(\pi_i).$$

(d) The feasible set $\{\mu \in \mathcal{Q} : U(\mu) \leq \varepsilon\}$ need not be convex.

The occupancy-polytope and concavity facts for C are standard (Puterman, 1994; Hazan et al., 2019); here we extend this geometric picture to snapshot entropy. The certainty constraint is reverse-convex: mixing individually predictable policies can make their mixture uncertain(Section B.6).

Intractability of certain-route selection. We call $(s_0, \ldots, s_{K-1})$ a *route* if every run visits state s_t at time t. Equivalently, a policy has a route exactly when $U{=}0$. In a deterministic MDP, every deterministic policy has a route, so one might hope to maximize coverage simply by searching over routes. Unfortunately, this problem is already hard.

Theorem 6. *Consider deterministic MDPs with n states, fixed initial state, horizon $K{=}n$, discount $\gamma{=}1{-}1/n$, and $0 \leq \varepsilon \leq 1/(1000\, e\, n^3)$, where e is Euler's number. Unless P = NP, no polynomial-time algorithm returns a policy with $U_{\gamma,K} {\leq} \varepsilon$ whose coverage is within $1/(3n)$ bits of the optimum.*

The reduction is from directed Hamiltonian path. When $\varepsilon{=}0$, maximum coverage requires a route visiting all states without repetition; the argument extends to the stated positive budget (Section B.8).

A local threshold at $\lambda = 1$. Hardness concerns global planning with a known model. Certainty PG instead makes local policy updates, so we study $\mathcal{L}_\lambda = C - \lambda U$ near a deterministic route.

Proposition 7. *Let f be a deterministic policy with route (s_t), and let $\bar{\kappa}(\pi) = \max_t(1 - d^\pi_t(s_t))$ be the largest probability, over times, that π is off this route.*

(a) If $\lambda > 1$, then f is a strict local maximum with respect to its induced state laws: $\mathcal{L}_\lambda(\pi) < \mathcal{L}_\lambda(f)$ whenever $0 < \bar{\kappa}(\pi) < r_\lambda$, where $r_\lambda = (w^2_{K-1}/(en))^{1/(\lambda-1)}$.

(b) If changing one action creates a route differing on total weight W and visiting previously unseen states on weight M, then $\lambda W < M$ implies that mixing in this change with sufficiently small probability increases $\mathcal{L}_\lambda$.

The threshold comes from two competing terms of order $\kappa \log_2(1/\kappa)$, where κ is the small probability of leaving the route: near a route, both the coverage gain and the uncertainty penalty have this order, but the penalty is scaled by λ. Thus for $\lambda > 1$, even incomplete routes can be locally optimal; below one, detours into new states can improve the objective (Section B.9).

Feasibility under transition noise. Some snapshot uncertainty may be unavoidable. Let $h(s) = \min_a H(P(\cdot \mid s, a))$ be the least next-state entropy achievable from s.

Proposition 8 (A computable feasibility floor)**.** *For every $\pi \in \Pi_K$,*

$$U_{\gamma,K}(\pi) \geq w_0 H(\rho) + \min_{\pi'} \sum_{t=1}^{K-1} w_t \, \mathbb{E}_{\pi'}[h(S_{t-1})]. \tag{14}$$

The minimum on the right is the optimal value of an ordinary finite-horizon expected-cost MDP problem and can be computed by backward induction.

The bound follows because conditioning on the previous state and action can only reduce uncertainty, while the next-state entropy is at least $h(s)$. The bound need not equal $\min_\pi U_{\gamma,K}(\pi)$, but it provides a computable certificate that sufficiently small budgets are infeasible (Section B.10).

6 Experiments

We investigate three questions: does snapshot entropy expose behavior hidden by pooled coverage, can Certainty PG learn broad and repeatable exploration when such behavior is achievable, and how can enforcing certainty fail? Tabular environments allow exact evaluation of the state laws, while Ant tests whether the same distinction remains visible in continuous control.

Setup. The learner observes states, chooses actions, and can reset to a fresh state from ρ, but does not know P. Tabular experiments use time-dependent softmax policies with matched interaction budgets and optimizers across methods. We compare Certainty PG with Pooled PG, which maximizes C; Trajectory-entropy PG, which maximizes expected within-run entropy J; Count PG; and uniform actions. Additional objectives and ablations, including Snapshot-min PG, are reported in the appendix. Unless stated otherwise, we use $B = 256$, $K = 33$, $\gamma = 0.97$, $\varepsilon = 0.1$ bits, and ten evaluation seeds; full settings are in Section H. Ant uses $K = 257$, $\gamma = 0.995$, a one-bit target, and five seeds; full continuous-control settings are in Section K.

Our primary structured task is Rooms-8, an eight-room extension of Figure 1 in which 16 room visits must be scheduled from a common hub. We also use a 16-state directed cycle, noisy versions of both environments, and navigation stress tests. We report exact pooled coverage C and snapshot entropy U, together with the minimum probability of visiting any noninitial state, $D_{\min}(\pi) = \min_{g \neq s_0} \Pr_\pi(\exists t < K : S_t = g)$, and the probability D_{all} that a single run visits every state.

6.1 Coverage, repeatability, and their separation

Similar pooled coverage can conceal very different explorations. On clean Rooms-8, Pooled PG attains $C = 2.450$ and $U = 1.418$ bits, whereas Certainty PG attains $C = 2.402$ and $U = 0.010$. Thus Certainty PG retains 98.0% of Pooled PG's pooled entropy while increasing the probability of visiting every state in one run from 31.54% to 99.75% (Table 1). The two objectives therefore assign nearly the same value to pooled coverage while producing qualitatively different exploration policies.

The directed cycle shows the same phenomenon under different dynamics: Pooled PG approaches the maximal four pooled bits but retains $U = 2.006$ bits, whereas Certainty PG reaches $C = 3.926$ with $U = 0.0047$ and worst-state discovery above 99.99%. Figure 2 makes the difference explicit. Certainty PG learns a concentrated state law that moves through the state space with time; Pooled PG instead obtains high marginal coverage partly by remaining dispersed across executions at each time.

Single-run coverage and repeatability are distinct objectives. Trajectory-entropy PG directly rewards within-run coverage and reaches 99.33% all-state coverage on Rooms-8, essentially matching

Table 1: Held-out final policies (ten seeds per cell). Entries are mean [95% seed-bootstrap interval]; entropies are in bits. $D_{\min}$ is the minimum probability of visiting any noninitial state, and D_{all} is the probability that one run visits every state; both are percentages. No goal rewards are used for training, and noisy rows may violate the constraint.

Method	C	U	$I(T;X)$	$D_{\min}$	D_{all}
Rooms-8 (clean)					
Random	2.450 [2.450,2.450]	1.451 [1.451,1.451]	0.999 [0.999,0.999]	88.2 [88.2,88.2]	31.31 [30.72,31.84]
Pooled PG	2.450 [2.450,2.450]	1.418 [1.415,1.420]	1.032 [1.030,1.035]	87.1 [86.7,87.4]	31.54 [31.13,31.85]
Count PG	2.450 [2.449,2.450]	1.420 [1.419,1.422]	1.029 [1.028,1.031]	87.4 [87.1,87.7]	31.12 [30.53,31.71]
Certainty PG	2.402 [2.384,2.416]	0.010 [0.010,0.010]	2.391 [2.374,2.406]	99.8 [99.8,99.9]	99.75 [99.68,99.82]
Rooms-8 (slip 0.15)					
Random	2.450 [2.450,2.450]	1.451 [1.451,1.451]	0.999 [0.999,0.999]	88.2 [88.2,88.2]	31.19 [30.72,31.66]
Pooled PG	2.450 [2.450,2.450]	1.427 [1.426,1.428]	1.023 [1.021,1.024]	87.2 [87.0,87.4]	31.37 [30.97,31.77]
Count PG	2.449 [2.449,2.450]	1.429 [1.428,1.430]	1.021 [1.020,1.022]	87.1 [86.8,87.4]	30.95 [30.59,31.25]
Certainty PG	2.405 [2.392,2.416]	0.452 [0.452,0.452]	1.953 [1.940,1.964]	83.7 [70.9,90.1]	70.86 [58.71,78.95]
Directed cycle-16 (clean)					
Random	3.895 [3.895,3.895]	2.585 [2.585,2.585]	1.309 [1.309,1.309]	70.2 [70.2,70.2]	70.25 [69.97,70.47]
Pooled PG	4.000 [4.000,4.000]	2.006 [1.971,2.040]	1.993 [1.959,2.027]	92.3 [89.8,94.4]	92.63 [90.08,94.75]
Count PG	3.960 [3.955,3.965]	2.434 [2.401,2.469]	1.526 [1.492,1.559]	76.0 [72.9,79.4]	76.06 [72.75,79.64]
Certainty PG	3.926 [3.913,3.940]	0.005 [0.005,0.005]	3.921 [3.908,3.935]	100.0 [100.0,100.0]	100.00 [100.00,100.00]
Directed cycle-16 (slip 0.10)					
Random	3.895 [3.895,3.895]	2.585 [2.585,2.585]	1.309 [1.309,1.309]	70.2 [70.2,70.2]	70.41 [69.99,70.73]
Pooled PG	4.000 [4.000,4.000]	1.983 [1.930,2.033]	2.017 [1.967,2.068]	94.8 [92.8,96.4]	94.98 [93.15,96.51]
Count PG	3.952 [3.948,3.956]	2.441 [2.409,2.477]	1.511 [1.476,1.542]	78.8 [76.9,80.5]	79.15 [77.21,80.90]
Certainty PG	3.954 [3.943,3.963]	0.327 [0.321,0.332]	3.627 [3.613,3.640]	99.9 [99.8,99.9]	99.90 [99.84,99.96]

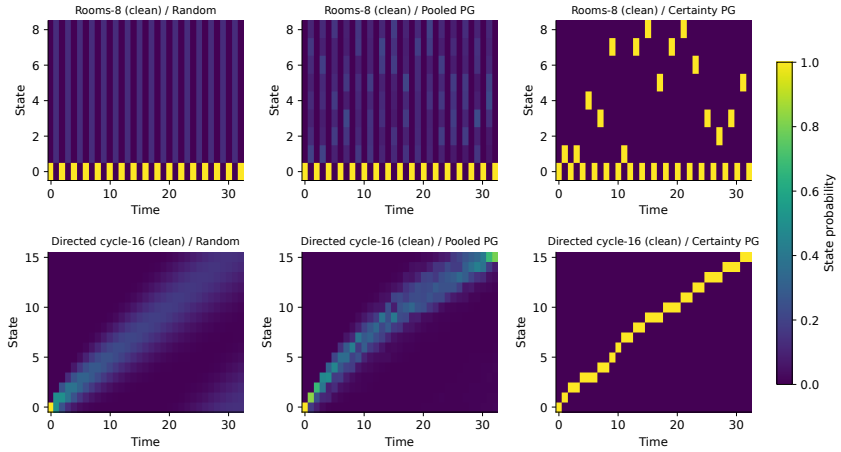
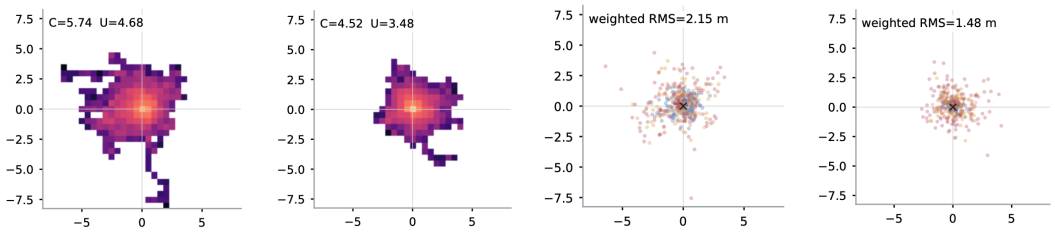


Figure 2: **Learned visitation schedules.** Exact state laws of representative trained policies, with time along the horizontal axis and state along the vertical axis. Certainty PG moves a concentrated state law through the state space, whereas Pooled PG attains similar pooled coverage while remaining dispersed across runs at the same time.

Certainty PG, with $C{=}2.445$ bits. Its snapshot entropy, however, is $U{=}0.114$ bits, compared with 0.010 for Certainty PG. Its expected within-run entropy is also slightly higher, 2.436 versus 2.401 bits. Thus strong single-run coverage does not by itself imply that independent executions follow the same time-structured exploration pattern. Certainty PG's effect is to align those runs in time.

The same distinction appears on Ant, but with a coverage cost. We next test whether time-aligned dispersion remains visible beyond structured tabular environments. On Ant, a continuous-control benchmark used in prior maximum state-entropy exploration work, we define spatial occupancy using torso position. Pooled PG spreads this occupancy broadly, but independent executions remain far apart at the same elapsed time. Certainty PG reduces the weighted RMS same-time separation from 2.51 m to 1.13 m and snapshot entropy from 4.88 to 3.11 bits. The contraction is therefore visible both in the entropy objective and in an unbinned geometric measure.

Unlike Rooms-8, however, Ant does not exhibit a broad reproducible tour: the reduction in dispersion is accompanied by a drop in pooled entropy from 6.19 to 3.88 bits and in distinct spatial bins per run



(a) Pooled: occupancy (b) Certainty: occupancy (c) Pooled: aligned (d) Certainty: aligned

Figure 3: **Pooled coverage v. time-aligned dispersion on Ant.** (a,b) Pooled torso-position occupancy; (c,d) 128 synchronized executions at $t = 64, 128, 256$. Certainty PG yields tighter aligned executions but a smaller pooled spatial footprint. Representative seed; aggregate results are reported in the text.

from 15.4 to 8.6. Moreover, all Certainty PG seeds remain above the one-bit target. Ant therefore supports the distinction between pooled coverage and time-aligned uncertainty, but also shows that reducing snapshot entropy need not preserve broad exploration or achieve the desired certainty level.

6.2 When certainty is infeasible or incomplete

Transition noise creates unavoidable uncertainty. On noisy Rooms-8 with replacement probability $p = 0.15$, the lower bound from Proposition 8 is $U \geq 0.4494$ bits, so the target $\varepsilon = 0.1$ is infeasible for every policy. Certainty PG reaches $U = 0.452$, only 0.003 bits above this floor; at $p = 0.30$ it reaches 0.760 bits against a 0.758-bit floor. Across the noise sweep, Figure 4 shows that achieved uncertainty closely tracks the computable lower bound. Thus these constraint violations are largely explained by irreducible transition noise rather than failure of the optimizer. Budget sweeps and finite-batch effects appear in Section I.6.

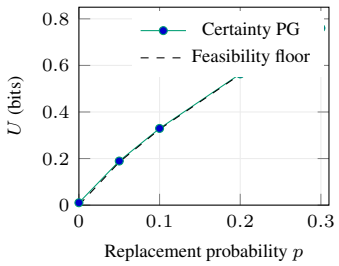


Figure 4: **Transition noise imposes irreducible uncertainty.** Certainty PG closely tracks the feasibility floor on noisy Rooms-8; points show means with 95% bootstrap intervals.

Feasible certainty may lock onto an incomplete route. The opposite failure occurs in deterministic navigation tasks, where low-uncertainty broad routes do exist. With 16 rooms, Certainty PG reaches $U = 0.0216$ bits but visits every state in only 30.0% of runs. On the bidirectional ring and open grid, every learned policy satisfies the constraint, yet no evaluation run visits every state. In the open grid, the learned policy instead repeatedly traverses only a subset of the states, even though a deterministic route covering the entire grid with $U = 0$ fits within the horizon.

This behavior is consistent with the local landscape in Proposition 7: the learned multiplier crosses $\lambda = 1$ early, after which small deviations from an already concentrated route can lower the Lagrangian. The experiment does not establish that the learned finite-logit policies are local maxima, but it illustrates the mechanism: certainty can be achieved before sufficient coverage has been discovered. Further stress tests and mitigation attempts appear in the appendix.

7 Conclusion

Snapshot entropy separates broad pooled visitation from repeatable, time-structured exploration across executions. This distinction leads to a natural certainty-constrained exploration objective, an exact model-free policy-gradient method, and structural results characterizing when certainty is achievable and when it can be computationally or locally difficult.

Empirically, Certainty PG can recover broad reproducible routes when they exist, but the experiments also expose its limits: transition noise can impose irreducible uncertainty, while aggressive concentration can lock the learner onto incomplete routes. These results suggest that coverage and repeatability should be treated as distinct design objectives in reward-free exploration.

AI USE STATEMENT

Generative AI tools were used during the preparation of this work to assist with brainstorming and refining the technical presentation, including refining proof exposition, developing and debugging code, assisting with literature search and discovery, and editing the manuscript to meet the page limit and improve clarity and concision. All mathematical statements, proofs, experimental methodology, implementations, results, and citations were independently reviewed and verified by the authors. The authors take full responsibility for the content and correctness of the paper.

REPRODUCIBILITY STATEMENT

We have taken several steps to support reproducibility. Complete proofs of the theoretical results, together with additional derivations and technical details, are provided in the appendix. The appendix also specifies the environments, algorithms, hyperparameters, training budgets, seed splits, evaluation procedures, and additional experimental results used in the empirical study. The main experiments use fixed experimental protocols, held-out evaluation seeds, and final iterates without checkpoint selection; reported uncertainty intervals are computed across evaluation seeds. Pseudocode and implementation details for Certainty PG and the comparison methods are provided to facilitate independent reproduction of the results. The source code used to run the experiments and generate the reported results will be released upon acceptance.

REFERENCES

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/afda332245e2af431fb7b672a68b659d-Abstract.html.

Glen Berseth, Daniel Geng, Coline Devin, Nicholas Rhinehart, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. SMiRL: Surprise minimizing reinforcement learning in unstable environments. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=cPZOyoDloxl.

Jean Cardinal, Samuel Fiorini, and Gwenaël Joret. Tight results on minimum entropy set cover. *Algorithmica*, 51(1):49–60, 2008.

Jean Cardinal, Samuel Fiorini, and Gwenaël Joret. Minimum entropy combinatorial optimization problems. *Theory of Computing Systems*, 51(1):4–21, 2012. doi: 10.1007/s00224-011-9371-2. URL https://arxiv.org/abs/1008.2928.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2 edition, 2006. doi: 10.1002/047174882X.

Elad Hazan, Sham M. Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2681–2691. PMLR, 2019. URL https://proceedings.mlr.press/v97/hazan19a.html.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4870–4879. PMLR, 2020. URL https://proceedings.mlr.press/v119/jin20d.html.

Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher (eds.), *Complexity of Computer Computations*, pp. 85–103. Plenum Press, 1972. doi: 10.1007/978-1-4684-2001-2_9.

Colin McDiarmid. On the method of bounded differences. In Johannes Siemons (ed.), *Surveys in Combinatorics, 1989*, volume 141 of *London Mathematical Society Lecture Note Series*, pp. 148–188. Cambridge University Press, 1989. doi: 10.1017/CBO9781107359949.008.

Mirco Mutti, Lorenzo Pratissoli, and Marcello Restelli. Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9028–9036, 2021.

Mirco Mutti, Riccardo De Santi, and Marcello Restelli. The importance of non-Markovianity in maximum state entropy exploration. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16223–16239. PMLR, 2022. URL https://proceedings.mlr.press/v162/mutti22a.html.

Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003. doi: 10.1162/089976603321780272.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, NY, 1994. doi: 10.1002/9780470316887.

Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9443–9454. PMLR, 2021. URL https://proceedings.mlr.press/v139/seo21a.html.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Jin Shen Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024. doi: 10.48550/arXiv.2407.17032. URL https://arxiv.org/abs/2407.17032.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

GUIDE TO THE APPENDIX

The appendix has three parts. *Theory* (Sections A to C) proves every result stated in the main text. *Extensions* (Sections D to G) treat the infinite-horizon objective, which the main text does not need. *Experiments* (Sections H to L) give the settings and complete results behind Section 6. Table 2 lists where each main-text item is proved or detailed, and Figure 5 shows which appendix sections rely on which. Section A fixes the notation used by every other section.

Table 2: Where each part of the main text is proved or detailed.

Main text	Appendix
Section 3: Proposition 1, Proposition 2, Theorem 3	Proofs in Sections B.2 to B.4, using the facts in Section B.1.
Section 3: time-dependent Markov policies suffice	Lemma 11 in Section B.6.
Section 3: discounted, infinite-horizon version	Section D.
Section 4: Theorem 4	Proof in Section B.5.
Section 4: Algorithm 1	What is guaranteed in Section C; implementation in Section H.
Section 5: Proposition 5, Theorem 6, Proposition 7, Proposition 8	Proofs in Sections B.6 and B.8 to B.10; the last three use the coverage lemma of Section B.7.
Beyond Section 5: hardness of minimizing U alone	Theorem 22 in Section G.4, with dynamic programming over state distributions in Sections E and F.
Section 6: tabular results	Settings in Sections H and I; complete tables, sweeps, stress tests and follow-up checks in Section I.
Section 6: Ant	Section K (results, including Humanoid) and Section L (protocol and all checkpoints).

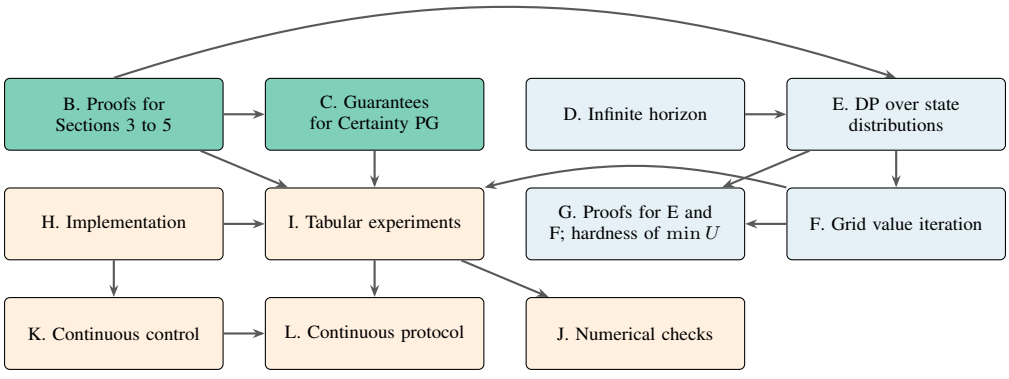


Figure 5: Dependencies between appendix sections. An arrow from X to Y means that Y uses a definition, result or setting from X. Green: theory for the main text; blue: extensions; orange: experiments. Section A (notation) is used by all sections and is omitted.

A NOTATION AND INFORMATION-THEORETIC BACKGROUND

For convenience, we collect the notation used throughout the appendix.

Notation	Meaning
$\Delta(\mathcal{Y})$	Probability distributions on a finite set $\mathcal{Y}$.
$\mathrm{supp}(d) = \{y \in \mathcal{Y} : d(y) > 0\}$	Support of $d \in \Delta(\mathcal{Y})$.
δ_y	Point mass at y.
$Y \sim d$	The random variable Y has law d.
$\mathbb{P},\ \mathbb{E}$	Probability and expectation.
$\mathbf{1}\{E\}$	Indicator of event E: 1 if E holds and 0 otherwise.
$[x]_+ = \max\{x, 0\}$	Positive part of x.
$\log_2,\ \ln$	Base-2 and natural logarithms, respectively.
$e \approx 2.718$	Euler's number, the base of $\ln$.
$h_2(x)$	Binary entropy.
$\nabla_\theta f$	Gradient of f with respect to $\theta \in \mathbb{R}^p$.

For a differentiable function $f : \mathbb{R}^p \to \mathbb{R}$,

$$\nabla_\theta f = \left(\frac{\partial f}{\partial \theta_1}, \ldots, \frac{\partial f}{\partial \theta_p} \right).$$

The binary entropy function is

$$h_2(x) = -x \log_2 x - (1-x) \log_2 (1-x), \qquad x \in [0, 1],$$

with the convention $0 \log_2 0 = 0$.

A.1 Shannon entropy and mutual information

All random variables in this paper take values in finite sets, and all logarithms are base 2, so entropies are measured in bits. For $d \in \Delta(\mathcal{Y})$,

$$H(d) = -\sum_{y \in \mathcal{Y}} d(y) \log_2 d(y), \qquad 0 \log_2 0 = 0.$$

If $Y \sim d$, we write $H(Y) = H(d)$. Entropy is zero exactly when Y is constant and is maximized by the uniform distribution.

For a pair (Y, Z), the *conditional entropy* is

$$H(Y \mid Z) = \sum_z \mathbb{P}(Z = z)\, H(Y \mid Z = z),$$

the average uncertainty about Y remaining after Z is observed. The *mutual information*

$$I(Y; Z) = H(Y) - H(Y \mid Z)$$

is the reduction in uncertainty about Y gained by observing Z. It is symmetric and nonnegative, so $H(Y \mid Z) \le H(Y)$. Conditional versions such as $H(Y \mid Z, W)$ and $I(Y; Z \mid W)$ are defined analogously. The standard properties used in our proofs are collected in Section B.1.

B Proofs of the Results in Sections 3 to 5

Throughout, $\mathcal{M} = (\mathcal{S}, (\mathcal{A}(s))_s, P, \rho)$ is a finite MDP, $K \ge 2$, $\gamma \in (0, 1)$, and $w_t = (1-\gamma)\gamma^t/(1-\gamma^K)$ for $0 \le t < K$. Every w_t is strictly positive, $\sum_t w_t = 1$, and $w_0 > w_1 > \cdots > w_{K-1}$. A policy $\pi \in \Pi_K$ is a sequence $(\pi_t)_{t=0}^{K-2}$ of maps $s \mapsto \pi_t(\cdot \mid s) \in \Delta(\mathcal{A}(s))$. A run $\tau = (S_0, A_0, S_1, \ldots, A_{K-2}, S_{K-1})$ has probability

$$\mathbb{P}_\pi(\tau) = \rho(S_0) \prod_{t=0}^{K-2} \pi_t(A_t \mid S_t)\, P(S_{t+1} \mid S_t, A_t). \tag{15}$$

In the finite-horizon arguments, the states, actions, and time index take finitely many values, so the entropy chain rules require no further conditions. The infinite-horizon arguments are stated separately; the geometric time law also has finite entropy.

B.1 Information-theoretic facts used in all proofs

We use the following standard facts for discrete random variables Y, Z, W (Cover & Thomas, 2006, Ch. 2).

(F1) *Chain rules.* $H(Y,Z) = H(Y) + H(Z \mid Y)$ and $I(Y; Z, W) = I(Y; Z) + I(Y; W \mid Z)$.

(F2) *Conditioning reduces entropy.* $H(Y \mid Z) \le H(Y)$, and $I(Y; Z \mid W) \ge 0$.

(F3) *Functions.* If $Y = g(Z)$ then $H(Y \mid Z) = 0$. If Y and Z are independent then $I(Y; Z) = 0$.

(F4) *Support bound.* $H(Y) \le \log_2 |\mathrm{supp}(Y)|$, and $H(Y) = 0$ if and only if Y is almost surely constant.

(F5) *Concavity.* $d \mapsto H(d)$ is concave on $\Delta(\mathcal{S})$.

(F6) *Grouping.* If $Z = g(Y)$, then $H(Y) = H(Z) + H(Y \mid Z) \ge H(Z)$: merging outcomes never increases entropy.

Lemma 9 (Entropy lost by merging two atoms)**.** *Let $q \in \Delta(\{1, \dots, k\})$ and $i \ne j$ with $q_i + q_j > 0$. Let q' be obtained from q by merging atoms i and j into one atom of mass $q_i + q_j$. Then*

$$H(q) - H(q') = q_i \log_2 \frac{q_i + q_j}{q_i} + q_j \log_2 \frac{q_i + q_j}{q_j} \;\ge\; \min\{q_i, q_j\},$$

with the convention $0 \log_2(\cdot/0) = 0$.

Proof. All terms other than those of i and j cancel, so $H(q) - H(q') = -q_i \log_2 q_i - q_j \log_2 q_j + (q_i + q_j) \log_2 (q_i + q_j)$, which rearranges to the displayed identity. Both terms are nonnegative. Let $b = \min\{q_i, q_j\}$ and $a = \max\{q_i, q_j\}$. If $b = 0$ the bound is trivial. Otherwise the term belonging to b is $b \log_2 \frac{a+b}{b} \ge b \log_2 2 = b$, because $a \ge b$. □

B.2 Proof of the temporal decomposition (Proposition 1, Section 3)

Let T be independent of the run with $\mathbb{P}(T = t) = w_t$, and let $X = S_T$. Then

$$\mathbb{P}(X = s) = \sum_t w_t \, \mathbb{P}_\pi(S_t = s) = \bar{d}^\pi_{\gamma,K}(s), \qquad \mathbb{P}(X = s \mid T = t) = d^\pi_t(s),$$

so $H(X) = C_{\gamma,K}(\pi)$ and $H(X \mid T) = \sum_t w_t H(d^\pi_t) = U_{\gamma,K}(\pi)$. By definition $I(T; X) = H(X) - H(X \mid T)$, which is (6). Mutual information is nonnegative (F2), and $I(T; X) \le \min\{H(X), H(T)\}$ with $H(X) \le \log_2 n$ by (F4).

The infinite-horizon identity is proved the same way with $\mathbb{P}(T = t) = (1-\gamma)\gamma^t$ for all $t \ge 0$: then X has law $\bar{d}^\pi_\gamma$, $H(X \mid T) = \mathcal{H}_\gamma(\pi)$, and $H(T) = -\log_2(1-\gamma) - \frac{\gamma}{1-\gamma} \log_2 \gamma$ by direct summation of the geometric series.

B.3 Proof of Proposition 2, Section 3

Let S_T, S'_T be the states of two independent runs at a common time $T \sim w$ drawn independently of both. Conditional on $T = t$ the two states are independent with law d_t, so $\mathbb{P}(S_T = S'_T \mid T = t) = \sum_s d_t(s)^2$ and $\mathbb{P}(S_T \ne S'_T) = \sum_t w_t \big(1 - \sum_s d_t(s)^2\big)$.

For any $d \in \Delta(\mathcal{S})$, Jensen's inequality for the concave logarithm gives

$$\log_2 \sum_s d(s)^2 = \log_2 \mathbb{E}_{S \sim d}[d(S)] \;\ge\; \mathbb{E}_{S \sim d}[\log_2 d(S)] = -H(d),$$

so $\sum_s d(s)^2 \ge 2^{-H(d)}$. Applying Jensen's inequality again to the convex map $x \mapsto 2^{-x}$ with weights w_t,

$$\sum_t w_t \sum_s d_t(s)^2 \;\ge\; \sum_t w_t 2^{-H(d_t)} \;\ge\; 2^{-\sum_t w_t H(d_t)} = 2^{-U_{\gamma,K}(\pi)},$$

which is (7). Finally, predicting a most likely state $\hat{s}_t \in \arg\max_s d_t(s)$ errs with probability $1 - \max_s d_t(s)$, and $\sum_s d_t(s)^2 \le \max_s d_t(s) \sum_s d_t(s) = \max_s d_t(s)$, so this error is at most $1 - \sum_s d_t(s)^2$ at every t and the same bound follows. □

B.4 Proof of the single-run coverage bound (Theorem 3, Section 3)

Let $\tau \sim \mathbb{P}_\pi$ be a run and let T be independent of τ with $\mathbb{P}(T = t) = w_t$. Define $X = S_T$, the state of this run at the sampled time. We establish four facts.

Step 1: marginal and time-conditional laws. As in Section B.2, X has law $\bar{d}^\pi_{\gamma,K}$ and, given $T = t$, law d^π_t. Hence $H(X) = C$, $H(X \mid T) = U$ and $I(T; X) = C - U$.

Step 2: run-conditional law. Fix a realization $\bar{\tau}$ with states $(\bar{s}_0, \ldots, \bar{s}_{K-1})$. Since T is independent of τ,

$$\mathbb{P}(X = s \mid \tau = \bar{\tau}) = \sum_t w_t \mathbf{1}\{\bar{s}_t = s\} = m_{\bar{\tau}}(s).$$

Therefore $H(X \mid \tau = \bar{\tau}) = H(m_{\bar{\tau}})$ and, averaging over $\bar{\tau}$, $H(X \mid \tau) = \mathbb{E}_\tau[H(m_\tau)]$.

Step 3: the lower bound. X is a function of (T, τ), so $H(X \mid T, \tau) = 0$ by (F3), and

$$\mathbb{E}_\tau[H(m_\tau)] = H(X \mid \tau) = H(X \mid \tau) - H(X \mid T, \tau) = I(T; X \mid \tau).$$

Expand $I(T; X, \tau)$ in two orders using (F1):

$$I(T; \tau) + I(T; X \mid \tau) \;=\; I(T; X, \tau) \;=\; I(T; X) + I(T; \tau \mid X).$$

T is independent of τ, so $I(T; \tau) = 0$ (F3). Hence

$$\mathbb{E}_\tau[H(m_\tau)] = I(T; X) + I(T; \tau \mid X) \;\geq\; I(T; X) = C - U,$$

using $I(T; \tau \mid X) \geq 0$ (F2) and Step 1.

Step 4: the upper bound and the support bound. By (F2), $\mathbb{E}_\tau[H(m_\tau)] = H(X \mid \tau) \leq H(X) = C$. Since every $w_t > 0$, the support of m_τ is exactly the set of distinct states visited by τ, so (F4) gives $|\{S_0, \ldots, S_{K-1}\}| = |\mathrm{supp}(m_\tau)| \geq 2^{H(m_\tau)}$. Taking expectations and applying Jensen's inequality to the convex map $x \mapsto 2^x$,

$$\mathbb{E}_\tau|\{S_0, \ldots, S_{K-1}\}| \;\geq\; \mathbb{E}_\tau\big[2^{H(m_\tau)}\big] \;\geq\; 2^{\mathbb{E}_\tau H(m_\tau)} \;\geq\; 2^{C-U}.$$

This proves (8). □

When the bounds are tight. The lower bound holds with equality if and only if $I(T; \tau \mid X) = 0$: once the state is known, the rest of the run carries no further information about the time. It holds, for instance, when every run follows the same route ($U = 0$), in which case both bounds equal C.

B.5 Proof of the policy gradient theorem (Theorem 4, Section 4)

Let $\theta \in \mathbb{R}^p$, and assume that P and ρ do not depend on θ, that $\theta \mapsto \pi_\theta(a \mid s, t)$ is differentiable, and that $\pi_\theta(a \mid s, t) > 0$ for every allowed action. Write $d_t = d^\theta_t$ and $g_k = \nabla_\theta \log \pi_\theta(A_k \mid S_k, k)$ (natural logarithm). For a prefix $\tau_{0:t} = (s_0, a_0, \ldots, s_t)$ let $p_\theta(\tau_{0:t}) = \rho(s_0) \prod_{k<t} \pi_\theta(a_k \mid s_k, k) P(s_{k+1} \mid s_k, a_k)$.

Step 1: supports do not move. Because every policy factor is positive, $p_\theta(\tau_{0:t}) > 0$ if and only if $\rho(s_0) \prod_{k<t} P(s_{k+1} \mid s_k, a_k) > 0$, a condition that does not involve θ. Hence the set $\mathcal{S}_t = \{s : d_t(s) > 0\}$ is the same for all θ; on $\mathcal{S}_t$, $d_t(s)$ is positive and differentiable, and off $\mathcal{S}_t$ it is identically zero, with zero derivative.

Step 2: derivative of the entropy. On $\mathcal{S}_t$, $H(d_t) = -\sum_{s \in \mathcal{S}_t} d_t(s) \log_2 d_t(s)$ and

$$\nabla_\theta H(d_t) = -\sum_{s \in \mathcal{S}_t} \Big(\log_2 d_t(s) + \tfrac{1}{\ln 2}\Big) \nabla_\theta d_t(s) = -\sum_{s \in \mathcal{S}_t} \log_2 d_t(s)\, \nabla_\theta d_t(s),$$

because $\sum_{s \in \mathcal{S}_t} \nabla_\theta d_t(s) = \nabla_\theta \sum_s d_t(s) = \nabla_\theta 1 = 0$. This is why the density can be treated as a stopped-gradient target.

Step 3: likelihood ratio. Step 3: likelihood ratio. The event $\{S_t = s\}$ is the disjoint union, over prefixes $\tau_{0:t}$ with $s_t = s$, of the events that the run begins with $\tau_{0:t}$; by (15), summing out the transitions after time t, each such event has probability $p_\theta(\tau_{0:t})$. Hence $d_t(s) = \sum_{\tau_{0:t}: s_t = s} p_\theta(\tau_{0:t})$, a finite sum. For prefixes with $p_\theta(\tau_{0:t}) > 0$, $\nabla_\theta p_\theta = p_\theta \nabla_\theta \log p_\theta$ and $\nabla_\theta \log p_\theta(\tau_{0:t}) = \sum_{k<t} \nabla_\theta \log \pi_\theta(a_k \mid$

$s_k, k)$ since ρ and P do not depend on θ; prefixes of probability zero stay at zero by Step 1. Hence $\nabla_\theta d_t(s) = \mathbb{E}\big[\mathbf{1}\{S_t = s\} \sum_{k<t} g_k\big]$. Inserting this in Step 2, and using $S_t \in \mathcal{S}_t$ almost surely,

$$\nabla_\theta H(d_t) = \mathbb{E}\Big[-\log_2 d_t(S_t) \sum_{k=0}^{t-1} g_k\Big].$$

The integrand depends only on the prefix $\tau_{0:t}$, so the expectation may be taken over full runs. For $t = 0$ the sum is empty, consistent with $d_0 = \rho$.

Step 4: reindexing. By definition $U(\theta) = \sum_{t=0}^{K-1} w_t H(d_t)$, and the weights w_t do not depend on θ, so

$$\nabla_\theta U(\theta) = \sum_{t=0}^{K-1} w_t \, \nabla_\theta H(d_t).$$

The $t = 0$ term vanishes because $d_0 = \rho$ does not depend on θ. For $1 \le t \le K-1$, Step 3 gives $\nabla_\theta H(d_t) = \mathbb{E}\big[-\log_2 d_t(S_t) \sum_{k=0}^{t-1} g_k\big]$. Substituting, and using linearity of expectation over the finitely many terms,

$$\nabla_\theta U(\theta) = \sum_{t=1}^{K-1} w_t \, \mathbb{E}\Big[-\log_2 d_t(S_t) \sum_{k=0}^{t-1} g_k\Big] = \mathbb{E}\Big[\sum_{t=1}^{K-1} \sum_{k=0}^{t-1} w_t\big(-\log_2 d_t(S_t)\big) g_k\Big].$$

The double sum runs over all pairs (k, t) with $0 \le k < t \le K-1$. We now sum over the same pairs in the other order: first fix the action time k, which ranges over $0, \dots, K-2$; the state times paired with it are $t = k+1, \dots, K-1$. Since g_k does not depend on t, it factors out of the inner sum:

$$\sum_{t=1}^{K-1} \sum_{k=0}^{t-1} w_t\big(-\log_2 d_t(S_t)\big) g_k = \sum_{k=0}^{K-2} \sum_{t=k+1}^{K-1} w_t\big(-\log_2 d_t(S_t)\big) g_k = \sum_{k=0}^{K-2} g_k \sum_{t=k+1}^{K-1} w_t\big(-\log_2 d_t(S_t)\big).$$

Taking expectations gives

$$\nabla_\theta U(\theta) = \mathbb{E}\Big[\sum_{k=0}^{K-2} g_k \sum_{t=k+1}^{K-1} w_t\big(-\log_2 d_t(S_t)\big)\Big],$$

which is (12). The inner sum is the *future return* of the action at time k. It collects the scores of the states $S_{k+1}, \dots, S_{K-1}$ that come after that action, and the action cannot affect the earlier states.

Step 5: pooled entropy and the Lagrangian. $\bar{d} = \sum_t w_t d_t$ has support $\bigcup_t \mathcal{S}_t$, which is also fixed. Steps 2 and 3 applied to $C = H(\bar{d})$ with $\nabla \bar{d} = \sum_t w_t \nabla d_t$ give $\nabla_\theta C = \mathbb{E}\big[\sum_t w_t(-\log_2 \bar{d}(S_t)) \sum_{k<t} g_k\big]$, and the reindexing of Step 4 applies unchanged. Hence $\nabla_\theta \mathcal{L} = \nabla_\theta C - \lambda \nabla_\theta U = \mathbb{E}\big[\sum_k g_k G_k^\lambda(\tau)\big]$ with G_k^λ defined as:

$$G_k^\lambda(\tau) = \sum_{t=k+1}^{K-1} w_t \left[-\log_2 \bar{d}_{\gamma,K}(S_t) + \lambda \log_2 d_t(S_t)\right], \tag{16}$$

Step 6: baselines. Algorithm 1 subtracts a baseline b_k from each return to reduce the variance of the sampled gradient. We show that this does not change its expectation. Let b_k be a function of S_k, and possibly of data collected before the current batch, on which we condition; it must not depend on A_k or later variables. The per-time moving average used in Algorithm 1 is constant within a batch, so it is a special case. By the tower property, $\mathbb{E}[g_k b_k(S_k)] = \mathbb{E}\big[b_k(S_k)\, \mathbb{E}[g_k \mid S_k]\big]$, and

$$\mathbb{E}[g_k \mid S_k = s] = \sum_a \pi_\theta(a \mid s, k) \, \nabla_\theta \ln \pi_\theta(a \mid s, k) = \sum_a \nabla_\theta \pi_\theta(a \mid s, k) = \nabla_\theta \sum_a \pi_\theta(a \mid s, k) = \nabla_\theta 1 = 0.$$

Hence $\mathbb{E}\big[\sum_k g_k(G_k - b_k)\big] = \mathbb{E}\big[\sum_k g_k G_k\big]$: the baseline leaves the expected gradient unchanged and affects only its variance. □

B.6 Proof of the occupancy geometry (Proposition 5, Section 5)

Let $\mathcal{F}_K$ be the finite set of deterministic time-dependent policies $f = (f_t)_{t=0}^{K-2}$ with $f_t(s) \in \mathcal{A}(s)$. For $\pi \in \Pi_K$, the occupancy $\mu^\pi = (\mu_t^\pi)_{t=0}^{K-2}$ is $\mu_t^\pi(s,a) = \mathbb{P}_\pi(S_t = s, A_t = a)$. Every occupancy satisfies the linear *flow constraints*

$$\mu_t \geq 0, \qquad \sum_a \mu_0(s,a) = \rho(s), \qquad \sum_a \mu_{t+1}(s',a) = \sum_{s,a} P(s' \mid s,a)\, \mu_t(s,a) \quad (0 \leq t \leq K-3), \tag{17}$$

because $\mathbb{P}(S_{t+1} = s') = \sum_{s,a} \mathbb{P}(S_t = s, A_t = a) P(s' \mid s, a)$ by (15). The state laws are the linear images For each time $t < K$, define the map L_t that sends an occupancy sequence $\mu = (\mu_u)_{u=0}^{K-2}$ to a vector on $\mathcal{S}$:

$$(L_t\mu)(s) = \sum_a \mu_t(s,a) \quad (t \leq K-2), \qquad (L_{K-1}\mu)(s') = \sum_{s,a} P(s' \mid s,a)\, \mu_{K-2}(s,a). \tag{18}$$

Each L_t is linear in μ. For $t \leq K-2$ it sums out the action, and for $t = K-1$ it applies one transition, since no action is taken at the final time and μ stops at $K-2$. By (15), $d_t^\pi = L_t\mu^\pi$ for every $\pi \in \Pi_K$ and every $t < K$.

Since C and U depend on π only through $(d_t^\pi)_t$, they are functions of μ^π; we write $C(\mu)$ and $U(\mu)$.

Lemma 10 (Pre-sampling a randomized policy)**.** *Let $\pi \in \Pi_K$. Draw, independently of each other and of the initial state and transition noise, one action $F_t(s) \sim \pi_t(\cdot \mid s)$ for every pair (t,s), and let the agent play $A_t = F_t(S_t)$. The resulting run has law $\mathbb{P}_\pi$. Consequently*

$$\mu^\pi = \sum_{f \in \mathcal{F}_K} \lambda_f\, \mu^f, \qquad \lambda_f = \prod_{t=0}^{K-2} \prod_{s \in \mathcal{S}} \pi_t(f_t(s) \mid s),$$

and $(\lambda_f)_f$ is a probability vector.

Proof. Since the table F is drawn independently of the initial state and the transition noise, conditional on $F = f$ the pre-sampled run is a run of the deterministic policy f. Its law is therefore $\sum_f \mathbb{P}(F = f)\mathbb{P}_f = \sum_f \lambda_f \mathbb{P}_f$, and it suffices to show that this equals $\mathbb{P}_\pi$.

Fix a run $\tau = (s_0, a_0, \ldots, s_{K-1})$. Under a deterministic policy f, by (15),

$$\mathbb{P}_f(\tau) = \rho(s_0) \prod_{t=0}^{K-2} \mathbf{1}\{f_t(s_t) = a_t\}\, P(s_{t+1} \mid s_t, a_t).$$

Average this over f with weights λ_f. The factors ρ and P do not depend on f, so

$$\begin{aligned} \sum_f \lambda_f \mathbb{P}_f(\tau) &= \rho(s_0) \left(\prod_{t=0}^{K-2} P(s_{t+1} \mid s_t, a_t) \right) \cdot \sum_f \lambda_f \prod_{t=0}^{K-2} \mathbf{1}\{f_t(s_t) = a_t\} \\ &= \rho(s_0) \left(\prod_{t=0}^{K-2} P(s_{t+1} \mid s_t, a_t) \right) \cdot \mathbb{P}\big(F_t(s_t) = a_t \text{ for all } t \leq K-2\big), \end{aligned}$$

where F is a random table with law λ. Under λ the entries $F_t(s)$ are independent with $F_t(s) \sim \pi_t(\cdot \mid s)$. The entries $F_0(s_0), \ldots, F_{K-2}(s_{K-2})$ belong to different times, so they are distinct entries of the table, and hence independent. Therefore

$$\mathbb{P}\big(F_t(s_t) = a_t \text{ for all } t\big) = \prod_{t=0}^{K-2} \pi_t(a_t \mid s_t),$$

and $\sum_f \lambda_f \mathbb{P}_f(\tau) = \mathbb{P}_\pi(\tau)$ by (15). The two run laws agree for every run τ. Summing over the runs with $S_t = s$ and $A_t = a$ gives $\mu_t^\pi(s,a) = \sum_f \lambda_f \mu_t^f(s,a)$. Finally, λ is a product of probability vectors, so its entries are nonnegative and sum to one. □

The argument uses that the policy is indexed by time: each coordinate $F_t(s)$ is consulted at most once per run. It fails for stationary policies, which may revisit a state.

Lemma 11 (Markov realization)**.** *Let $\nu = (\nu_t)_{t=0}^{K-2}$ satisfy the flow constraints* (17)*. Let $n_t(s) = \sum_a \nu_t(s,a)$ and define $\pi_t(a \mid s) = \nu_t(s,a)/n_t(s)$ when $n_t(s) > 0$, and an arbitrary fixed distribution in $\Delta(\mathcal{A}(s))$ otherwise. Then $\pi \in \Pi_K$ and $\mu^\pi = \nu$.*

Proof. Each $\pi_t(\cdot \mid s)$ is a probability vector, so $\pi \in \Pi_K$. We show $\mu_t^\pi = \nu_t$ by induction. For $t = 0$, $d_0^\pi = \rho = n_0$ by (17), so $\mu_0^\pi(s,a) = \rho(s)\pi_0(a \mid s) = n_0(s)\pi_0(a \mid s) = \nu_0(s,a)$; when $n_0(s) = 0$ both sides vanish. If $\mu_t^\pi = \nu_t$, then by (15) and (17),

$$d_{t+1}^\pi(s') = \sum_{s,a} P(s' \mid s,a)\mu_t^\pi(s,a) = \sum_{s,a} P(s' \mid s,a)\nu_t(s,a) = n_{t+1}(s'),$$

so $\mu_{t+1}^\pi(s',a) = n_{t+1}(s')\pi_{t+1}(a \mid s') = \nu_{t+1}(s',a)$, again with both sides zero when $n_{t+1}(s') = 0$. □

Proof of (a). Let $\mathcal{Q} = \{\mu^\pi : \pi \in \Pi_K\}$ be the set of occupancies, and let $\mathcal{C} = \operatorname{conv}\{\mu^f : f \in \mathcal{F}_K\}$, where $\operatorname{conv} X$ denotes the *convex hull* of a set X, the set of all convex combinations $\sum_i \alpha_i x_i$ of finitely many points $x_i \in X$ with $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$.

Lemma 10 gives $\mathcal{M} \subseteq \mathcal{C}$. Conversely, every μ^f satisfies the linear constraints (17), so every convex combination does as well, and Lemma 11 realizes it by a policy in Π_K; hence $\mathcal{C} \subseteq \mathcal{M}$. Thus $\mathcal{M} = \mathcal{C}$, the convex hull of finitely many points, which is a polytope and in particular compact and convex.

For $\mu \in \mathcal{M}$, each $d_t = L_t\mu$ is a probability vector and L_t is linear, see (18). Composition of the concave function H (F5) with an affine map is concave, so $\mu \mapsto H(L_t\mu)$ and $\mu \mapsto C(\mu) = H\big(\sum_t w_t L_t \mu\big)$ are concave on $\mathcal{M}$. $U(\mu) = \sum_t w_t H(L_t\mu)$ is a nonnegative combination of concave functions and is therefore concave.

Proof of (b). Let $\pi \in \Pi_K$. By (a), $\mu^\pi = \sum_f \lambda_f \mu^f$ for some probability vector λ. Concavity gives

$$U(\pi) = U(\mu^\pi) \;\geq\; \sum_f \lambda_f U(\mu^f) \;\geq\; \min_{f \in \mathcal{F}_K} U(f).$$

$\mathcal{F}_K$ is finite, so the minimum on the right is attained by some $f^\star$, and $U(f^\star) \leq U(\pi)$ for every $\pi \in \Pi_K$.

Proof of (c). Let Z be the index of the component chosen at the start of the run: $\mathbb{P}(Z = i) = \alpha_i$, drawn independently of the initial state and the transition noise, after which the run follows π_Z. Then $\mathbb{P}(S_t = s, A_t = a) = \sum_i \alpha_i \mu_t^{\pi_i}(s,a)$, a convex combination of points of $\mathcal{Q}$. By (a) it lies in $\mathcal{Q}$ and is realized by some policy $\pi_{\text{mix}} \in \Pi_K$ (Lemma 11). Because C and U depend on a policy only through its occupancy, the mixture's values are $C(\sum_i \alpha_i \mu^{\pi_i})$ and $U(\sum_i \alpha_i \mu^{\pi_i})$.

The two lower bounds are concavity of C and U from (a). For the upper bound on U, fix t. Under the mixture, $d_t = \sum_i \alpha_i d_t^{(i)}$, where $d_t^{(i)}$ is the law of S_t under π_i. By (F2) and (F1),

$$H(d_t) = H(S_t) \;\leq\; H(S_t, Z) = H(Z) + H(S_t \mid Z) = H(\alpha) + \sum_i \alpha_i H\big(d_t^{(i)}\big).$$

Multiplying by w_t and summing over t, with $\sum_t w_t = 1$, gives $U(\text{mix}) \leq H(\alpha) + \sum_i \alpha_i U(\pi_i)$.

A sharper bound. By the chain rule, $H(S_t) = H(S_t \mid Z) + I(Z; S_t)$ exactly, so the increase over the average component uncertainty is

$$U(\text{mix}) - \sum_i \alpha_i U(\pi_i) = \sum_t w_t \, I(Z; S_t).$$

Each term satisfies $I(Z; S_t) \leq H(Z) = H(\alpha)$. At time zero all components start from the same law ρ, so S_0 is independent of Z and $I(Z; S_0) = 0$. Hence

$$U(\text{mix}) - \sum_i \alpha_i U(\pi_i) \leq (1 - w_0) H(\alpha).$$

Equality holds if the components with $\alpha_i > 0$ have disjoint state supports at every $t \geq 1$, since then S_t determines Z and $I(Z; S_t) = H(\alpha)$. In particular, the bound $H(\alpha)$ stated in (c) is valid but is never attained by a mixture with $H(\alpha) > 0$, because $w_0 > 0$.

Proof of (d). We first show a general mechanism and then give a concrete instance.

Two distinct routes break convexity. Let the MDP be deterministic and let f and f' be deterministic policies with routes $(s_t)_{t<K}$ and $(s'_t)_{t<K}$. All their state laws are point masses, so $U(f) = U(f') = 0$. Let $\mathcal{T} = \{t : s_t \neq s'_t\}$ and suppose $\mathcal{T} \neq \emptyset$. By (a), $\mathcal{Q}$ is convex, so the midpoint $\bar{\mu} = \frac{1}{2}\mu^f + \frac{1}{2}\mu^{f'}$ lies in $\mathcal{Q}$, and since each L_t is linear, its state laws are $d_t = L_t\bar{\mu} = \frac{1}{2}\delta_{s_t} + \frac{1}{2}\delta_{s'_t}$. Hence $H(d_t) = 1$ for $t \in \mathcal{T}$ and $H(d_t) = 0$ otherwise, so

$$U(\bar{\mu}) = \sum_{t\in\mathcal{T}} w_t > 0.$$

For every $0 \leq \varepsilon < \sum_{t\in\mathcal{T}} w_t$, the set $\{\mu \in \mathcal{Q} : U(\mu) \leq \varepsilon\}$ therefore contains μ^f and $\mu^{f'}$ but not their midpoint, so it is not convex.

A concrete instance. Take the two-room MDP of Figure 1: states $\{h, L, R\}$, $\rho = \delta_h$; from h the action L or R moves deterministically to that room, and from either room every action returns to h. Let $K = 5$. Let f^{LR} choose L at time 0 and R at time 2, and let f^{RL} do the reverse. Their routes are (h, L, h, R, h) and (h, R, h, L, h), which differ exactly at $\mathcal{T} = \{1, 3\}$. By the argument above, the midpoint has $d_1 = d_3 = (0, \frac{1}{2}, \frac{1}{2})$, point masses at even times, and $U = w_1 + w_3 > 0$. So the feasible set is not convex for any $0 \leq \varepsilon < w_1 + w_3$.

The complement is always convex. In every MDP and for every ε, the set $\{\mu \in \mathcal{Q} : U(\mu) > \varepsilon\}$ is convex. Indeed, if $U(\mu) > \varepsilon$ and $U(\mu') > \varepsilon$, then for $\alpha \in [0, 1]$ concavity of U from (a) gives $U(\alpha\mu + (1-\alpha)\mu') \geq \alpha U(\mu) + (1-\alpha)U(\mu') > \varepsilon$. The feasible set is thus the complement of a convex set within the convex set $\mathcal{Q}$; such sets are called *reverse convex*. □

B.7 A COVERAGE LEMMA (USED FOR THEOREM 6 AND PROPOSITION 7)

The next lemma bounds the pooled entropy of any policy whose state laws stay close to a fixed sequence of states. It holds in every finite MDP and is used in the proofs of Theorem 6 and Proposition 7. We use the elementary bound, valid for $x \in [0, 1)$,

$$h_2(x) = x\log_2\tfrac{1}{x} + (1-x)\log_2\tfrac{1}{1-x} \;\leq\; x\log_2\tfrac{1}{x} + x\log_2 e, \tag{19}$$

which follows from $-(1-x)\ln(1-x) \leq x$ (the function $x + (1-x)\ln(1-x)$ vanishes at 0 and has derivative $-\ln(1-x) \geq 0$).

Lemma 12 (Coverage near a sequence of states)**.** *Let $\pi \in \Pi_K$, let $(s_t)_{t<K}$ be any states, and let $m(s) = \sum_{t:s_t=s} w_t$. Let $\kappa_t = 1 - d_t^\pi(s_t)$ be the probability that the run is not at s_t at time t, and $\Delta = \sum_t w_t\kappa_t$. If $\Delta < 1$, then*

$$C(\pi) \;\leq\; H(m) + \Delta\log_2\tfrac{1}{\Delta} + \Delta\log_2\frac{e\,n}{w_{K-1}},$$

with $0\log_2(1/0) = 0$.

Proof. Let $\mathcal{R} = \{s_0, \ldots, s_{K-1}\}$; m is positive exactly on $\mathcal{R}$, with $m(s) \geq w_{K-1}$ there. Let $T \sim w$ be independent of the run, $Y = S_T$ and $Z = \mathbf{1}\{S_T \neq s_T\}$, so that $H(Y) = C(\pi)$ as in Section B.2. Since T is independent of the run,

$$\mathbb{P}(Z = 1) = \sum_t \mathbb{P}(T = t)\,\mathbb{P}(S_t \neq s_t \mid T = t) = \sum_t w_t\big(1 - d_t^\pi(s_t)\big) = \sum_t w_t\kappa_t = \Delta.$$

By the chain rule (F1) in two orders, $H(Y) + H(Z \mid Y) = H(Y, Z) = H(Z) + H(Y \mid Z)$. Since $H(Z \mid Y) \geq 0$, this gives $H(Y) \leq H(Z) + H(Y \mid Z)$. Here Z is a binary variable with $\mathbb{P}(Z = 1) = \Delta$, so $H(Z) = h_2(\Delta)$, and by the definition of conditional entropy $H(Y \mid Z) = (1-\Delta)H(Y \mid Z = 0) + \Delta H(Y \mid Z = 1)$. Hence

$$C(\pi) = H(Y) \;\leq\; h_2(\Delta) + (1-\Delta)H(Y \mid Z = 0) + \Delta H(Y \mid Z = 1),$$

and $H(Y \mid Z = 1) \leq \log_2 n$ by (F4).

On the event $Z = 0$ we have $S_T = s_T$, so $Y = s_T$. Conditioning on T, which is independent of the run,

$$\mathbb{P}(Y = s, Z = 0) = \sum_t w_t\,\mathbb{P}(S_t = s_t)\,\mathbf{1}\{s_t = s\} = \sum_{t:\,s_t=s} w_t(1-\kappa_t).$$

Dividing by $\mathbb{P}(Z=0) = 1-\Delta > 0$, the conditional law of Y given $Z=0$ is

$$\tilde{m}(s) = \frac{1}{1-\Delta} \sum_{t:\, s_t = s} w_t(1-\kappa_t).$$

The sum is empty unless $s = s_t$ for some t, so $\tilde{m}$ is supported in $\mathcal{R}$. Compared with $m(s) = \sum_{t:\, s_t=s} w_t$, each weight w_t is scaled by the probability $1-\kappa_t$ of being on the sequence at time t, and the result is renormalized.

Gibbs' inequality (nonnegativity of $\mathrm{KL}(\tilde{m}\|m)$, valid since $\tilde{m}$ is supported in $\mathcal{R} = \mathrm{supp}(m)$) gives

$$H(\tilde{m}) \;\le\; -\sum_s \tilde{m}(s)\log_2 m(s) = H(m) + \sum_{s\in\mathcal{R}} \big(\tilde{m}(s) - m(s)\big)\log_2 \tfrac{1}{m(s)}.$$

In the last sum, terms with $\tilde{m}(s) < m(s)$ are nonpositive. Since $m(s) \ge w_{K-1}$ on $\mathcal{R}$, the other terms are at most $(\tilde{m}(s) - m(s))\log_2(1/w_{K-1})$. Both $\tilde{m}$ and m sum to one, so the positive parts of $\tilde{m} - m$ sum to $\frac{1}{2}\|\tilde{m} - m\|_1$. By the triangle inequality,

$$\|\tilde{m} - m\|_1 \le \sum_t w_t \Big|\frac{1-\kappa_t}{1-\Delta} - 1\Big| = \frac{\sum_t w_t|\Delta - \kappa_t|}{1-\Delta} \le \frac{\sum_t w_t(\Delta+\kappa_t)}{1-\Delta} = \frac{2\Delta}{1-\Delta}.$$

Hence $H(\tilde{m}) \le H(m) + \frac{\Delta}{1-\Delta}\log_2 \frac{1}{w_{K-1}}$, that is,

$$(1-\Delta)H(Y \mid Z=0) = (1-\Delta)H(\tilde{m}) \le (1-\Delta)H(m) + \Delta\log_2 \tfrac{1}{w_{K-1}}.$$

Substituting this and $H(Y \mid Z=1) \le \log_2 n$ into the bound on $C(\pi)$ above, and dropping $-\Delta H(m) \le 0$,

$$C(\pi) \;\le\; h_2(\Delta) + H(m) + \Delta\log_2 \tfrac{1}{w_{K-1}} + \Delta\log_2 n.$$

Finally, (19) gives $h_2(\Delta) \le \Delta\log_2 \frac{1}{\Delta} + \Delta\log_2 e$, so

$$C(\pi) \;\le\; H(m) + \Delta\log_2 \tfrac{1}{\Delta} + \Delta\log_2 \frac{e\,n}{w_{K-1}},$$

which is the claim. □

B.8 Proof that choosing a certain route is hard (Theorem 6, Section 5)

Source problem. Directed Hamiltonian Path with Start (DHP-S): given a directed graph $G = (V, E)$ and $v_0 \in V$, decide whether there is a directed path starting at v_0 that visits every vertex exactly once. DHP-S is NP-complete. It is in NP, and Directed Hamiltonian Cycle, which is NP-complete (Karp, 1972), reduces to it: pick any vertex u, replace it by a vertex u^{out} carrying all edges out of u and a vertex u^{in} carrying all edges into u, and set $v_0 = u^{\mathrm{out}}$. Since u^{in} has no outgoing edge, a Hamiltonian path from u^{out} must end at u^{in}, and such paths correspond one-to-one to Hamiltonian cycles of G through u.

Construction. Given (G, v_0) with $n = |V| \ge 2$, let $\mathcal{M}_G$ be the MDP with states V, initial law $\rho = \delta_{v_0}$, and at each v one action stay with $P(v \mid v, \mathsf{stay}) = 1$ plus one action go_e for every edge $e = (v, v') \in E$ with $P(v' \mid v, \mathsf{go}_e) = 1$. Every action set is nonempty and all transitions are deterministic. Set $K = n$ and $\gamma = 1 - 1/n$, a rational in $(0, 1)$. The construction takes time polynomial in $|V| + |E|$.

A *walk* is a sequence $v_0 = s_0, s_1, \ldots, s_{K-1}$ with $s_{t+1} = s_t$ or $(s_t, s_{t+1}) \in E$. For a walk, let $m(v) = \sum_{t:s_t=v} w_t$.

Lemma 13 (Zero budget means one route)**.** *In $\mathcal{M}_G$, a policy $\pi \in \Pi_K$ satisfies $U_{\gamma,K}(\pi) = 0$ if and only if there is a walk (s_t) with $d_t^\pi = \delta_{s_t}$ for all $t < K$. In that case $C_{\gamma,K}(\pi) = H(m)$. Conversely, every walk is followed by some deterministic policy with $U = 0$.*

Proof. Since every $w_t > 0$ and $H \ge 0$, $U(\pi) = 0$ if and only if $H(d_t^\pi) = 0$ for all t, that is, by (F4), every d_t^π is a point mass δ_{s_t}. Then $s_0 = v_0$, and since d_{t+1}^π is the law of the deterministic successor of s_t under the action drawn from $\pi_t(\cdot \mid s_t)$, that successor is s_{t+1} for every action in the support;

hence (s_t) is a walk. The pooled law is then $\bar{d}(v) = \sum_t w_t \mathbf{1}\{s_t = v\} = m(v)$, so $C = H(m)$. For the converse, given a walk let $f_t(s_t)$ be stay if $s_{t+1} = s_t$ and $\mathsf{go}_{(s_t, s_{t+1})}$ otherwise, with arbitrary actions at other states. We show $d_t^f = \delta_{s_t}$ by induction on t. For $t = 0$, $d_0^f = \rho = \delta_{v_0} = \delta_{s_0}$. If $S_t = s_t$ with probability one, then f plays the action $f_t(s_t)$, whose deterministic successor is s_{t+1} by construction, so $S_{t+1} = s_{t+1}$ with probability one. □

Lemma 14 (Repetition costs coverage)**.** *Let $w = (w_0, \dots, w_{K-1})$ be the time weights of* (2)*, viewed as a distribution on $\{0, \dots, K-1\}$. If the walk visits K distinct states then $H(m) = H(w)$. If $s_i = s_j$ for some $i \neq j$, then $H(m) \leq H(w) - w_{K-1}$.*

Proof. Let $T \sim w$. The state visited at time T, namely s_T, has law

$$\mathbb{P}(s_T = v) = \sum_{t:\, s_t = v} w_t = m(v),$$

so $H(m) = H(s_T)$, and $s_T = g(T)$ is a function of T with $g(t) = s_t$.

Distinct states. If $s_0, \dots, s_{K-1}$ are distinct, g is one-to-one, so T is also a function of s_T. Then $H(T \mid s_T) = 0$ and, by (F6), $H(m) = H(s_T) = H(T) - H(T \mid s_T) = H(w)$. Equivalently, m is w with its atoms relabeled by the states.

A repeated state. Suppose $s_i = s_j$ with $i \neq j$. Merge the two times into one: let $T' = i$ if $T \in \{i, j\}$ and $T' = T$ otherwise. The law w' of T' is w with atoms i and j merged into one atom of mass $w_i + w_j$. Since $s_i = s_j$, the state s_T is determined by T': $s_T = s_{T'}$. By (F6), merging outcomes cannot increase entropy, so

$$H(m) = H(s_T) \leq H(T') = H(w').$$

Lemma 9, applied to $q = w$ and the atoms $i \neq j$, gives

$$H(w) - H(w') = w_i \log_2 \frac{w_i + w_j}{w_i} + w_j \log_2 \frac{w_i + w_j}{w_j} \;\geq\; \min\{w_i, w_j\}.$$

To see the inequality, let $b = \min\{w_i, w_j\}$ and $a = \max\{w_i, w_j\}$. Both terms are nonnegative, and the term belonging to b is $b \log_2 \frac{a+b}{b} \geq b \log_2 2 = b$, because $a \geq b$.

The weights decrease in time: $w_{t+1}/w_t = \gamma < 1$ by (2), so $w_0 > w_1 > \cdots > w_{K-1}$. Hence the smaller of w_i and w_j is the one with the later time, $\min\{w_i, w_j\} = w_{\max\{i,j\}}$, and since $\max\{i, j\} \leq K - 1$,

$$\min\{w_i, w_j\} \;\geq\; w_{K-1}.$$

Combining the three inequalities,

$$H(m) \;\leq\; H(w') \;\leq\; H(w) - \min\{w_i, w_j\} \;\leq\; H(w) - w_{K-1}.$$

□

Lemma 15 (Size of the smallest weight)**.** *For $K = n \geq 2$ and $\gamma = 1 - 1/n$, $w_{K-1} \geq 1/(en) > 1/(3n)$.*

Proof. By (2) with $K = n$, $w_{K-1} = (1 - \gamma)\gamma^{n-1}/(1 - \gamma^n)$.

Step 1: drop the denominator. Since $0 < \gamma < 1$, we have $0 < 1 - \gamma^n \leq 1$, and dividing by a number in $(0, 1]$ does not decrease a positive quantity. With $1 - \gamma = \frac{1}{n}$,

$$w_{K-1} \;\geq\; (1 - \gamma)\gamma^{n-1} = \frac{1}{n}\Big(1 - \frac{1}{n}\Big)^{n-1}.$$

Step 2: bound the power. Write $1 - \frac{1}{n} = \frac{n-1}{n} = \big(1 + \frac{1}{n-1}\big)^{-1}$, so

$$\Big(1 - \frac{1}{n}\Big)^{n-1} = \Big(1 + \frac{1}{n-1}\Big)^{-(n-1)}.$$

For every $x > 0$, $\ln(1+x) \le x$, hence $(1+x)^{1/x} \le e$. Taking $x = \frac{1}{n-1}$ gives $\left(1+\frac{1}{n-1}\right)^{n-1} \le e$, and taking reciprocals,

$$\left(1-\frac{1}{n}\right)^{n-1} \ge e^{-1}.$$

Step 3: combine. Steps 1 and 2 give $w_{K-1} \ge \frac{1}{en}$, and $e < 3$ gives $\frac{1}{en} > \frac{1}{3n}$. □

Lemma 16 (A small budget forces a near-route)**.** *In $\mathcal{M}_G$, let $\pi \in \Pi_K$ satisfy $U(\pi) \le \varepsilon$ with $\eta := \varepsilon / w_{K-1} < 1$. Then for every $t < K$ there is a unique state s_t with $d_t^\pi(s_t) > \frac{1}{2}$; the off-route probabilities $\kappa_t = 1 - d_t^\pi(s_t)$ satisfy $\kappa_t \le \eta/2$; and (s_t) is a walk.*

Proof. Since all terms of U are nonnegative, $w_t H(d_t^\pi) \le U(\pi) \le \varepsilon$, so $H(d_t^\pi) \le \varepsilon / w_t \le \eta < 1$. If every state had $d_t^\pi(s) \le \frac{1}{2}$, then $\log_2(1/d_t^\pi(s)) \ge 1$ on the support and $H(d_t^\pi) \ge 1$, a contradiction; so some state s_t has mass above $\frac{1}{2}$, and it is unique. Grouping $\{s_t\}$ against its complement (F6) gives $h_2(\kappa_t) \le H(d_t^\pi) \le \eta$. The binary entropy is concave with $h_2(0) = 0$ and $h_2(\frac{1}{2}) = 1$, so $h_2(x) \ge 2x$ on $[0, \frac{1}{2}]$; since $\kappa_t < \frac{1}{2}$, this gives $\kappa_t \le \eta/2$.

For the walk property, $d_0^\pi = \delta_{v_0}$ gives $s_0 = v_0$. Transitions are deterministic, so $d_{t+1}^\pi(s') = \sum_s d_t^\pi(s) \sum_a \pi_t(a \mid s) \mathbf{1}\{s \xrightarrow{a} s'\}$. If s_{t+1} were not a one-step successor of s_t, the term $s = s_t$ would vanish and $d_{t+1}^\pi(s_{t+1}) \le \sum_{s \ne s_t} d_t^\pi(s) = \kappa_t < \frac{1}{2}$, contradicting $d_{t+1}^\pi(s_{t+1}) > \frac{1}{2}$. In $\mathcal{M}_G$ the one-step successors of v are v itself and the heads of edges out of v, so (s_t) is a walk. □

The reduction. Fix a budget $\varepsilon = \varepsilon_n$ with $0 \le \varepsilon \le 1/(1000\, e\, n^3)$, and let $\mathrm{OPT}_\varepsilon = \max\{C(\pi) : \pi \in \Pi_K,\ U(\pi) \le \varepsilon\}$. The maximum exists. By Proposition 5(a), the set $\mathcal{Q}$ of occupancies is compact, and C and U are continuous functions of the occupancy, so the feasible set $\{\mu \in \mathcal{Q} : U(\mu) \le \varepsilon\}$ is compact. It is nonempty, since the policy that always plays stay has $U = 0$. Hence C attains its maximum on it, at an occupancy that is realized by some policy in Π_K (Lemma 11). Suppose an algorithm Alg runs in polynomial time and, on every instance of the form above, returns a rational policy table $\hat\pi \in \Pi_K$ with $U(\hat\pi) \le \varepsilon$ and $C(\hat\pi) \ge \mathrm{OPT}_\varepsilon - 1/(3n)$.

We decide DHP-S as follows: build $\mathcal{M}_G$, run Alg, compute the laws $d_t^{\hat\pi}$ exactly by forward propagation in rational arithmetic ($K = n$ steps over n states, with bit lengths growing additively), let s_t be the state with $d_t^{\hat\pi}(s_t) > \frac{1}{2}$, and answer "yes" if and only if $s_0, \dots, s_{K-1}$ are distinct. Every step is polynomial.

The rounded route. By Lemma 15, $\eta = \varepsilon / w_{K-1} \le en\varepsilon \le 1/(1000n^2) < 1$, so Lemma 16 applies: the states s_t exist, form a walk, and $\Delta = \sum_t w_t \kappa_t \le \max_t \kappa_t \le x_0 := 1/(2000n^2)$. The map $x \mapsto x \log_2(1/x)$ is increasing on $[0, 1/e]$ and $x_0 < 1/e$, and $en/w_{K-1} \le e^2 n^2$ by Lemma 15. So Lemma 12 gives

$$C(\hat\pi) \ \le\ H(m) + g_n, \qquad g_n := x_0 \log_2 \tfrac{1}{x_0} + x_0 \log_2(e^2 n^2) = \frac{\log_2(2000\, e^2 n^4)}{2000\, n^2}, \tag{20}$$

where m is the pooled law of the walk. Moreover $g_n < \frac{3-e}{3e\,n}$ for every $n \ge 2$. Indeed, with $C_0 = 2000e^2$,

$$\frac{g_n}{(3-e)/(3en)} = \frac{3e}{2000(3-e)} \cdot \frac{\log_2(C_0 n^4)}{n},$$

and $\frac{d}{dn}\frac{\log_2(C_0 n^4)}{n} = \frac{4/\ln 2 - \log_2(C_0 n^4)}{n^2} < 0$ because $n \ge 2$ gives $\log_2(C_0 n^4) \ge \log_2(16 C_0) > 17 > 4/\ln 2$. So the ratio is largest at $n = 2$, where it equals $0.129\ldots < 1$.

Yes-instances. A Hamiltonian path from v_0 is a walk with $n = K$ distinct states; by Lemmas 13 and 14 it is followed by a policy with $U = 0 \le \varepsilon$ and $C = H(w)$, so $\mathrm{OPT}_\varepsilon \ge H(w)$. By (20) and the guarantee of Alg,

$$H(m) \ \ge\ C(\hat\pi) - g_n \ \ge\ H(w) - \frac{1}{3n} - g_n \ >\ H(w) - \frac{1}{en} \ \ge\ H(w) - w_{K-1},$$

using $\frac{1}{3n} + g_n < \frac{1}{en}$ and Lemma 15. By Lemma 14, the walk (s_t) has no repeated state, and we answer "yes".

No-instances. A walk with $K = n$ distinct states never uses stay, so it would be a directed path through all of V starting at v_0, which does not exist. Hence the walk (s_t) repeats a state and we answer "no".

Thus Alg would decide an NP-complete problem in polynomial time, which is impossible unless P = NP. For $\varepsilon = 0$ the rounding is exact ($\Delta = 0$), and the argument reduces to Lemma 13. □

Remarks. (i) The accuracy $1/(3n)$ is polynomial because $\gamma = 1-1/n$ grows with the instance. For a fixed rational γ and zero budget, the same argument, with $1/(3n)$ replaced by a suitable fraction of $w_{K-1} = \Theta(\gamma^n)$, gives hardness for exponentially small accuracy. A positive budget would also need to shrink enough to preserve the rounding gap. (ii) The budget may be any value up to $1/(1000\, e\, n^3)$, including zero. Hardness for budgets bounded away from zero is left open; the constants above were chosen for simplicity, not optimized. The snapshot-only result Theorem 22 already holds for a fixed additive accuracy.

B.9 Proof of the threshold at $\lambda = 1$ (Proposition 7, Section 5)

Throughout, the MDP is deterministic, $\rho = \delta_{s_0}$, and f is a deterministic policy whose route is $(s_t)_{t<K}$, so $d_t^f = \delta_{s_t}$, $U(f) = 0$, and $C(f) = H(m)$ with $m(s) = \sum_{t:s_t=s} w_t$. Let $\mathcal{R} = \{s_0, \dots, s_{K-1}\}$ be the set of route states; m is positive exactly on $\mathcal{R}$, and $m(s) \geq w_{K-1}$ for $s \in \mathcal{R}$. Write $\mathcal{L}_\lambda = C - \lambda U$ for $\lambda \geq 0$ (the constant $\lambda\varepsilon$ plays no role), and recall (19).

Part (a). Fix $\pi \in \Pi_K$ and let $\kappa_t = 1 - d_t^\pi(s_t)$ be the probability that π is off the route at time t, $\bar\kappa = \max_t \kappa_t$, and $\Delta = \sum_t w_t\kappa_t$. Assume $0 < \bar\kappa < r_\lambda$. Since $r_\lambda < 1$, we have $\Delta \leq \bar\kappa < 1$, and $\Delta \geq w_{t^\star}\bar\kappa \geq w_{K-1}\bar\kappa > 0$ where $t^\star$ attains the maximum.

Lower bound on $U(\pi)$. Grouping the states of S_t into $\{s_t\}$ and its complement (F6), $H(d_t^\pi) \geq h_2(\kappa_t) \geq \kappa_t \log_2(1/\kappa_t) \geq \kappa_t \log_2(1/\bar\kappa)$, the last step because $\kappa_t \leq \bar\kappa$. Summing with weights,

$$U(\pi) \;\geq\; \Delta \log_2(1/\bar\kappa). \tag{21}$$

Upper bound on $C(\pi)$. Apply Lemma 12 to π and the route (s_t). Its hypothesis $\Delta < 1$ holds, and for the route $H(m) = C(f)$. The lemma gives

$$C(\pi) \;\leq\; C(f) + \Delta \log_2 \tfrac{1}{\Delta} + \Delta \log_2 \frac{e\, n}{w_{K-1}},$$

that is, splitting the last logarithm,

$$C(\pi) - C(f) \;\leq\; \Delta \log_2 \tfrac{1}{\Delta} + \Delta \log_2 e + \Delta \log_2 \tfrac{1}{w_{K-1}} + \Delta \log_2 n.$$

The first term still contains Δ inside the logarithm; we bound it in terms of $\bar\kappa$. All terms of $\Delta = \sum_t w_t\kappa_t$ are nonnegative, so keeping only the term at $t^\star$, where $\kappa_{t^\star} = \bar\kappa$, gives

$$\Delta \;\geq\; w_{t^\star}\bar\kappa \;\geq\; w_{K-1}\bar\kappa,$$

since the weights decrease in t. Taking reciprocals and logarithms,

$$\log_2 \tfrac{1}{\Delta} \;\leq\; \log_2 \frac{1}{w_{K-1}\bar\kappa} = \log_2 \tfrac{1}{\bar\kappa} + \log_2 \tfrac{1}{w_{K-1}}.$$

Multiplying by $\Delta > 0$ and substituting,

$$C(\pi) - C(f) \;\leq\; \Delta\Big[\log_2 \tfrac{1}{\bar\kappa} + \log_2 \tfrac{1}{w_{K-1}} + \log_2 e + \log_2 \tfrac{1}{w_{K-1}} + \log_2 n\Big].$$

Collecting the terms that do not depend on π,

$$C(\pi) - C(f) \;\leq\; \Delta\Big[\log_2 \tfrac{1}{\bar\kappa} + A\Big], \qquad A = 2\log_2 \tfrac{1}{w_{K-1}} + \log_2(en). \tag{22}$$

Conclusion. Since $U(f) = 0$,

$$\mathcal{L}_\lambda(\pi) - \mathcal{L}_\lambda(f) = \big[C(\pi) - C(f)\big] - \lambda\, U(\pi).$$

Bound the first bracket from above by (22), and bound $-\lambda U(\pi)$ from above by (21), which gives $-\lambda U(\pi) \le -\lambda\Delta \log_2(1/\bar\kappa)$. Adding,

$$\mathcal{L}_\lambda(\pi) - \mathcal{L}_\lambda(f) \;\le\; \Delta\Big[\log_2 \tfrac{1}{\bar\kappa} + A\Big] - \lambda\Delta \log_2 \tfrac{1}{\bar\kappa} = \Delta\Big[(1-\lambda)\log_2 \tfrac{1}{\bar\kappa} + A\Big].$$

Since $\Delta > 0$, the right side is negative if and only if the bracket is negative. Since $\lambda > 1$, we can divide by $\lambda - 1 > 0$:

$$(1-\lambda)\log_2 \tfrac{1}{\bar\kappa} + A < 0 \iff \log_2 \tfrac{1}{\bar\kappa} > \frac{A}{\lambda-1} \iff \bar\kappa < 2^{-A/(\lambda-1)}.$$

By the definition of A,

$$2^{-A} = 2^{-2\log_2(1/w_{K-1}) - \log_2(en)} = \frac{w_{K-1}^2}{en}, \qquad \text{so} \qquad 2^{-A/(\lambda-1)} = \Big(\frac{w_{K-1}^2}{en}\Big)^{1/(\lambda-1)} = r_\lambda.$$

Hence $0 < \bar\kappa < r_\lambda$ implies $\mathcal{L}_\lambda(\pi) < \mathcal{L}_\lambda(f)$. □

Part (b). *Setup.* Let f' agree with f except $f'_k(s_k) = a$, and let (s'_t) be its route. As in Proposition 7, let

$$W = \sum_{t:\, s'_t \neq s_t} w_t, \qquad M = \sum_{t:\, s'_t \notin \mathcal{R}} w_t,$$

where $\mathcal{R} = \{s_0, \dots, s_{K-1}\}$ is the set of states on the route of f. Thus W is the weight of the times at which the two routes differ, and M the weight of the times at which the new route is at a state that f never visits.

W is the weight of the times at which the two routes differ, and M the weight of the times at which the new route is at a state that f never visits. A time with $s'_t \notin \mathcal{R}$ has $s'_t \neq s_t$, so $M \le W$. For $\xi \in (0,1)$, let π_ξ agree with f except that at time k in state s_k it plays a with probability ξ and $f_k(s_k)$ with probability $1-\xi$. So π_ξ takes the detour with a small probability ξ.

Step 1: state laws. The state s_k is visited at time k with probability one, and transitions are deterministic. So a run of π_ξ follows the route of f with probability $1-\xi$ and the route of f' with probability ξ. Hence

$$d_t^{\pi_\xi} = (1-\xi)\,\delta_{s_t} + \xi\,\delta_{s'_t}, \qquad \bar d^{\pi_\xi} = q := (1-\xi)m + \xi m',$$

where $m'(s) = \sum_{t: s'_t = s} w_t$ is the pooled law of the route of f'.

Step 2: snapshot entropy. If $s'_t = s_t$, then $d_t^{\pi_\xi}$ is a point mass and has entropy 0. If $s'_t \neq s_t$, it puts mass $1-\xi$ and ξ on two states and has entropy $h_2(\xi)$. By definition $U(\pi_\xi) = \sum_t w_t H(d_t^{\pi_\xi})$. The terms with $s'_t = s_t$ vanish, and the others all equal $w_t h_2(\xi)$, so

$$U(\pi_\xi) = \sum_{t:\, s'_t \neq s_t} w_t\, h_2(\xi) = W\, h_2(\xi).$$

Applying (19), $h_2(\xi) \le \xi \log_2 \frac{1}{\xi} + \xi \log_2 e$, and multiplying by $W \ge 0$,

$$U(\pi_\xi) \le W\big(\xi \log_2 \tfrac{1}{\xi} + \xi \log_2 e\big).$$

Step 3: pooled entropy. Split $-\sum_s q(s)\log_2 q(s)$ according to $q = (1-\xi)m + \xi m'$, and use $-\sum_s m(s)\log_2 q(s) = H(m) + \mathrm{KL}(m\|q)$ and the same identity for m':

$$H(q) = (1-\xi)\big[H(m) + \mathrm{KL}(m\|q)\big] + \xi\big[H(m') + \mathrm{KL}(m'\|q)\big].$$

Drop the nonnegative terms $\mathrm{KL}(m\|q)$ and $H(m')$, and use $(1-\xi)H(m) = H(m) - \xi H(m) \ge C(f) - \xi \log_2 n$:

$$C(\pi_\xi) = H(q) \;\ge\; C(f) - \xi \log_2 n + \xi\, \mathrm{KL}(m'\|q).$$

It remains to bound $\mathrm{KL}(m'\|q) = \sum_s m'(s)\log_2 \frac{m'(s)}{q(s)}$ from below. Split the sum over $\mathcal{O} = \mathcal{S} \setminus \mathcal{R}$ and $\mathcal{R}$.

- On $\mathcal{O}$, $m = 0$, so $q = \xi m'$ and each term equals $m'(s)\log_2(1/\xi)$. These terms sum to $m'(\mathcal{O})\log_2(1/\xi) = M \log_2(1/\xi)$, since $m'(\mathcal{O}) = \sum_{t: s'_t \notin \mathcal{R}} w_t = M$.

- On $\mathcal{R}$, the log-sum inequality gives $\sum_{s\in\mathcal{R}} m'(s)\log_2\frac{m'(s)}{q(s)} \;\geq\; m'(\mathcal{R})\log_2\frac{m'(\mathcal{R})}{q(\mathcal{R})} \;\geq\; m'(\mathcal{R})\log_2 m'(\mathcal{R})$, because $q(\mathcal{R}) \leq 1$. The function $x\log_2 x$ has minimum $-\log_2(e)/e$ on $[0,1]$, attained at $x = 1/e$.

Hence $\mathrm{KL}(m'\|q) \geq M\log_2(1/\xi) - \log_2(e)/e$, and

$$C(\pi_\xi) \;\geq\; C(f) + \xi\Big[M\log_2\tfrac{1}{\xi} - \log_2 n - \tfrac{\log_2 e}{e}\Big].$$

Step 4: conclusion. Since $U(f) = 0$, $\mathcal{L}_\lambda(\pi_\xi) - \mathcal{L}_\lambda(f) = \big[C(\pi_\xi) - C(f)\big] - \lambda U(\pi_\xi)$. Steps 2 and 3 give

$$\mathcal{L}_\lambda(\pi_\xi) - \mathcal{L}_\lambda(f) \;\geq\; \xi\Big[(M-\lambda W)\log_2\tfrac{1}{\xi} - c\Big], \qquad c = \log_2 n + \tfrac{\log_2 e}{e} + \lambda W\log_2 e > 0.$$

Suppose $M > \lambda W$. Since $\xi > 0$, the right side is positive when the bracket is, that is, when

$$\log_2\tfrac{1}{\xi} > \frac{c}{M-\lambda W} \iff \xi < \xi_1 := 2^{-c/(M-\lambda W)}.$$

Hence $\mathcal{L}_\lambda(\pi_\xi) > \mathcal{L}_\lambda(f)$ for every $0 < \xi < \xi_1$: mixing in the detour with small enough probability increases the objective.

When the condition can hold. Since $M \leq W$, the condition $M > \lambda W$ requires $\lambda < 1$. Conversely, if the detour differs from the original route only at states that f never visits, then $M = W > 0$, and the condition holds for every $0 \leq \lambda < 1$. □

B.10 Proof of the feasibility floor (Proposition 8, Section 5) and the rooms and cycle floor

The bound. Fix $\pi \in \Pi_K$. We bound each term $H(d_t^\pi) = H(S_t)$ of $U(\pi)$ from below.

Time 0. $S_0 \sim \rho$ for every policy, so $H(S_0) = H(\rho)$.

Times $1 \leq t \leq K-1$. Conditioning never increases entropy (F2), so

$$H(S_t) \;\geq\; H(S_t \mid S_{t-1}, A_{t-1}).$$

By the definition of conditional entropy, the right side averages $H(S_t \mid S_{t-1} = s, A_{t-1} = a)$ over the pairs (s,a), with weights $\mathbb{P}_\pi(S_{t-1} = s, A_{t-1} = a) = \mu_{t-1}^\pi(s,a)$. Given $S_{t-1} = s$ and $A_{t-1} = a$, the next state has law $P(\cdot \mid s,a)$ by (15). So

$$H(S_t \mid S_{t-1}, A_{t-1}) = \sum_{s,a} \mu_{t-1}^\pi(s,a)\, H\big(P(\cdot \mid s,a)\big).$$

Every action a available at s has $H(P(\cdot \mid s,a)) \geq h(s) = \min_{a'} H(P(\cdot \mid s,a'))$. Replacing each entropy by $h(s)$ and using $\sum_a \mu_{t-1}^\pi(s,a) = d_{t-1}^\pi(s)$,

$$H(S_t) \;\geq\; \sum_{s,a} \mu_{t-1}^\pi(s,a)\, h(s) = \sum_s d_{t-1}^\pi(s)\, h(s) = \mathbb{E}_\pi\big[h(S_{t-1})\big].$$

In words: whatever the policy does, the state at time t is at least as uncertain as the least noisy action from the previous state allows.

Summing. Multiply the bound at each time by w_t and add:

$$U(\pi) = \sum_{t=0}^{K-1} w_t H(S_t) \;\geq\; J_h(\pi) := w_0 H(\rho) + \sum_{t=1}^{K-1} w_t\, \mathbb{E}_\pi[h(S_{t-1})].$$

which is (14). The right side still depends on π. To get a single number that bounds U for every policy, take the smallest value of J_h:

$$U(\pi) \;\geq\; J_h(\pi) \;\geq\; \min_{\pi'\in\Pi_K} J_h(\pi') \qquad \text{for every } \pi \in \Pi_K,$$

Computing the minimum. Substituting $u = t - 1$, the policy-dependent part of J_h is

$$\mathbb{E}_\pi\Big[\sum_{u=0}^{K-2} c_u(S_u)\Big], \qquad c_u(s) = w_{u+1}h(s).$$

This is the expected total cost of an ordinary finite-horizon MDP problem: at each time u the agent pays $c_u(S_u)$, which depends only on the current state, and its actions affect only which states it visits later. Such problems are solved by backward induction. The action at time $K-2$ affects only S_{K-1}, which carries no cost, so the last stage is $V_{K-2}(s) = c_{K-2}(s)$. For $u = K-3, \dots, 0$, the cost-to-go is the cost now plus the smallest expected cost-to-go over the next action:

$$V_u(s) = c_u(s) + \min_{a \in \mathcal{A}(s)} \sum_{s'} P(s' \mid s, a)\, V_{u+1}(s').$$

By standard dynamic programming for finite-horizon MDPs (Puterman, 1994, Ch. 4),

$$\min_{\pi' \in \Pi_K} J_h(\pi') = w_0 H(\rho) + \sum_s \rho(s) V_0(s).$$

The minimum is attained by the deterministic policy that picks a minimizing action in each backup.

Cost. Computing h takes one entropy per state-action pair. Each of the $K-1$ backups then examines every state s, every action $a \in \mathcal{A}(s)$, and every next state s', for $O(K\, n^2 \max_s |\mathcal{A}(s)|)$ arithmetic operations in total.

What this policy is. The minimizing policy minimizes the lower bound J_h, not necessarily U itself. Its value is a certificate: no policy can have U below it, so every budget ε below this value is infeasible.

Noisy rooms: the bound is exact. States are the hub 0 and rooms $1, \dots, R$, with $\rho = \delta_0$. From the hub, action a leads to room a with probability $1 - p + p/R$ and to each other room with probability p/R; from any room, every action returns to the hub. Every hub action has the same next-state entropy $\eta = H(1 - p + \frac{p}{R}, \frac{p}{R}, \dots, \frac{p}{R})$, so $h(0) = \eta$, and $h(j) = 0$ for rooms. Under every policy, S_u is the hub for even u and a room for odd u. Hence $\mathbb{E}_\pi[h(S_{t-1})] = \eta\, \mathbf{1}\{t \text{ odd}\}$ for all π, and the bound reads $U(\pi) \geq \eta \sum_{t \text{ odd}} w_t$. Any deterministic policy attains it: at odd t its law d_t is the next-state law of one hub action, with entropy η, and at even t it is δ_0. This proves (34); for $R = 8$, $p = 0.15$, $K = 33$, $\gamma = 0.97$ it evaluates to 0.449415 bits.

Noisy cycle. States are $\{0, \dots, 15\}$, $\rho = \delta_0$, actions advance ($s \mapsto s + 1 \bmod 16$) or wait, and with probability p the chosen action is replaced by a uniformly random one. Each state-action pair therefore reaches its intended successor with probability $1 - p/2$ and the other one with probability $p/2$, and the two successors differ. So $H(P(\cdot \mid s, a)) = h_2(p/2)$ for all (s, a), $h \equiv h_2(p/2)$, and $J_h(\pi) = (1 - w_0)\, h_2(p/2)$ for every policy. For $p = 0.10$, $K = 33$, $\gamma = 0.97$: $h_2(0.05) = 0.2864$ bits and $w_0 = 0.0473$, giving $U(\pi) \geq 0.2728$ bits. Unlike the rooms task, this bound need not be attained, since position errors accumulate.

C Guarantees for Certainty PG (Algorithm 1, Section 4)

This section states what can be proved about the optimization and the estimate that drives the multiplier. Neither result covers the full sampled algorithm, which additionally uses estimated densities in the score, Adam, and an adaptive multiplier.

C.1 Convergence to stationary points with exact densities

Let $\theta \in \mathbb{R}^p$ collect the tabular logits $\theta_{t,s,a}$, with $\pi_\theta(a \mid s, t) \propto \exp(\theta_{t,s,a})$, and let $\Theta_c = [-c, c]^p$ be the box to which the implementation clips logits. Fix $\lambda \geq 0$ and write $\mathcal{L}(\theta) = C(\theta) - \lambda U(\theta)$.

Proposition 17 (Smoothness and projected ascent). *$\mathcal{L}$ is infinitely differentiable on $\mathbb{R}^p$, and $\nabla\mathcal{L}$ is Lipschitz on Θ_c with some constant $\ell < \infty$. Let $\theta_0 \in \Theta_c$ and $\theta_{j+1} = \Pi_{\Theta_c}\big(\theta_j + \eta \nabla\mathcal{L}(\theta_j)\big)$ with $0 < \eta \leq 1/\ell$, and $G_\eta(\theta) = \big(\Pi_{\Theta_c}(\theta + \eta\nabla\mathcal{L}(\theta)) - \theta\big)/\eta$. Then $\mathcal{L}(\theta_j)$ is nondecreasing, the right side below is at most $2(1+\lambda)\log_2 n/(\eta J)$,*

$$\min_{0 \leq j < J} \|G_\eta(\theta_j)\|^2 \;\leq\; \frac{2\big(\mathcal{L}^\star - \mathcal{L}(\theta_0)\big)}{\eta J}, \qquad \mathcal{L}^\star = \max_{\Theta_c} \mathcal{L},$$

and every limit point $\bar{\theta}$ satisfies $G_\eta(\bar{\theta}) = 0$, i.e. it is a first-order stationary point of $\mathcal{L}$ over Θ_c.

Proof. Smoothness. Softmax probabilities are infinitely differentiable and strictly positive on all of $\mathbb{R}^p$. By Step 1 of Section B.5, the support $\mathcal{S}_t$ of d_t^θ does not depend on θ, and for $s \in \mathcal{S}_t$, $d_t^\theta(s)$ is a finite sum of products of policy probabilities, transition probabilities, and ρ, hence infinitely differentiable and strictly positive. The same holds for $\bar{d}^\theta$ on its fixed support. Since $x \mapsto x \log_2 x$ is infinitely differentiable on $(0, \infty)$, C and U, and hence $\mathcal{L}$, are infinitely differentiable on $\mathbb{R}^p$. The Hessian is continuous, so its operator norm is bounded on the compact convex set Θ_c by some ℓ, and the mean value theorem gives $\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta')\| \le \ell\|\theta - \theta'\|$ there.

Sufficient increase. For ℓ-smooth $\mathcal{L}$ and $\theta, \theta^+ \in \Theta_c$, Taylor's theorem gives $\mathcal{L}(\theta^+) \ge \mathcal{L}(\theta) + \langle\nabla\mathcal{L}(\theta), \theta^+ - \theta\rangle - \frac{\ell}{2}\|\theta^+ - \theta\|^2$. Take $\theta^+ = \Pi_{\Theta_c}(\theta + \eta\nabla\mathcal{L}(\theta))$. The projection onto a closed convex set satisfies $\langle\theta + \eta\nabla\mathcal{L}(\theta) - \theta^+,\, \theta - \theta^+\rangle \le 0$, which rearranges to $\langle\nabla\mathcal{L}(\theta), \theta^+ - \theta\rangle \ge \frac{1}{\eta}\|\theta^+ - \theta\|^2$. Therefore

$$\mathcal{L}(\theta^+) - \mathcal{L}(\theta) \;\ge\; \Big(\frac{1}{\eta} - \frac{\ell}{2}\Big)\|\theta^+ - \theta\|^2 \;\ge\; \frac{\eta}{2}\|G_\eta(\theta)\|^2,$$

using $\eta \le 1/\ell$ and $\theta^+ - \theta = \eta G_\eta(\theta)$.

Rate and limit points. Summing over $j < J$ and using $\mathcal{L}(\theta_J) \le \mathcal{L}^\star$ gives $\frac{\eta}{2}\sum_{j<J}\|G_\eta(\theta_j)\|^2 \le \mathcal{L}^\star - \mathcal{L}(\theta_0)$, which implies the rate. Since $0 \le C, U \le \log_2 n$ and $\lambda \ge 0$, $\mathcal{L}^\star \le \log_2 n$ and $\mathcal{L}(\theta_0) \ge -\lambda \log_2 n$, which gives the stated numerical bound. The series converges, so $G_\eta(\theta_j) \to 0$; G_η is continuous, so every limit point has $G_\eta(\bar{\theta}) = 0$. Finally, $G_\eta(\bar{\theta}) = 0$ means $\bar{\theta}$ is a fixed point of the projected step, which is equivalent to $\langle\nabla\mathcal{L}(\bar{\theta}), \theta - \bar{\theta}\rangle \le 0$ for all $\theta \in \Theta_c$ by the variational characterization of the projection. □

What this does and does not say. The exact gradient is available only with the true state laws. With exact densities, the sampled estimator of Theorem 4 is unbiased with bounded variance (scores are bounded on Θ_c), and standard results for projected stochastic gradient methods on smooth nonconvex problems apply. With plug-in densities, as in Algorithm 1, the score is biased; Proposition 18 quantifies the bias of the constraint estimate, not the bias of the actor gradient. Moreover, Proposition 7 exhibits suboptimal local maxima at deterministic state-marginal sequences when $\lambda > 1$. Those boundary policies are not represented exactly by finite softmax logits. The proposition does not locate stationary points inside Θ_c; the projected-ascent result makes no claim of global optimality or constraint satisfaction.

C.2 Bias of the plug-in estimate of U used by the multiplier

Let $\tau^1, \ldots, \tau^B$ be independent runs of a fixed policy π, $\hat{d}_t = N_t/B$ the empirical law at time t, and $\widehat{U}_B = \sum_t w_t H(\hat{d}_t)$, the quantity used by the multiplier in Algorithm 1. Let $n_t = |\mathrm{supp}(d_t)|$; for a policy with full action support, n_t is the number of states reachable at time t.

Proposition 18 (Bias and a high-probability bound)**.** *(a)* $\mathbb{E}\,\widehat{U}_B \;\le\; U \;\le\; \mathbb{E}\,\widehat{U}_B + \sum_t w_t \log_2\big(1 + \frac{n_t - 1}{B}\big)$.

(b) For $B \ge 3$ and $\delta \in (0,1)$, with probability at least $1 - \delta$,

$$U \;\le\; \widehat{U}_B + \sum_t w_t \log_2\Big(1 + \frac{n_t - 1}{B}\Big) + \log_2 B\sqrt{\frac{2\ln(1/\delta)}{B}}.$$

Proof. (a) Fix t and write $d = d_t$, $\hat{d} = \hat{d}_t$. Almost surely $\mathrm{supp}(\hat{d}) \subseteq \mathrm{supp}(d)$, and on this support

$$H(\hat{d}) = -\sum_s \hat{d}(s)\log_2 d(s) - \mathrm{KL}(\hat{d}\|d).$$

Since $\mathbb{E}\hat{d}(s) = d(s)$, taking expectations gives $\mathbb{E}H(\hat{d}) = H(d) - \mathbb{E}\,\mathrm{KL}(\hat{d}\|d)$, so $\mathbb{E}H(\hat{d}) \le H(d)$. For the other direction, Jensen's inequality for the concave logarithm under $s \sim \hat{d}$ gives $\mathrm{KL}(\hat{d}\|d) =$

$\mathbb{E}_{s\sim\hat{d}} \log_2 \frac{\hat{d}(s)}{d(s)} \le \log_2 \sum_s \frac{\hat{d}(s)^2}{d(s)} = \log_2\big(1+\chi^2(\hat{d}\|d)\big)$, and applying Jensen again over the sample, $\mathbb{E}\,\mathrm{KL}(\hat{d}\|d) \le \log_2\big(1+\mathbb{E}\chi^2(\hat{d}\|d)\big)$. Each $B\hat{d}(s)$ is binomial with parameters $(B, d(s))$, so

$$\mathbb{E}\chi^2(\hat{d}\|d) = \sum_{s\in\mathrm{supp}(d)} \frac{\mathrm{Var}\,\hat{d}(s)}{d(s)} = \sum_{s\in\mathrm{supp}(d)} \frac{1-d(s)}{B} = \frac{n_t-1}{B}.$$

Multiplying by w_t and summing over t proves (a).

(b) $\widehat{U}_B$ is a function of the B independent runs. Replacing one run changes, at each time t, at most one count $N_t(s)$ down by one and one count $N_t(s')$ up by one. With $\varphi(x) = -x\log_2 x$ and $x \in \{0, \frac{1}{B}, \dots, \frac{B-1}{B}\}$, we claim $|\varphi(x+\frac{1}{B}) - \varphi(x)| \le \frac{\log_2 B}{B}$ for $B \ge 3$. For $x = 0$ the difference is $\varphi(1/B) = \frac{\log_2 B}{B}$. For $x \ge \frac{1}{B}$, the mean value theorem gives a point $\xi \in [\frac{1}{B}, 1]$ with $|\varphi'(\xi)| = |\log_2(1/\xi) - \log_2 e| \le \max\{\log_2 e, \log_2 B - \log_2 e\} \le \log_2 B$, the last step because $\log_2 B \ge \log_2 3 > \log_2 e$. Hence $H(\hat{d}_t)$ changes by at most $2\log_2 B/B$ and $\widehat{U}_B$ by at most $c = 2\log_2 B/B$. McDiarmid's inequality (McDiarmid, 1989) gives $\mathbb{P}\big(\mathbb{E}\widehat{U}_B - \widehat{U}_B \ge u\big) \le \exp\big(-2u^2/(Bc^2)\big) = \exp\big(-u^2 B/(2\log_2^2 B)\big)$. Setting the right side to δ and combining with (a) proves (b). □

Size of the terms. On Rooms-8 with the main settings, $n_t = 1$ at even times and $n_t = 8$ at odd times, so the bias allowance in (a) is $\big(\sum_{t\ \mathrm{odd}} w_t\big)\log_2(1+7/B)$: 0.138, 0.072, 0.037 and 0.019 bits for $B = 32$, 64, 128 and 256. These match the feasibility pattern of the batch sweep (0/10, 5/10, 10/10, 10/10 at $\varepsilon = 0.1$). The deviation term in (b) is conservative: it is about 1.2 bits at $B = 256$ and $\delta = 0.05$, because bounded differences ignore that near-deterministic laws have small variance. A variance-adaptive certificate for small entropies would be a useful refinement.

D Infinite-Horizon Version of the Objectives in Section 3

The normalized discounted snapshot entropy and pooled law are

$$\mathcal{H}_\gamma(\pi) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t H(d_t^\pi) \in [0, \log_2 n], \qquad \bar{d}_\gamma^\pi = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t d_t^\pi. \tag{23}$$

The infinite-horizon constrained problem replaces $(C_{\gamma,K}, U_{\gamma,K})$ by $(H(\bar{d}_\gamma), \mathcal{H}_\gamma)$; the experiments use the finite target. Each infinite-horizon quantity is a mixture of its normalized prefix (mass $1-\gamma^K$) and normalized tail (mass γ^K), so for any continuation of a K-step policy

$$|\mathcal{H}_\gamma(\pi) - U_{\gamma,K}(\pi)| \le \gamma^K \log_2 n, \qquad d_{\mathrm{TV}}(\bar{d}_\gamma^\pi, \bar{d}_{\gamma,K}^\pi) \le \gamma^K. \tag{24}$$

Finite-horizon feasibility therefore need not imply infinite-horizon feasibility at the same threshold. At the main experimental setting, $\gamma^K \simeq 0.366$.

An exact two-state example. Start from $d_0 = \delta_0$. Process A alternates deterministically between states 0 and 1; its snapshot entropy is zero and its discounted visitation is $(1/(1+\gamma), \gamma/(1+\gamma))$. Process B moves at time one to $q = (\gamma/(1+\gamma), 1/(1+\gamma))$ and stays at this law. It has the same discounted visitation as A, but snapshot entropy $\gamma h(1/(1+\gamma)) > 0$, where h is binary entropy. At $\gamma = 0.97$ both have pooled entropy 0.999833 bits, while snapshot entropy is 0 and 0.969838 bits.

E Minimizing Snapshot Entropy by Dynamic Programming over State Distributions

Controlled distribution dynamics. A *statewise control* is a tuple $u = (u(\cdot \mid s))_{s\in\mathcal{S}}$ with $u(\cdot \mid s) \in \Delta(\mathcal{A}(s))$; the compact set of such controls is $\mathcal{U}$. A control is *pure* if it selects one action at every state; the finite set of pure controls is $\mathcal{U}_{\mathrm{det}}$, with $|\mathcal{U}_{\mathrm{det}}| = \prod_s |\mathcal{A}(s)|$. For $d \in \Delta(\mathcal{S})$ and $u \in \mathcal{U}$,

$$F(d,u)(s') = \sum_{s\in\mathcal{S}} d(s) \sum_{a\in\mathcal{A}(s)} u(a \mid s) P(s' \mid s, a) \tag{25}$$

is the law of the next state when each independent execution samples its action from $u(\cdot \mid S_t)$. A *lifted policy* chooses controls from the distribution history; the recursion $d_{t+1} = F(d_t, u_t)$ is then deterministic. A stationary lifted policy is a map $\alpha : \Delta(\mathcal{S}) \to \mathcal{U}$. From a fixed ρ, the induced control sequence can be executed as an ordinary time-dependent policy without observing d_t online. Randomness, including any shared random choice of a policy, is averaged into the marginal before entropy is evaluated.

The state d_t is sufficient for snapshot-only planning but not for the constrained coverage objective: two histories can reach the same d_t after visiting different states, so coverage planning would also need the accumulated mass $\sum_{j<t} w_j d_j$ and the remaining budget.

Bellman operators. Let $V_{\min}(d) = \inf_\pi \mathcal{H}_\gamma(\pi; d)$ and $V_{\max}(d) = \sup_\pi \mathcal{H}_\gamma(\pi; d)$. For f in the continuous functions $C(\Delta(\mathcal{S}))$ with the supremum norm, define

$$(\mathcal{T}_{\min} f)(d) = (1-\gamma)H(d) + \gamma \min_{u \in \mathcal{U}} f(F(d,u)), \tag{26}$$

$$(\mathcal{T}_{\max} f)(d) = (1-\gamma)H(d) + \gamma \max_{u \in \mathcal{U}} f(F(d,u)). \tag{27}$$

Theorem 19 (Bellman characterization)**.** *Both operators map $C(\Delta(\mathcal{S}))$ to itself and are γ-contractions. Their unique fixed points are $V_{\min}$ and $V_{\max}$, and each objective has an optimal stationary lifted policy. If $W_0 \equiv 0$ and $W_{K+1} = \mathcal{T}_\circ W_K$, then W_K is the K-stage value and*

$$0 \le V_\circ(d) - W_K(d) \le \gamma^K \log_2 n \quad \textit{for all } d \in \Delta(\mathcal{S}). \tag{28}$$

For the finite policy class, the smallest normalized uncertainty is $W_K^{\min}(\rho)/(1-\gamma^K)$, since the Bellman iterate uses unnormalized prefix weights. This decides whether the finite constraint is feasible before any parameterization is imposed. To make truncation error at most ϵ it suffices to take $K \ge \max\{1, \lceil \log(\epsilon/\log_2 n)/\log\gamma \rceil\}$.

Pure minimizers on the simplex. Proposition 5(b) shows that deterministic policies minimize the finite-horizon objective. The lifted formulation gives the same conclusion for the infinite horizon.

Theorem 20 (Pure minimizers, infinite horizon)**.** *Every finite-horizon minimizing value W_K is concave on $\Delta(\mathcal{S})$, and so is $V_{\min}$. At every backup,*

$$\min_{u \in \mathcal{U}} W_K(F(d,u)) = \min_{a \in \mathcal{U}_{\det}} W_K(F(d,a)), \tag{29}$$

and the infinite-horizon value admits an optimal stationary lifted policy that selects a pure profile at every distribution.

Because the profile may change with the distribution, the resulting policy on the original MDP can be time-dependent. Even a pure policy can have positive snapshot entropy when transitions are stochastic.

Maximization can require randomization. Consider a state s with two actions that move deterministically to absorbing states q_0 and q_1. From δ_s, every pure control has snapshot value zero. Choosing each action with probability $1/2$ creates the uniform law on $\{q_0, q_1\}$ from time one onward and obtains value γ. The pure-control result is therefore specific to minimization.

F Grid Value Iteration: A Reference Solver for Section E

The exact Bellman state has $n-1$ free coordinates, and tabulating values per original state is insufficient because the entropy of a mixture is not the mixture of entropies. Algorithm 2 replaces the simplex by a mesh, propagates each grid law through the known model, and interpolates continuation values. For integer $N \ge 1$, let

$$G_N = \{d \in \Delta(\mathcal{S}) : Nd(s) \in \mathbb{N}_0 \text{ for every } s\}, \qquad |G_N| = \binom{N+n-1}{n-1}, \tag{30}$$

and let $\mathcal{I}_N : \mathbb{R}^{G_N} \to C(\Delta(\mathcal{S}))$ be barycentric interpolation on a triangulation, which agrees with grid values and is nonexpansive. For $v : G_N \to \mathbb{R}$, define

$$(\widehat{\mathcal{T}}_{\circ,N} v)(d) = (1-\gamma)H(d) + \gamma \operatorname*{ext}_{u \in \mathcal{U}} (\mathcal{I}_N v)(F(d,u)), \qquad d \in G_N, \tag{31}$$

with $\mathrm{ext} = \min$ or $\max$.

Algorithm 2 Simplex-grid value iteration

Require: Model P, discount γ, grid resolution N, iterations K, mode $\circ \in \{\min, \max\}$
1: Construct G_N and a barycentric interpolator $\mathcal{I}_N$
2: $v_0(d) \leftarrow 0$ for every $d \in G_N$
3: **for** $k = 0, \dots, K-1$ **do**
4: **for** each $d \in G_N$ **do**
5: $v_{k+1}(d) \leftarrow (1-\gamma)H(d) + \gamma\, \mathrm{ext}_{u\in\mathcal{U}}(\mathcal{I}_N v_k)(F(d,u))$
6: **end for**
7: **end for**
8: **return** $\mathcal{I}_N v_K$ and the final greedy selector

Both extrema are compact continuous optimization problems. An interpolated iterate need not remain concave, so the certified implementation keeps the full control space rather than restricting to pure profiles. Let $\epsilon_N(V) = \|V - \mathcal{I}_N(V|_{G_N})\|_\infty$.

Theorem 21 (Grid approximation)**.** *Let $\widehat{V}_{\circ,N}$ be the fixed point of the grid backup. Then*

$$\|\mathcal{I}_N \widehat{V}_{\circ,N} - V_\circ\|_\infty \le \frac{\epsilon_N(V_\circ)}{1-\gamma}, \qquad \|\mathcal{I}_N v_K - V_\circ\|_\infty \le \gamma^K \log_2 n + \frac{\epsilon_N(V_\circ)}{1-\gamma}. \tag{32}$$

If every interpolation cell containing d has vertices within total variation $r_N \le 1 - 1/n$ of d, then $\epsilon_N(V_\circ) \le r_N \log_2(n-1) + h_2(r_N)$; otherwise $\epsilon_N(V_\circ) \le \log_2 n$.

More iterations reduce the first term; only a finer mesh reduces the second. The grid has $\binom{N+n-1}{n-1}$ points, so this method is a reference solver for tiny MDPs, not a scalable algorithm. The certificate assumes each backup is solved globally. At the resolutions used in our tiny-state check, the analytic bound exceeds the range of the value function (Section I.7), so in practice we use the solver as a reference whose empirical error we measure, not as a certificate.

G Proofs for Sections E and F and Hardness of Snapshot Minimization

G.1 Proof of the Bellman characterization (Theorem 19)

$\Delta(\mathcal{S})$ and $\mathcal{U}$ are compact and $(d,u) \mapsto F(d,u)$ is continuous, so for $f \in C(\Delta(\mathcal{S}))$ the extrema in (26) and (27) are attained and, by the maximum theorem, vary continuously with d. Let $\delta = \|f - g\|_\infty$. For every d, u, $|f(F(d,u)) - g(F(d,u))| \le \delta$; taking a minimum or maximum preserves this bound and the entropy terms cancel, so $\|\mathcal{T}_\circ f - \mathcal{T}_\circ g\|_\infty \le \gamma\delta$. Banach's fixed-point theorem gives a unique fixed point.

For $K+1$ rewards, peel off the current reward; the remaining K begin at $F(d,u)$ with one more factor γ, so backward induction gives $W_{K+1} = \mathcal{T}_\circ W_K$. Compactness gives a measurable extremizing selector at every step, and a measurable selection theorem supplies a Borel stationary selector for the fixed point. Iterating its Bellman equality leaves a terminal term at most $\gamma^K \log_2 n$, which vanishes; for any other control sequence, the Bellman inequality gives the one-sided bound, proving optimality. Finally every trajectory has tail $(1-\gamma)\sum_{t\ge K} \gamma^t H(d_t) \le \gamma^K \log_2 n$, uniformly over policies, which proves (28).

G.2 Proof that pure controls minimize (Theorem 20)

We show by induction that the minimizing iterates are concave. $W_0 = 0$ is concave. If W_K is concave, then for fixed u the map $d \mapsto W_K(F(d,u))$ is concave because F is linear in d; a pointwise infimum of concave functions is concave, and adding $(1-\gamma)H(d)$ preserves concavity.

Fix d and a mixed control u. Sampling one action independently at every state induces weights $\lambda_a = \prod_s u(a(s) \mid s)$ over $a \in \mathcal{U}_{\mathrm{det}}$, which sum to one, and $F(d,u) = \sum_a \lambda_a F(d,a)$. By concavity,

$$W_K(F(d,u)) \geq \sum_a \lambda_a W_K(F(d,a)) \geq \min_{a \in \mathcal{U}_{\mathrm{det}}} W_K(F(d,a)).$$

The reverse inequality holds because $\mathcal{U}_{\mathrm{det}} \subset \mathcal{U}$, proving (29). Convergence in (28) is uniform, so $V_{\min}$ is concave and the same argument applies to its backup. Since $\mathcal{U}_{\mathrm{det}}$ is finite, ties broken by a fixed order give a Borel stationary selector. From a fixed ρ, the selected profiles form a time-indexed sequence of deterministic actions, which is the statement of Theorem 20.

G.3 Proof of the grid error bound (Theorem 21)

Define $\widetilde{\mathcal{T}}_{\circ,N} f = \mathcal{I}_N((\mathcal{T}_\circ f)|_{G_N})$. Nonexpansiveness of $\mathcal{I}_N$ and contraction of $\mathcal{T}_\circ$ make it a γ-contraction with fixed point $\mathcal{I}_N \widehat{V}_{\circ,N}$. Since $\mathcal{T}_\circ V_\circ = V_\circ$,

$$\|\mathcal{I}_N \widehat{V}_{\circ,N} - V_\circ\|_\infty \leq \|\widetilde{\mathcal{T}}_{\circ,N}(\mathcal{I}_N \widehat{V}_{\circ,N}) - \widetilde{\mathcal{T}}_{\circ,N} V_\circ\|_\infty + \|\mathcal{I}_N(V_\circ|_{G_N}) - V_\circ\|_\infty \leq \gamma \|\mathcal{I}_N \widehat{V}_{\circ,N} - V_\circ\|_\infty + \epsilon_N(V_\circ).$$

Rearranging gives the first bound; iterating from zero adds at most $\gamma^K \log_2 n$.

For the modulus, let $\Omega_n(r) = r \log_2(n-1) + h_2(r)$ for $0 \leq r \leq 1 - 1/n$ and $\Omega_n(r) = \log_2 n$ above. Finite-alphabet continuity gives $|H(d) - H(d')| \leq \Omega_n(d_{\mathrm{TV}}(d,d'))$, and every stochastic matrix is nonexpansive in total variation. Using the same control at two distributions and $|\operatorname{ext}_u f_u - \operatorname{ext}_u g_u| \leq \sup_u |f_u - g_u|$, induction gives $|W_K(d) - W_K(d')| \leq (1-\gamma^K)\Omega_n(d_{\mathrm{TV}}(d,d'))$, and in the limit $|V_\circ(d) - V_\circ(d')| \leq \Omega_n(d_{\mathrm{TV}}(d,d'))$. Barycentric interpolation is a convex combination of vertex values, so vertices within r_N give the stated bound by monotonicity of Ω_n.

G.4 Snapshot minimization alone is hard to approximate (reduction from minimum-entropy set cover)

Theorem 22 (Snapshot minimization is hard to approximate)**.** *Fix a rational $\gamma \in (0,1)$ and an explicit rational encoding of the MDP and ρ. For every fixed $0 < \eta < \gamma \log_2 e$, finding a deterministic memoryless policy with $\mathcal{H}_\gamma$ at most $V_{\min}(\rho) + \eta$ is NP-hard, already on deterministic MDPs whose unrestricted optimum is attained by such a policy. For every fixed rational $\delta > 0$, the promise problem separating $V_{\min}(\rho) \leq b$ from $V_{\min}(\rho) \geq b + \delta$ has polynomially checkable certificates.*

The two parts of Theorem 22 concern different tasks: verifying a promised threshold and producing a near-optimal policy. Hardness of the second alone does not establish NP-completeness of the first.

Membership. For fixed rational $\delta > 0$, choose $K \geq 1$ with $\gamma^K \lceil \log_2 n \rceil \leq \delta/4$; for fixed γ, δ, $K = O(1 + \log\log n)$, so a certificate of $K-1$ pure profiles is polynomial. Propagating rational laws for K steps uses polynomial bit length, and each nonzero entropy term can be evaluated so the total weighted error is at most $\delta/8$. If $V_{\min}(\rho) \leq b$, an optimal pure prefix has numerical value at most $b + \delta/8$. If $V_{\min}(\rho) \geq b + \delta$, every pure prefix costs at least $b + 3\delta/4$ by the tail bound, so its numerical value is at least $b + 5\delta/8$. Accepting at $b + \delta/2$ separates the promised cases.

Source problem. Minimum entropy set cover takes a universe E of size m and sets $C_1, \ldots, C_\ell$ covering E. An assignment $f : E \to \{1, \ldots, \ell\}$ with $e \in C_{f(e)}$ costs $H(p_f)$, where $p_f(j) = |f^{-1}(j)|/m$. By Cardinal et al. (2008) (Theorem 1 of Cardinal et al. (2012)), for every fixed $\zeta > 0$ with $\log_2 e - \zeta > 0$, producing an assignment within $\log_2 e - \zeta$ bits of optimal is NP-hard.

Reduction. Create one state per element and one absorbing state per set. Put initial mass $1/m$ on each element state. The actions at element e route deterministically to the set states containing e; all later transitions are self-loops. A pure policy is exactly an assignment f, and

$$\mathcal{H}_\gamma(f) = (1-\gamma)\log_2 m + \gamma H(p_f), \qquad V_{\min}(\rho) = (1-\gamma)\log_2 m + \gamma\, \mathrm{OPT}_{\mathrm{MESC}}\,. \tag{33}$$

Concavity ensures mixed assignments cannot improve the minimum. A polynomial algorithm returning a policy with $\mathcal{H}_\gamma(f) \leq V_{\min}(\rho) + \eta$ would give $H(p_f) \leq \mathrm{OPT}_{\mathrm{MESC}} + \eta/\gamma$; for fixed $\eta < \gamma \log_2 e$ this contradicts the cited theorem with $\zeta = \log_2 e - \eta/\gamma$, unless P = NP. The constant $\log_2 m$ cancels in the gap, so it need not be encoded.

Table 3: Comparison methods, training signals, and roles. Learned methods share the sampled actor-update procedure; the stationary ablation also shares logits across time. Per-state scores are weighted by w_t and converted to future returns. The trajectory-entropy follow-up instead uses a whole-run return.

Name	Objective / per-state score	Purpose
Random	Uniform independent actions	Uninformed reference with maximal action entropy; not a SAC implementation.
Pooled PG	Maximize C; $-\log_2 \widehat{\bar{d}}(S_t)$	Coverage-only comparison with the same time-conditioned actor as Certainty PG.
Trajectory-entropy PG	Maximize J; return $H(m_\tau)$ for every action	Direct within-run coverage without a snapshot budget; follow-up check.
Count PG	$1/\sqrt{1+N_{\mathrm{past}}(S_t)}$	Visitation novelty; counts from previous batches, frozen during an update.
Snapshot-min PG	Minimize U; $\log_2 \widehat{d}_t(S_t)$	Concentration only, coverage incentive removed.
Certainty PG	Maximize C s.t. $U \le \varepsilon$; $-\log_2 \widehat{\bar{d}}(S_t) + \lambda \log_2 \widehat{d}_t(S_t)$	Joint coverage and repeatability with a learned multiplier.
Stationary	Same objective; $\pi(a \mid s)$	Time-conditioning ablation.
Scripted tour	Fixed cyclic room order, or always advance	Hand-specified structural reference; not a learned result or optimality bound.

H Implementation of Certainty PG and the Comparison Methods (Sections 4 and 6)

Actor. The tabular actor uses independent logits per time, state, and action, $\pi_\theta(a \mid s,t) \propto \exp(\theta_{t,s,a})$. Finite logits give every allowed action positive probability during training.

Optimization and estimation. The actor step is Adam ascent with gradient clipping and bounded logits; the multiplier cap is $\lambda_{\max} = 20$. The baseline is a per-time exponential moving average of previous batches' returns, updated after the current gradient is formed, so it affects variance but not the zero-mean identity. The multiplier uses the *unsmoothed* plug-in entropy of the batch counts, separately from the smoothed leave-one-out scores. By concavity this plug-in estimate is biased downward on average, so a small observed violation can hide a larger true one; neither smoothing nor leave-one-out removal gives a finite-sample feasibility certificate. The sampled algorithm additionally faces density-estimation error, stochastic credit assignment, and nonconvex constrained optimization, which is why all experiments report exact post-training violations.

Objective variants. Removing the snapshot penalty gives Pooled PG; keeping only negative snapshot surprise gives Snapshot-min PG; reversing its sign gives Snapshot-max PG, used only on the tiny absorbing branch. All reuse the same actor, rollouts, and score-function backbone. Table 3 summarizes them.

Pooled and count objectives follow maximum state entropy and count-based exploration (Hazan et al., 2019; Bellemare et al., 2016); they are objective-level comparisons in a shared tabular optimizer, not reproductions of Hazan et al.'s planner, RE3, random network distillation, or soft actor-critic. Snapshot-min isolates the concentration objective without the full SMiRL implementation.

I Tabular Experiments: Details and Complete Results (Section 6)

I.1 Benchmarks, seeds, and reporting

The original tabular experiment deck contains 770 seed jobs in 77 benchmark–method cells, ten seeds per cell. This count includes repeated settings shared between sweeps, so individual problem instances can appear in several cells. It comprises 280 main-suite jobs, 60 constraint-sweep jobs, 150 noise-sweep jobs, 50 batch-sweep jobs, 90 size-sweep jobs, 120 stress-test jobs, and 20 tiny-reference jobs. Calibration runs use seeds 0–9 and are kept separate from the final artifacts. Final runs use

Table 4: Executed benchmark families. All training is reward-free except for the objective-specific intrinsic signals. A horizon counts states, including the reset state, so each rollout has $K - 1$ transitions.

Family	States	K	Dynamics and parameters
Scheduled rooms	$R + 1$	$4R + 1$	Hub action chooses a room, followed by a forced return. $R \in \{4, 8, 16\}$; replacement-room noise $p \in \{0, 0.05, 0.10, 0.15, 0.20, 0.30\}$ as specified by the suite.
Directed cycle	16	33	Advance or wait; intended action replaced by a uniform action with probability $p \in \{0, 0.10\}$.
Bidirectional ring	10	33	Clockwise, counterclockwise, or wait; deterministic transitions.
Open grid	25	65	5×5, four compass actions, blocked moves stay put; deterministic transitions.
Two-room grid	21	65	Central wall with a single central doorway; uniform action replacement with probability 0.10.
Absorbing branch	3	65	Two actions at state 0 lead to absorbing states 1 and 2; $\rho = (0.6, 0.2, 0.2)$, $\gamma = 0.9$.

seeds 100–109; evaluation sampling uses the independent seed offset 10,000,019. These original configurations were frozen before the held-out run. No task-specific hyperparameter search was conducted on the final seeds, and the harder bidirectional environment that failed during calibration is retained. The later follow-up checks in Section I.6 are kept separate and do not alter this deck or its source snapshots.

The controlled tabular design follows the diagnostic use of small MDPs in Mutti et al. (2022). The combination of coverage, exploration behavior, learning curves, and sensitivity studies is motivated by Seo et al. (2021); comparing concentration with uninformed and novelty baselines also follows the experimental logic of Berseth et al. (2021). These studies inform the experimental design; our tabular comparators use a shared optimizer and differ from the published deep-RL implementations. We report seed intervals separately for each task, since entropy values from differently sized state spaces are not directly comparable as an aggregate score. These intervals describe learning variability within a fixed MDP, rather than generalization across environments.

Scheduled rooms. Let 0 denote the hub and $1, \dots, R$ the rooms. From the hub,

$$P(j \mid 0, a) = (1 - p)\mathbf{1}\{j = a + 1\} + p/R, \qquad P(0 \mid j, a) = 1 \quad (j > 0).$$

Every action is legal everywhere; room actions have identical dynamics. The reset law is δ_0, and odd times are room visits. The scripted reference cycles through room indices at successive hub visits. This hand-specified policy uses structural knowledge and consumes no training samples. Under uniform actions and $K = 4R + 1$, each room's exact hit probability is $1 - (1 - 1/R)^{2R}$. The exact uncertainty floor is

$$U^{\text{floor}}_{\gamma,K} = \left(\sum_{\substack{t<K \\ t \text{ odd}}} w_t \right) H\Big(1 - p + \frac{p}{R}, \frac{p}{R}, \dots, \frac{p}{R}\Big). \tag{34}$$

Concavity of entropy makes a pure room selection minimal, and every hub return restores a point mass. Thus this floor is attainable by deterministic schedules and is independent of their coverage. It equals 0.449415 bits for $R = 8$, $p = 0.15$, $K = 33$, and $\gamma = 0.97$.

Cycles and gridworlds. The directed-cycle actions send s to $(s + 1) \bmod n$ or s. Noise is applied independently for every transition and replica. The bidirectional stress task additionally allows $(s - 1) \bmod n$. The open grid starts in its upper-left corner. The two-room grid removes the middle column except for its central doorway, leaving 21 accessible states. These purpose-built navigation problems use fully observed tabular states rather than MiniGrid visual inputs. The scripted cycle always advances, and its state distribution disperses under noise. It therefore provides a behavioral reference rather than an uncertainty optimum or an upper bound on learned coverage–certainty performance.

Table 5: Default Algorithm 1 settings. The same actor and optimizer are used for all PG baselines. Suite-specific exceptions are stated in the text and stored in each run.

Parameter	Value
Policy	Independent softmax logits per time, state, action
Initialization	IID normal logits, standard deviation 0.05
Discount / horizon	0.97 / 33 in the main suite
Synchronized replicas / updates	256 / 600
Actor optimizer	Adam, step size 0.05, $(\beta_1, \beta_2) = (0.9, 0.999)$
Adam denominator offset	10^{-8}
Logit clipping / gradient norm clipping	$[-6, 6]$ / 10
Baseline	Per-time EMA of past-batch returns, rate 0.2
Dual update	Step size 0.01, initialization 0, projection to $[0, 20]$
Snapshot constraint	0.1 bits, except target sweep
Density smoothing	0.5 pseudocount per state
Evaluation interval	Every 10 updates and the final update
Evaluation trajectories	4,096 (1,024 for the tiny-reference suite)
Bootstrap	5,000 seed resamples, percentile endpoints at 2.5/97.5%

I.2 Training settings and evaluation metrics

The tiny-reference runs use 300 updates rather than 600, with $B = 256$, $K = 65$, and $\gamma = 0.9$. The batch sweep uses the update counts specified below and logs every 80, 40, 20, 10, or 5 updates for $B = 32, 64, 128, 256, 512$, respectively. Learned tabular policies are evaluated at the final iterate, with no early stopping or checkpoint selection.

Score-function implementation. The implementation uses the future-state return

$$G_k^{\lambda}(\tau) = \sum_{t=k+1}^{K-1} w_t \left[-\log_2 \bar{d}_{\gamma,K}(S_t) + \lambda \log_2 d_t(S_t) \right], \tag{35}$$

with smoothed leave-one-trajectory-out density estimates. If $C_t(s)$ is the synchronized count and trajectory i has weighted occupancy $m_i(s) = \sum_t w_t \mathbf{1}\{S_t^i = s\}$, the density targets are

$$\widehat{d}_t^{-i}(S_t^i) = \frac{C_t(S_t^i) - 1 + \alpha}{B - 1 + \alpha n}, \qquad \widehat{\bar{d}}^{-i}(s) = \frac{\sum_t w_t C_t(s) - m_i(s) + \alpha}{B - 1 + \alpha n}.$$

Pooled leave-one-trajectory-out estimation removes the entire weighted contribution of trajectory i, including its visits at other times. The baseline uses previous batches only, avoiding a baseline fitted to the current trajectory's own return. The dual uses the unsmoothed plug-in snapshot entropy of the training batch, as in the algorithm. Exact marginals are available to the logger and evaluator but never enter actor, baseline, density, count, or dual updates. With estimated densities the training gradient has finite-batch bias and carries no finite-sample feasibility guarantee.

Comparators and ablations. Pooled PG maximizes pooled surprise, snapshot-min PG reverses the snapshot entropy gradient, and snapshot-max PG uses it directly on the tiny branch. Count PG uses $1/\sqrt{1 + N(s)}$, with counts accumulated from previous batches and frozen during a policy update. All use the same rollout budget, softmax initialization distribution, Adam settings, and baseline. These matched-optimizer comparisons isolate the objective using shared settings, without separately optimizing the tuning of each comparator. The stationary ablation shares logits across time and sums their score gradients; it has less representational capacity by design. Uniform random actions are also the maximal action-entropy policy in these fixed-action spaces. This fixed reference is distinct from a learned SAC policy.

Evaluation targets and uncertainty. Evaluation uses the normalized target $U_{\gamma,K}$; the Bellman iterate instead uses unnormalized prefix weights. At $\gamma = 0.97$, $K = 33$, the omitted geometric tail mass is about 0.366, limiting the interpretation of these values as infinite-horizon estimates. Goal discovery propagates a killed Markov chain for each candidate goal: probability mass is removed upon first hitting that goal. This gives exact per-goal probabilities and avoids the downward sampling

Table 6: Final-policy metrics in bits and worst-goal discovery (percent). Cells show mean [lower, upper] of a 95% seed-bootstrap interval. Entropy and goal-hit probabilities are evaluated exactly.

Method	$H(\bar{d})$	$\mathcal{H}$	$I(T;X)$	Worst goal (%)
Rooms-8 (clean)				
Random	2.450 [2.450,2.450]	1.451 [1.451,1.451]	0.999 [0.999,0.999]	88.2 [88.2,88.2]
Pooled PG	2.450 [2.450,2.450]	1.418 [1.415,1.420]	1.032 [1.030,1.035]	87.1 [86.7,87.4]
Count PG	2.450 [2.449,2.450]	1.420 [1.419,1.422]	1.029 [1.028,1.031]	87.4 [87.1,87.7]
Snapshot-min	2.259 [2.218,2.302]	0.007 [0.006,0.007]	2.252 [2.211,2.294]	40.1 [10.2,70.0]
Certainty PG	2.402 [2.384,2.416]	0.010 [0.010,0.010]	2.391 [2.374,2.406]	99.8 [99.8,99.9]
Stationary ablation	1.004 [1.004,1.005]	0.005 [0.005,0.006]	0.999 [0.999,0.999]	0.1 [0.1,0.1]
Scripted tour	2.443 [2.443,2.443]	0.000 [0.000,0.000]	2.443 [2.443,2.443]	100.0 [100.0,100.0]
Rooms-8 (slip 0.15)				
Random	2.450 [2.450,2.450]	1.451 [1.451,1.451]	0.999 [0.999,0.999]	88.2 [88.2,88.2]
Pooled PG	2.450 [2.450,2.450]	1.427 [1.426,1.428]	1.023 [1.021,1.024]	87.2 [87.0,87.4]
Count PG	2.449 [2.449,2.450]	1.429 [1.428,1.430]	1.021 [1.020,1.022]	87.1 [86.8,87.4]
Snapshot-min	2.327 [2.295,2.359]	0.453 [0.453,0.453]	1.874 [1.842,1.906]	45.4 [26.3,64.5]
Certainty PG	2.405 [2.392,2.416]	0.452 [0.452,0.452]	1.953 [1.940,1.964]	83.7 [70.9,90.1]
Stationary ablation	1.450 [1.450,1.450]	0.451 [0.451,0.451]	0.999 [0.999,0.999]	26.2 [26.2,26.2]
Scripted tour	2.445 [2.445,2.445]	0.449 [0.449,0.449]	1.996 [1.996,1.996]	98.7 [98.7,98.7]
Directed cycle-16 (clean)				
Random	3.895 [3.895,3.895]	2.585 [2.585,2.585]	1.309 [1.309,1.309]	70.2 [70.2,70.2]
Pooled PG	4.000 [4.000,4.000]	2.006 [1.971,2.040]	1.993 [1.959,2.027]	92.3 [89.8,94.4]
Count PG	3.960 [3.955,3.965]	2.434 [2.401,2.469]	1.526 [1.492,1.559]	76.0 [72.9,79.4]
Snapshot-min	3.882 [3.867,3.897]	0.004 [0.003,0.004]	3.878 [3.862,3.893]	100.0 [100.0,100.0]
Certainty PG	3.926 [3.913,3.940]	0.005 [0.005,0.005]	3.921 [3.908,3.935]	100.0 [100.0,100.0]
Stationary ablation	3.203 [3.155,3.235]	0.006 [0.005,0.006]	3.198 [3.149,3.230]	80.0 [50.0,100.0]
Scripted tour	3.978 [3.978,3.978]	0.000 [0.000,0.000]	3.978 [3.978,3.978]	100.0 [100.0,100.0]
Directed cycle-16 (slip 0.10)				
Random	3.895 [3.895,3.895]	2.585 [2.585,2.585]	1.309 [1.309,1.309]	70.2 [70.2,70.2]
Pooled PG	4.000 [4.000,4.000]	1.983 [1.930,2.033]	2.017 [1.967,2.068]	94.8 [92.8,96.4]
Count PG	3.952 [3.948,3.956]	2.441 [2.409,2.477]	1.511 [1.476,1.542]	78.8 [76.9,80.5]
Snapshot-min	3.900 [3.880,3.917]	0.329 [0.324,0.333]	3.571 [3.549,3.590]	100.0 [100.0,100.0]
Certainty PG	3.954 [3.943,3.963]	0.327 [0.321,0.332]	3.627 [3.613,3.640]	99.9 [99.8,99.9]
Stationary ablation	3.530 [3.511,3.550]	1.055 [1.036,1.073]	2.476 [2.475,2.477]	33.8 [23.2,44.5]
Scripted tour	3.977 [3.977,3.977]	1.358 [1.358,1.358]	2.618 [2.618,2.618]	100.0 [100.0,100.0]

bias of taking a minimum over noisy goal estimates. The probability of visiting *all* states is a different, joint event, estimated from the saved independent trajectories. Additional recorded metrics are mean distinct states, per-trajectory empirical entropy, action entropy, temporal information, synchronized disagreement $\sum_t w_t(1 - \sum_s d_t(s)^2)$, and exact constraint violation. Action entropy includes states with redundant actions in the rooms task, so it need not vanish when the state law is concentrated.

The independent statistical units are training seeds, not transitions, time indices, or evaluation replicas. Bootstrap intervals in the tables use their actual asymmetric endpoints; zero-width intervals can occur for exact fixed-policy metrics or identical seed outcomes. Rounded 100.0% does not assert that a finite-logit policy is deterministic. Paired seed-bootstrap comparisons against Random, Pooled PG, and Count PG subtract the baseline metric from Certainty PG's metric within each matched seed before resampling. For fixed policies, seed variation reflects evaluation sampling only.

I.3 Complete main results (extends Table 1) and learning dynamics

Figure 2 in the main text and Figure 6 show learned state laws and discovery outcomes. Here we provide the full objective and stationarity ablations and their training dynamics. Method names and comparison purposes are defined in Table 3.

I.4 Sweeps over the budget, transition noise (Figure 4), batch size, and number of rooms

The target sweep uses $\varepsilon \in \{0, 0.05, 0.15, 0.4, 0.8, 1.2\}$ on clean Rooms-8. At zero, finite softmax logits retain positive snapshot entropy in every run, and these runs are recorded as violating the target. The achieved tradeoff describes empirical optimizer outcomes; tightness of the constraints and optimality of the frontier are not established. For transition noise, Equation (34) supplies an independent lower bound. At $p = 0.30$, Certainty PG reaches 0.760 snapshot bits while keeping

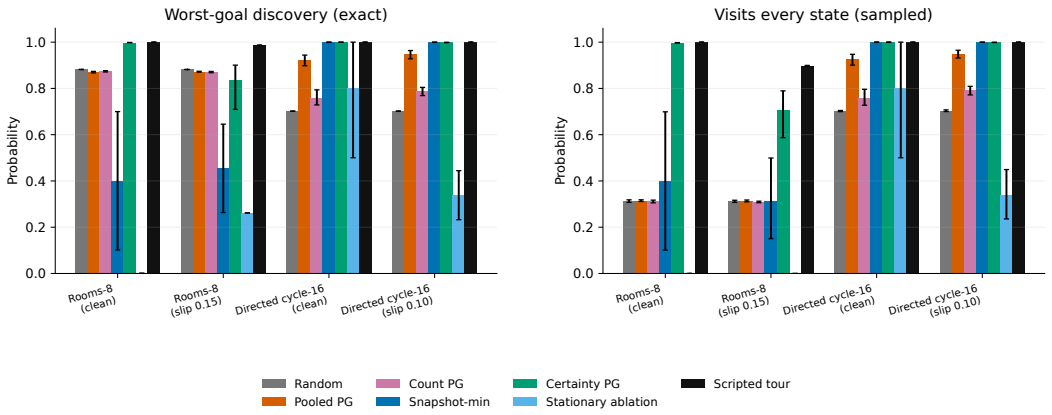


Figure 6: **Coverage within a single run.** Left: worst-goal discovery $D_{\min}$ (exact). Right: all-state coverage D_{all}. Neither is a training signal. Ten seeds, 95% bootstrap intervals, final iterates.

Table 7: Budget-matched synchronized-batch ablation on Rooms-8. Each run uses 4,915,200 training transitions. Intervals are 95% seed-bootstrap intervals; feasibility uses exact final snapshot entropy ≤ 0.1.

B	Updates	Pooled H	Snapshot H	All states (%)	Feasible
32	4800	2.449 [2.448,2.449]	0.111 [0.106,0.117]	99.8 [99.8,99.9]	0/10
64	2400	2.443 [2.441,2.446]	0.092 [0.070,0.109]	99.4 [98.7,99.8]	5/10
128	1200	2.392 [2.378,2.406]	0.008 [0.008,0.009]	99.8 [99.8,99.9]	10/10
256	600	2.402 [2.383,2.417]	0.010 [0.010,0.010]	99.8 [99.7,99.8]	10/10
512	300	2.385 [2.368,2.400]	0.013 [0.013,0.014]	99.6 [99.5,99.7]	10/10

2.416 pooled bits, but visits every state in only about 54.0% of trajectories. This distinguishes low excess uncertainty from complete coverage.

The batch sweep fixes 4,915,200 training transitions per run. Thus batch sizes 32, 64, 128, 256, and 512 use respectively 4,800, 2,400, 1,200, 600, and 300 updates. The sweep changes both estimator quality and update frequency, with Adam and dual step sizes held fixed throughout. Its effects therefore extend beyond estimator variance. High empirical training coverage at small B can still accompany a violation of the exact entropy constraint.

I.5 Navigation stress tests: when certainty locks onto an incomplete route

All ten Certainty PG runs in the clean bidirectional ring and open grid meet the 0.1-bit entropy constraint, yet none of their evaluation trajectories visit every state. In the slippery two-room grid the method also violates the constraint. The advantages over random exploration on structured tasks therefore do not extend uniformly to these environments. The failures are consistent with optimization difficulty and inadequate discovery before concentration, although the causal mechanism and local optimality remain unresolved. All results use the frozen hyperparameters.

Multipliers on the failure tasks. Figure 12 plots the original learning histories, not the scheduling follow-ups. All ten seeds cross one on every task. Logs are spaced ten updates apart, so the first logged crossing brackets the actual crossing: updates 31–40 on the ring, 21–30 on both grids, and 51–60 on Rooms-16. The cap is never reached. These observations establish that the multipliers enter the regime in Proposition 7; they do not by themselves identify the cause of a learned policy's failure.

I.6 Follow-up checks: alternative objectives, schedules, estimators, and robustness

The following checks were added after inspecting the original results. They address specific alternative explanations and do not replace the frozen-setting failures. Trajectory entropy and bias correction

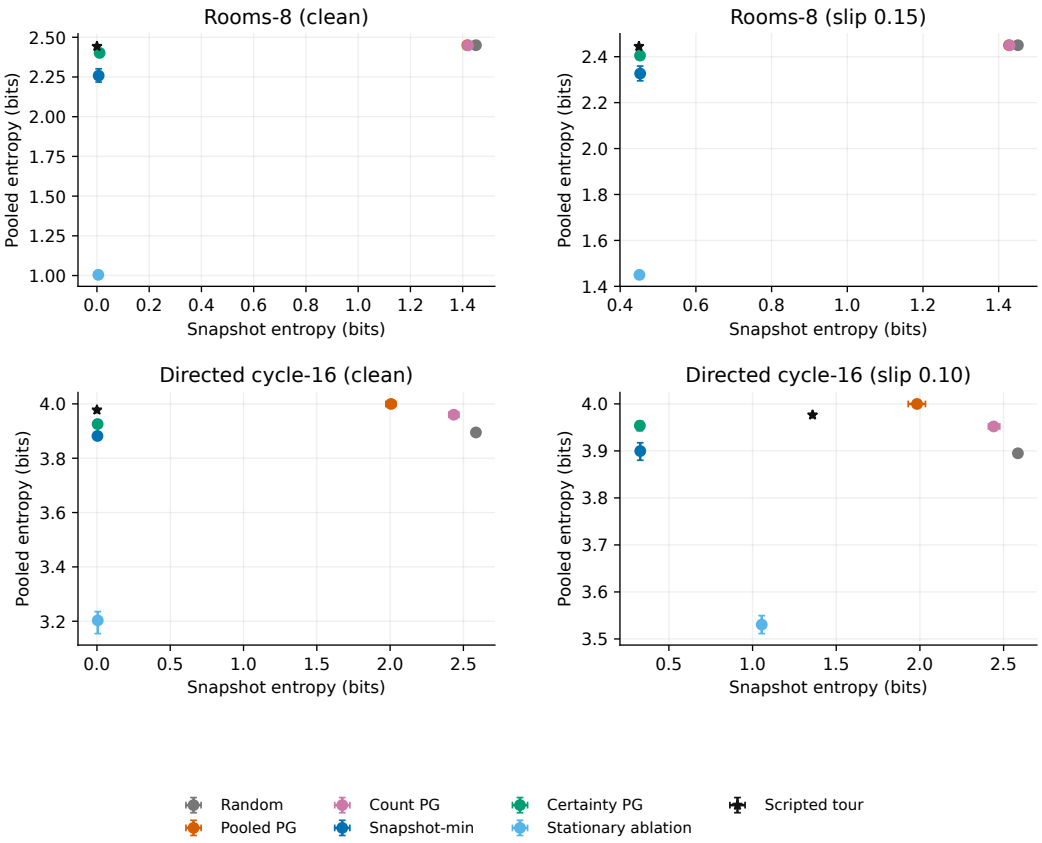


Figure 7: Coverage–uncertainty outcomes for all main-suite methods, with 95% seed-bootstrap intervals. The scripted tour is a hand-specified structural reference, with no general optimality guarantee in noisy dynamics.

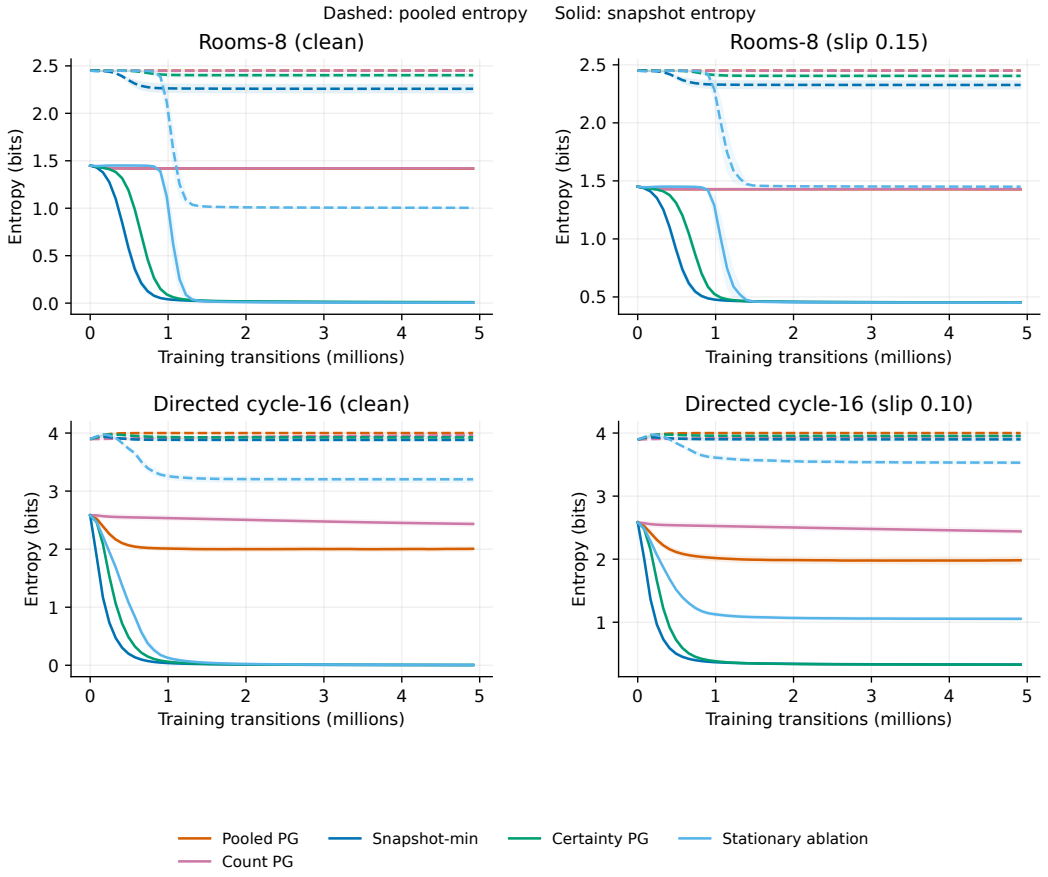


Figure 8: Main-suite learning curves against actual training transitions. Dashed curves are pooled entropy; solid curves are snapshot entropy. Shading uses 95% seed-bootstrap intervals. The final checkpoint is used even if an earlier one has a more favorable metric.

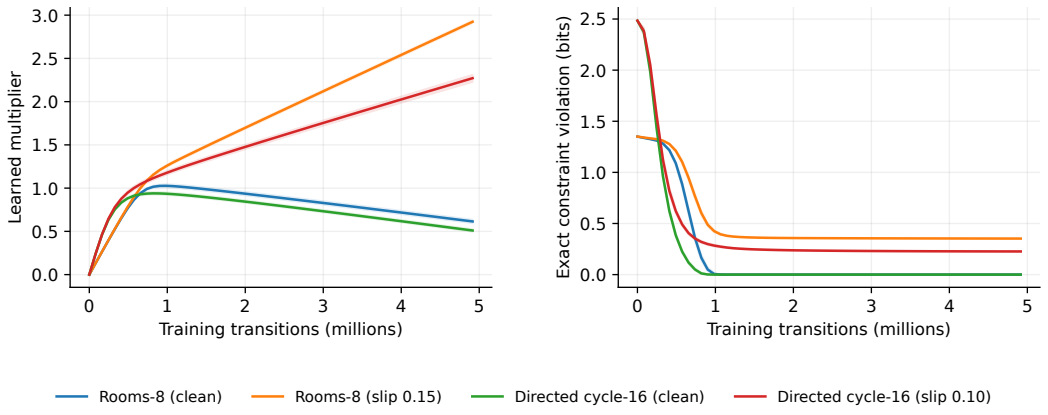


Figure 9: Certainty PG multipliers and exact constraint violations. Positive violations persist in the noisy environments despite increasing multipliers, so the specified 0.1-bit target remains unmet.

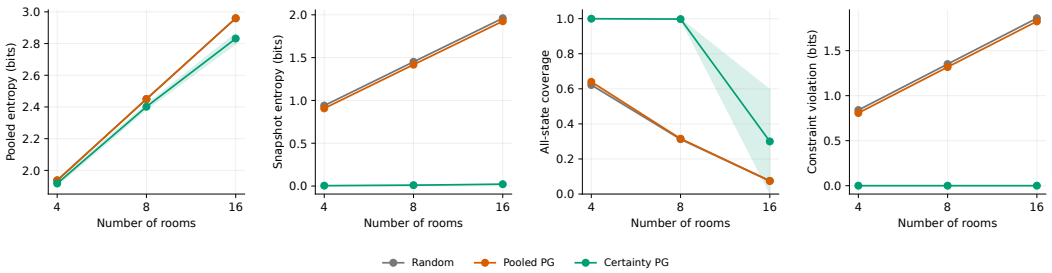


Figure 10: Size sweep with $K = 4R + 1$. Each task has two room visits per room on average, but larger action spaces make discovering complete schedules harder. Budgets match across methods within each size, not across sizes.

Table 8: Final-policy metrics in bits and worst-goal discovery (percent). Cells show mean [lower, upper] of a 95% seed-bootstrap interval. Entropy and goal-hit probabilities are evaluated exactly.

Method	$H(\bar{d})$	$\mathcal{H}$	$I(T;X)$	Worst goal (%)
Bidirectional ring-10				
Random	3.176 [3.176,3.176]	2.914 [2.914,2.914]	0.261 [0.261,0.261]	56.3 [56.3,56.3]
Pooled PG	3.321 [3.321,3.322]	2.827 [2.813,2.840]	0.494 [0.481,0.508]	67.1 [66.1,68.3]
Count PG	3.307 [3.305,3.309]	2.889 [2.879,2.899]	0.418 [0.410,0.427]	69.8 [69.0,70.4]
Certainty PG	2.731 [2.689,2.774]	0.004 [0.004,0.004]	2.727 [2.686,2.770]	0.0 [0.0,0.0]
Open grid 5×5				
Random	4.403 [4.403,4.403]	4.046 [4.046,4.046]	0.357 [0.357,0.357]	41.3 [41.3,41.3]
Pooled PG	4.643 [4.643,4.643]	4.016 [4.012,4.021]	0.627 [0.622,0.632]	53.8 [53.2,54.7]
Count PG	4.542 [4.538,4.547]	4.085 [4.076,4.094]	0.457 [0.452,0.462]	50.7 [50.5,50.9]
Certainty PG	3.483 [3.389,3.587]	0.006 [0.005,0.006]	3.478 [3.379,3.581]	0.0 [0.0,0.0]
Two-room grid (slip 0.10)				
Random	3.777 [3.777,3.777]	3.474 [3.474,3.474]	0.303 [0.303,0.303]	19.0 [19.0,19.0]
Pooled PG	4.392 [4.392,4.392]	3.632 [3.623,3.640]	0.760 [0.751,0.768]	38.4 [37.6,39.4]
Count PG	4.293 [4.286,4.301]	3.739 [3.722,3.757]	0.554 [0.542,0.566]	35.3 [34.0,36.5]
Certainty PG	3.622 [3.523,3.724]	0.585 [0.572,0.600]	3.036 [2.944,3.128]	0.1 [0.1,0.1]

inherit the original actor, initialization, learning rates, batch sizes, and transition budgets without a new hyperparameter search. Warm-up length is selected using calibration seeds 0–9 only. All reported follow-up comparisons use seeds 100–109 and the final iterate. Coverage checks use 4,096 evaluation trajectories per seed; the downstream check instead varies the number of collected episodes and evaluates planned policies exactly. Intervals use 5,000 percentile bootstrap resamples of whole seeds. As elsewhere, C, U, and $D_{\min}$ are evaluated by exact propagation; J and D_{all} are sampled. The mixture, medium-grid, downstream, and robustness protocols below were fixed before their respective outcomes were inspected. They are follow-ups, not part of the original frozen experiment deck.

The protocols and aggregate results are described below. Follow-up policies and raw evaluation trajectories were not retained, so independent reproduction requires implementing the stated modifications and rerunning the experiments; these summaries cannot be audited from saved follow-up trajectories. The original tabular and continuous-control records remain unchanged. The training-source fingerprint was checked before execution. Small-instance tests verified equality with the original optimizer when the modification is disabled, pooled-only behavior during warm-up, the occupied-support correction, and the trajectory-return gradient against finite differences (maximum error 8.7×10^{-12}).

I.6.1 Trajectory-entropy policy gradient (maximizing J directly)

For each sampled run, form its weighted empirical law $m_\tau(s) = \sum_t w_t \mathbf{1}\{S_t = s\}$. The return $H(m_\tau)$ is a function of that run and has no additional dependence on the policy parameters once the

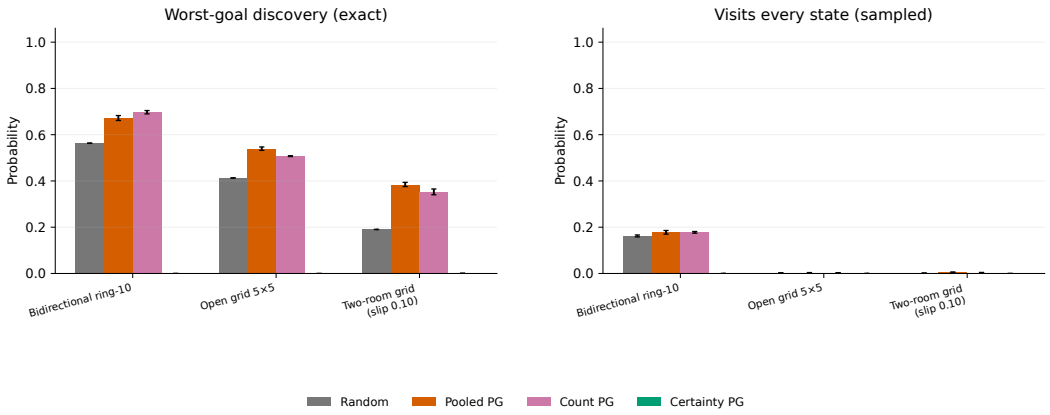


Figure 11: Navigation stress tests, including all run outcomes. Certainty PG's near-zero discovery rates reflect concentrated partial routes with poor coverage, even when the entropy constraint is satisfied.

Table 9: Trajectory-entropy PG: entropy means [95% seed-bootstrap intervals] in bits. J uses geometric within-run weights, not the unweighted trajectory metric stored in the original results.

Task	C	U	J
Rooms-4	1.935 [1.934,1.936]	0.017 [0.016,0.018]	1.934 [1.933,1.935]
Rooms-8, clean	2.445 [2.443,2.446]	0.114 [0.097,0.131]	2.436 [2.434,2.438]
Rooms-8, noisy	2.447 [2.447,2.448]	0.557 [0.541,0.574]	2.383 [2.380,2.386]
Rooms-16	2.952 [2.950,2.953]	0.563 [0.523,0.607]	2.908 [2.904,2.913]
Cycle-16, clean	3.974 [3.970,3.978]	0.117 [0.096,0.147]	3.968 [3.962,3.974]
Cycle-16, noisy	3.988 [3.986,3.989]	0.868 [0.749,0.987]	3.948 [3.943,3.953]
Bidirectional ring	3.311 [3.308,3.314]	0.196 [0.165,0.234]	3.307 [3.303,3.310]
Open grid	4.618 [4.613,4.623]	0.521 [0.472,0.572]	4.595 [4.590,4.600]
Slippery grid	4.381 [4.379,4.383]	1.547 [1.496,1.597]	4.292 [4.289,4.295]

run is fixed. The likelihood-ratio identity therefore gives the REINFORCE gradient (Williams, 1992)

$$\nabla_\theta J(\theta) = \mathbb{E}_\tau \left[H(m_\tau) \sum_{k=0}^{K-2} \nabla_\theta \log \pi_\theta(A_k \mid S_k, k) \right]. \tag{36}$$

We subtract the same per-time baseline fitted to previous batches and use the existing Adam step, norm clipping, and logit clipping. In particular, we do not substitute a trajectory-local density into the snapshot return: the empirical visitation depends on the whole run, including states before the action. Giving the terminal entropy to every action implements (36) directly and requires neither density smoothing nor leave-one-out estimation.

The check covers the nine distinct tabular coverage tasks in the main, size, and navigation suites (90 trainings); the three-state reference solver is not a coverage benchmark. The horizon is 17 for Rooms-4, 33 for the four main tasks and the bidirectional ring, and 65 for Rooms-16 and the two grids. All use 600 updates and 256 replicas. The policy remains time-dependent Markov, so this comparison does not test the potential advantage of a history-dependent actor established by Mutti et al. (2022). Also, the geometric weighting differs from an unweighted empirical visitation objective.

The results distinguish a trajectory objective from a repeatability constraint. On clean Rooms-8, both methods give high all-state coverage; the direct trajectory objective attains larger J and C, while Certainty PG has much smaller U. On the navigation tasks, trajectory entropy improves coverage substantially over Certainty PG. Its failure to meet a constraint it was not trained to satisfy is not an optimization failure of that comparator. Conversely, Certainty PG's feasibility in the clean grids does not compensate for its poor coverage.

Table 10: Trajectory-entropy PG: discovery means [95% seed-bootstrap intervals] in percent. The last column counts policies whose exact snapshot entropy meets 0.1 bits, although this comparator does not optimize that constraint. Rounded 100.00% is not a deterministic-policy claim.

Task	$D_{\min}$	D_{all}	Feasible
Rooms-4	100.00 [100.00,100.00]	100.00 [99.99,100.00]	10/10
Rooms-8, clean	99.58 [99.52,99.66]	99.33 [99.21,99.46]	5/10
Rooms-8, noisy	92.90 [91.81,94.24]	79.76 [77.61,81.81]	0/10
Rooms-16	98.15 [97.87,98.36]	83.04 [80.81,84.85]	0/10
Cycle-16, clean	100.00 [100.00,100.00]	100.00 [100.00,100.00]	3/10
Cycle-16, noisy	99.99 [99.99,99.99]	100.00 [99.99,100.00]	0/10
Bidirectional ring	99.99 [99.99,99.99]	99.98 [99.96,100.00]	0/10
Open grid	99.62 [99.42,99.73]	98.71 [98.38,98.96]	0/10
Slippery grid	96.99 [96.09,97.71]	89.62 [87.02,91.64]	0/10

Table 11: Scheduling follow-up: means over seeds 100–109 at the original interaction budget. C, U are bits; discovery is in percent. Intervals are 95% seed-bootstrap intervals for D_{all}. Original rows are retained for comparison.

Task	Schedule	C	U	$D_{\min}$	D_{all}	Feasible
Ring	Original	2.731	0.004	0.006	0.00	10/10
	Warm-up 300	2.859	0.006	0.035	0.007 [0.000,0.015]	10/10
	Annealed	3.004	0.121	34.50	29.51 [0.01,59.20]	6/10
Open grid	Original	3.483	0.006	0.001	0.00	10/10
	Warm-up 300	3.762	0.009	0.009	0.00 [0.00,0.00]	10/10
	Annealed	3.564	0.011	0.005	0.00 [0.00,0.00]	10/10
Slippery grid	Original	3.622	0.585	0.085	0.002	0/10
	Warm-up 100	3.599	0.622	0.337	0.00 [0.00,0.00]	0/10
	Annealed	3.703	0.609	0.123	0.012 [0.000,0.027]	0/10
Rooms-16	Original	2.831	0.022	30.44	30.00	10/10
	Warm-up 100	2.847	0.028	50.34	49.31 [20.17,78.50]	10/10
	Annealed	2.924	0.145	84.99	74.05 [54.31,88.83]	1/10

I.6.2 Warm-up and budget annealing for the multiplier

The warm-up variant holds $\lambda = 0$ for the first N updates, then uses the original dual update with target 0.1 bits. The actor, Adam moments, baseline, and random-number stream continue without a restart. We compare $N \in \{100, 200, 300\}$ on seeds 0–9. The selection rule is fixed before the held-out comparisons: prefer more feasible calibration seeds, then smaller mean positive violation, then larger mean exact $D_{\min}$, then a shorter warm-up. It selects $N = 300$ for the ring and open grid and $N = 100$ for the slippery grid and Rooms-16. All three warm-up choices are feasible in all calibration seeds on the clean tasks; none is feasible on the slippery grid. The ring choice is based on a very small difference in calibration discovery, so it should not be interpreted as evidence that 300 is intrinsically best.

The second variant uses no warm-up and updates the dual at iteration $j = 1, \ldots, 600$ with

$$\varepsilon_j = 1.2 \left(\frac{0.1}{1.2} \right)^{(j-1)/599}.$$

Its endpoints and decay law are fixed, and it is evaluated on both seed sets without selecting an alternative schedule. Both variants use the same total 600 updates and 256 replicas as the original runs. Calibration therefore comprises 160 trainings (four variants, four tasks, ten seeds), followed by 80 held-out trainings (two reported variants, four tasks, ten seeds). Feasibility is always assessed against the final 0.1-bit target, not the larger intermediate annealing budgets.

Warm-up gives a modest increase in pooled entropy on the open grid but does not recover complete routes. The large Rooms-16 intervals reflect different schedules learned across seeds, not uncertainty over a single policy's exact marginals. Annealing improves some coverage outcomes at the expense of final feasibility. These results neither establish a general benefit of scheduling nor rule out better calibrated schedules.

Table 12: Dual-estimator comparison on clean Rooms-8. Means [95% seed-bootstrap intervals] for exact U; feasibility uses each seed's exact value rather than the mean. Ten seeds per batch size.

B	Plug-in U	Corrected U	Plug-in feasible	Corrected feasible
32	0.111 [0.106,0.117]	0.106 [0.103,0.108]	0/10	1/10
64	0.092 [0.070,0.109]	0.080 [0.056,0.103]	5/10	6/10
128	0.008 [0.008,0.009]	0.008 [0.008,0.008]	10/10	10/10
256	0.010 [0.010,0.010]	0.010 [0.010,0.010]	10/10	10/10
512	0.013 [0.013,0.014]	0.013 [0.013,0.013]	10/10	10/10

Table 13: Greedified Pooled PG, ten saved policies per task. Means [95% seed-bootstrap intervals]; entropies are bits and discovery is percent.

Task	C	U	$D_{\min}$	D_{all}
Rooms-8, clean	2.343	0	50.0 [20.0,80.0]	50.0 [20.0,80.0]
Cycle-16, clean	3.733	0	100.0 [100.0,100.0]	100.0 [100.0,100.0]
Rooms-8, noisy	2.382	0.449	70.9 [51.7,90.1]	56.9 [41.1,70.9]
Cycle-16, noisy	3.887	0.802	99.2 [98.0,99.9]	99.3 [98.2,99.9]

I.6.3 Bias-corrected (Miller–Madow) estimate of U in the multiplier update

Let $\hat{n}_t = |\{s : N_t(s) > 0\}|$. We replace only the dual's estimate by

$$\widehat{U}_B^{\mathrm{MM}} = \sum_t w_t \left[H(N_t/B) + \frac{\hat{n}_t - 1}{2B \ln 2} \right]. \tag{37}$$

The actor still uses the original smoothed leave-one-trajectory-out scores. Batch sizes 32, 64, 128, 256, and 512 respectively receive 4,800, 2,400, 1,200, 600, and 300 updates, for 4,915,200 training transitions per run. There is no tuning of a correction coefficient.

For a fixed strictly positive law on its support, the leading bias of $H(\hat{d}_t)$ is $(n_t - 1)/(2B \ln 2)$, not the nonasymptotic upper bound in Proposition 18. On Rooms-8, the relevant support sizes are one at even times and eight at odd times. Weighting them gives nominal leading-order allowances of 0.0763, 0.0382, 0.0191, 0.0095, and 0.0048 bits at the five batch sizes. Using nine states at every time would overstate this quantity. Near deterministic laws, some positive probabilities are very small relative to $1/B$, and the expansion is not uniform. The occupied-support estimate can also miss rare states. Thus (37) is not a feasibility certificate.

The correction slightly improves the small-batch constraint outcomes but leaves violations in 9/10 and 4/10 seeds at $B = 32$ and 64. High coverage does not remove those violations: the corresponding corrected all-state rates are 99.88% and 99.41%. The larger batches remain feasible, with all-state rates of 99.86%, 99.75%, and 99.62%. These tests support reporting exact violations rather than treating a corrected training estimate as proof of feasibility.

I.6.4 Making Pooled PG policies deterministic after training

We evaluate the saved final Pooled PG policies after replacing each action distribution by a point mass on its largest entry. Ties are broken by the smallest action index. There is no retraining, change of clock, or selection of a favorable seed. The clean deterministic tasks then have exact $U = 0$; their D_{all} values are also exactly zero or one for each policy. The noisy tasks retain transition randomness and use the same 4,096-rollout evaluation protocol.

The clean cycle is therefore a case where this simple post-processing already obtains complete predictable routes. On Rooms-8, however, greedification does not coordinate the repeated hub choices into a complete schedule in every seed. Neither result implies that greedification preserves the pooled objective: it changes the state laws and decreases mean C on all four tasks.

Table 14: Sampled Frank–Wolfe mixtures at the main-suite interaction budget. Entropies are in bits; all-state coverage is in percent. Entries are means over ten seeds, with 95% seed-bootstrap intervals for the mixing excess and all-state coverage.

Task	C	U	$\sum_i \alpha_i U_i$	ΔU	D_{all}
Rooms-8, clean	2.422	1.252	0.795	0.457 [0.435,0.482]	3.73 [3.33,4.16]
Cycle-16, clean	3.960	1.911	0.983	0.928 [0.833,1.036]	62.48 [54.50,70.88]
Rooms-8, noisy	2.436	1.353	1.107	0.246 [0.220,0.269]	9.34 [8.09,10.65]
Cycle-16, noisy	3.991	1.826	1.397	0.430 [0.372,0.492]	79.43 [69.14,88.95]

I.6.5 Episode-level policy mixtures (Frank–Wolfe planner)

We test a sampled Frank–Wolfe entropy planner on the four main tasks. This comparison uses the same tabular policy-gradient optimizer as Pooled PG, rather than an exact planning oracle or the original planner of Hazan et al. (2019). It isolates one way in which entropy can be distributed between a policy's own randomness and the choice of policy at the start of a run.

The initial mixture contains the uniform-action policy. Each of ten stages first collects 256 episodes from the current mixture, choosing one component per episode and keeping it throughout that episode. If $N_t(s)$ is the number of these episodes in state s at time t, the stage uses the density estimate

$$\widehat{\bar{d}}(s) = \frac{\sum_t w_t N_t(s) + 0.5}{256 + 0.5n}.$$

The reward $-\log_2 \widehat{\bar{d}}(s)$ is then held fixed for 59 policy-gradient updates of 256 episodes each. Returns retain the geometric time weights. The actor starts from the original random-logit initialization; its parameters, Adam moments, and past-batch baseline continue across stages. All optimizer settings match the main suite. At the end of stage $r = 1, \ldots, 10$, we retain that policy as a component and update the mixture with step size $\eta_r = 2/(r+2)$. The previous weights are multiplied by $1 - \eta_r$, and the new component receives weight η_r. There is no stage selection or tuning on these seeds.

The ten density batches are charged to the interaction budget: 590 actor-update batches plus ten density batches equal the original 600 batches, or 4,915,200 transitions per seed. The comparison comprises 40 trainings. For evaluation, the mixture has state law $d_t = \sum_i \alpha_i d_t^{(i)}$. We compute entropy *after* forming this law, and average the component goal-hit vectors before taking their minimum. Sampling an averaged action table would change the trajectory law and is not used. The excess snapshot entropy due to the unobserved component identity is

$$\Delta U = U(\pi_\alpha) - \sum_i \alpha_i U(\pi_i) = \sum_t w_t I(I; S_t).$$

The mixtures achieve high pooled entropy but substantially less within-run coverage than trajectory-entropy PG. Both component randomness and component selection contribute to U. The positive mixing excess illustrates the mixture identity on these tasks; it does not mean every mixture must increase uncertainty. The short, warm-started PG planner is approximate, so these results are not a performance limit for Frank–Wolfe methods or a reproduction of Hazan's experiments.

I.6.6 A larger navigation task: a 73-state two-room grid

To test a larger state space without searching for a favorable layout, we fix a deterministic two-room grid before evaluating any seed. Its coordinates are $0 \le x, y \le 8$; the wall removes the cells with $x = 4$ except the doorway at $(4, 4)$, leaving 73 states. The reset is $(0, 0)$. The four actions move right, down, left, or up; a blocked move leaves the state unchanged. The state is the cell, with no direction variable. This extends the existing grid family rather than introducing a new observation model.

We use $K = 193$ states, $\gamma = 0.99$, and the original 600 updates of 256 episodes. All other optimizer settings, including the 0.1-bit constraint, are unchanged. Each learned run therefore uses 29,491,200 transitions. The four learned methods have identical budgets and actor classes; Random has no training phase. All 40 training runs contribute to the reported results, with ten evaluation batches per method. No hyperparameter or map is selected after viewing these results.

Table 15: The 73-state two-room grid. Means over ten seeds; entropies are bits and discovery is percent. N_{vis} is the exact expected number of distinct states, obtained by summing goal-hit probabilities over *all* states. Brackets give 95% seed-bootstrap intervals. Zero sampled all-state coverage is not an exact zero-probability claim.

Method	C	U	$\mathbb{E}N_{\text{vis}}$	$D_{\min}$	D_{all}	Feasible
Random	5.481	5.177	35.40 [35.40,35.40]	11.01	0.00 [0.00,0.00]	0/10
Pooled PG	6.189	5.480	43.29 [43.16,43.42]	30.86	0.00 [0.00,0.00]	0/10
Count PG	6.025	5.491	42.64 [42.52,42.76]	28.47	0.00 [0.00,0.00]	0/10
Trajectory PG	6.140	1.532	71.47 [71.02,71.88]	49.62	26.31 [10.62,42.62]	0/10
Certainty PG	3.904	0.010	24.80 [23.61,26.30]	1.34×10^{-5}	0.00 [0.00,0.00]	10/10

Table 16: Downstream goal success in percent, averaged over all non-reset goals and ten seeds. Brackets are 95% seed-bootstrap intervals; 100 denotes 100 [100,100]. The optimal rate is 100%, so subtracting an entry from 100 gives its mean optimality gap in percentage points. Goals within a seed are averaged before resampling seeds.

Task	Explorer	1 episode	4 episodes	16 episodes	64 episodes
Rooms-8	Random	61.25 [37.50,85.00]	100	100	100
	Pooled PG	45.00 [26.25,65.00]	100	100	100
	Trajectory PG	100	100	100	100
	Certainty PG	100	100	100	100
Open grid	Random	32.50 [16.67,48.33]	89.17 [72.08,99.17]	100	100
	Pooled PG	34.58 [17.92,52.08]	91.67 [75.00,100.00]	100	100
	Trajectory PG	95.83 [92.50,98.75]	96.67 [93.33,99.58]	99.17 [97.92,100.00]	100
	Certainty PG	25.00 [16.67,33.75]	25.00 [16.67,33.75]	25.00 [16.67,33.75]	25.00 [16.67,32.93]

The larger map reproduces the concentration failure: Certainty PG meets the budget but covers fewer states than Random. Trajectory-entropy PG usually visits nearly all cells, although its full-coverage probability varies substantially across seeds and it does not meet the uncertainty budget. Its worst-goal discovery is 49.62% [23.09,75.22], compared with 30.86% [30.18,31.47] for Pooled PG. This is evidence about one larger layout under a fixed shared optimizer, not a general scaling study or evidence that no better Certainty PG settings exist.

I.6.7 Downstream goal reaching from the collected data

We next ask whether the collected experience supports a task revealed after exploration, following the reward-free evaluation principle of Jin et al. (2020). The tasks are clean Rooms-8 and the open 5×5 grid. We compare Certainty PG, Pooled PG, trajectory-entropy PG, and Random. Saved original policies are used when available; the trajectory comparator is regenerated with the same training seeds and settings. This is a comparison of data collected by the deployed policies: their pretraining trajectories are not supplied to the downstream learner.

For each policy and seed, we collect 64 fresh episodes and use nested prefixes of 1, 4, 16, or 64 episodes to fit an empirical transition model. These correspond to 32–2,048 transitions on Rooms-8 and 64–4,096 on the grid. The sampling seed is the training seed plus 20,000,033, distinct from the training and coverage-evaluation streams. For observed state–action pairs, the model is the normalized successor count. Every unobserved pair is assigned a self-loop, with no access to the true transition law or grid geometry. This default is shared by all methods.

A goal is then drawn uniformly from the non-reset states. Rather than sampling goals, we enumerate them all and average their success rates, which computes the same uniform-goal expectation exactly. A goal reward is one if the run reaches the goal within the original horizon and zero otherwise. Backward dynamic programming maximizes this hitting probability in the fitted model; ties use the smallest action index. We evaluate each resulting time-dependent policy under the true model and compare it with the true-model optimum at the same horizon. Thus the evaluator knows the dynamics, but the planner does not. Every true-model optimum in these two deterministic tasks is one.

On Rooms-8, the broad routes learned by both trajectory entropy and Certainty PG provide useful experience within one episode. The advantage disappears after four episodes, when all explorers reach the optimum. On the grid, Certainty PG's repeated partial routes leave the downstream planner at 25%

Table 17: Robustness checks: means over ten seeds. Snapshot entropy is in bits and all-state coverage in percent. Delay and reset shift are separate conditions with unchanged trained policies. Rounded near-100% coverage does not assert a deterministic guarantee.

		U			D_{all}		
Task	Explorer	Nominal	Delay	Reset shift	Nominal	Delay	Reset shift
Rooms-8	Random	1.451	2.164	1.892	30.58	26.73	31.07
	Pooled PG	1.418	2.144	1.862	31.12	27.02	31.50
	Trajectory PG	0.114	1.187	0.674	99.30	99.06	93.65
	Certainty PG	0.010	1.098	0.580	99.69	99.64	93.99
Cycle-16	Random	2.585	2.399	2.917	70.05	56.68	70.05
	Pooled PG	2.006	2.042	2.391	92.45	57.74	88.96
	Trajectory PG	0.117	1.053	0.697	> 99.99	> 99.99	99.07
	Certainty PG	0.005	0.795	0.580	100.00	49.67	95.88

mean success even with 64 episodes. At that budget, it has observed only 30.7% of state–action pairs, compared with 100% for Pooled PG and 96.3% for trajectory-entropy PG. The check therefore shows a task-dependent benefit of within-run coverage, not a general downstream advantage of repeatability. It is also limited to sparse goal rewards, deterministic tasks, and one model-fitting rule. In particular, self-loop defaults and tie-breaking can affect small-data plans; these are not results for arbitrary downstream rewards or alternative model learners.

I.6.8 Robustness to start delays and reset shifts

We evaluate the same four policy types on clean Rooms-8 and Cycle-16 without adapting them. The nominal condition repeats the original task. The delay condition independently draws $D \sim \mathrm{Unif}\{1, 2, 3\}$ for each execution, holds the agent at the reset state for D physical transitions, and then starts its learned policy clock at zero. The physical horizon remains $K = 33$, so the agent has D fewer action opportunities. Snapshot entropy is measured against the common *physical* time, without conditioning on D. This tests start latency and a shortened execution window, not a corrupted policy clock during movement.

The reset-shift condition instead uses $\rho' = 0.9\delta_{s_0} + 0.1\,\mathrm{Unif}(\mathcal{S})$ with no delay; the policy clock starts at zero as usual. These perturbations are tested separately. Exact state laws and goal-hit probabilities marginalize over the delay or reset randomness before computing metrics. All-state coverage still requires visiting every state, and $D_{\min}$ retains the original goal set $\mathcal{S} \setminus \{s_0\}$. Each condition has 4,096 fresh evaluation runs per seed, with sampling seed equal to the training seed plus 30,000,049. Its nominal Monte Carlo coverage can therefore differ slightly from the original headline table.

Rooms-8 retains high all-state coverage under delay, but the population is no longer aligned in physical time: Certainty PG's U rises to 1.098 [1.090,1.106] bits. On the cycle, delay also exposes reliance on late visits: all-state coverage falls to 49.67% [36.41,63.27], while trajectory-entropy PG remains above 99.99% on average. The reset shift reduces Certainty PG's all-state rates to 93.99% [93.81,94.19] and 95.88% [95.73,96.05] on Rooms-8 and the cycle, respectively. In both cases U is approximately 0.58 bits. A budget learned for one reset law and timing convention does not automatically hold after either changes. These checks do not establish worst-case robustness or performance under other delays, reset laws, or mid-run disturbances.

I.7 Two-state diagnostic and grid value iteration (Section F) on a tiny MDP

The two-state diagnostic uses $\gamma = 0.97$. Its infinite-horizon pooled entropy is 0.999832709 bits for both processes, while snapshot entropy is 0 for alternation and 0.969837728 bits for a random location held after time one. Empirical synchronized batches use $B \in \{32, 128, 512, 2048, 8192\}$ and 32 independent seeds per size. A horizon of 512 leaves tail mass 1.69×10^{-7}. Temporal information is computed directly from the joint law, not defined as an entropy subtraction, and the maximum diagnostic residual is 1.2×10^{-15}.

For Algorithm 2, the three-state branch routes state 0 to one of two absorbing states and starts at $(0.6, 0.2, 0.2)$. At $\gamma = 0.9$, the infinite-horizon minimum and maximum are 0.786830345 and 1.037095059 bits. For an arbitrary initial distribution, the reachable next-state mass at absorbing

state 1 ranges from $d(1)$ to $d(1) + d(0)$, providing exact values on the entire simplex. We run $N \in \{4, 8, 12, 16\}$ and 10, 30, or 80 backups, for both extrema. Full-control backups intersect the convex hull of pure successors with each interpolation cell and optimize its affine value at intersection vertices. Since interpolation can lose the concavity required by the pure-minimizer theorem, these backups retain mixed controls as well.

At $N = 16$ and 80 backups, the initial-law errors are 0.02384 bits for minimization and 0.00076 bits for maximization. The corresponding maximum errors on 501 fixed test distributions are 0.11433 and 0.03462 bits. These sampled maxima are not uniform error certificates. The analytic uniform bound from Theorem 21 is conservative (about 6.686 bits here, larger than the value range), and does not include a formal floating-point round-off certificate. Algorithm 1 on the same branch is compared with the *normalized finite-horizon* optimum for its own $K = 65$ target, yielding mean gaps 0.00105 and 0.00037 bits. Grid value iteration is used here as a tiny-state reference; the 21–25-state gridworlds exceed its practical scale.

I.8 Software, tests, and compute

The implementation is a NumPy/SciPy package with separate core algorithms, parametric benchmark factories, YAML inputs, CLI execution, and plotting modules. For internal validation, each original seed job stores effective training/environment settings, a source fingerprint and source snapshot, package versions, final policy, exact state laws, raw evaluation states/actions, learning curves, and runtime. Cached results are keyed by source and configuration, so changing parameters cannot silently reuse an incompatible run. Internal checks verify completeness and regenerate original-study tables and figures from the complete raw records without performance-based run selection.

Tests verify finite-difference gradients for pooled, snapshot, and Lagrangian objectives; empirical score gradients; full-trajectory leave-one-out counts; exact goal-hit formulas; stationary sharing; mixed-control grid backups; and configuration-safe resumption. The final deck used four CPU worker processes, with one numerical-library thread per worker, on an AMD Ryzen AI 7 PRO 350 (8 cores, 16 logical processors). These runs were executed on Windows with Python 3.14.6, NumPy 2.5.3, SciPy 1.18.1, Matplotlib 3.11.2, and PyYAML 6.0.3. No GPU or external dataset is required. The deck totals 2,949,120,000 vectorized training transitions and 112,066,560 evaluation transitions. Per-job timing is recorded; summing worker times is not the same as elapsed wall time. Learned-density visual experiments, encoder ablations, and downstream fine-tuning remain outside the scope of these measurements.

The motivating two-room figure and the learning workflow are explanatory assets generated separately from the held-out results. The former uses analytically specified policies at $K = 5$, $\gamma = 0.9$. The empirical heatmaps show held-out seed 100 under the actual main-suite settings.

J Numerical Checks of the Theory and Reproducibility Records

Temporal information is computed independently from the joint law; across all 770 held-out records its decomposition residual is at most 1.6×10^{-15}. Finite-difference and sampled-gradient tests check the score estimator directly. On the three-state absorbing branch, Snapshot-min and Snapshot-max PG have mean gaps 0.00105 and 0.00037 bits to their exact finite-horizon optima (Section I.7).

The reproducibility information provided here consists of the proofs, algorithm descriptions, settings, evaluation procedures, and reported results. For the original studies, internal records retain the effective inputs, seeds, final policies, evaluation trajectories, and learning curves. The tabular deck comprises 770 jobs in 77 ten-seed cells, including repeated settings across sweeps. The continuous-control deck adds 50 jobs and 250 independently evaluated checkpoints; all saved spatial metrics recompute from the raw replicas.

K Continuous-Control Experiments: Ant and Humanoid (Section 6)

Ant and Humanoid extend the comparison from exactly evaluable state laws to neural policies acting through contact dynamics. They also connect directly to the spatial occupancy plots of Hazan et al.

(2019). We use Gymnasium Ant-v5 and Humanoid-v5 (Towers et al., 2024), discard the simulator's task reward, and estimate both entropies on a fixed grid of torso position $Z_t = (x_t, y_t)$. The grid has 33×33 cells, of width 0.5 m for Ant and 0.25 m for Humanoid, plus an overflow category. These are spatial-projection experiments: concentration in this representation is neither concentration of the full robot state nor successful locomotion.

Comparison methods and shared budget. Certainty PG uses the synchronized pooled and time-indexed densities from Algorithm 1, now with a time-conditioned neural Gaussian actor and a past-fitted value baseline. Pooled PG removes the snapshot penalty while keeping the actor and update procedure unchanged. Two further comparators implement the conditional-gradient construction of Hazan et al. (2019): estimate a policy mixture's occupancy, freeze a reward obtained by differentiating the occupancy functional, train a new component, and add it to the mixture. *FW-entropy (PG)* uses $-\log_2 \widehat{\bar{d}}$; *FW-uniform-KL (PG)* uses $1/(\widehat{\bar{d}} + \sigma)$, with $\sigma = n^{-1/2}$ for $n = 1090$ categories. The latter follows the smoothed uniform-target KL functional used in Hazan et al.'s experiments, which is distinct from Shannon entropy. Components are sampled once per execution; evaluation averages over that choice before computing the snapshot law.

Both mixture comparators use the same neural PG planner as the direct methods. They are controlled, limited-budget adaptations, not reproductions of the original SAC planner, high-dimensional density representation, or long-rollout protocol. Section L records these differences. Each trained policy uses 524,288 interactions: eight epochs of 65,536. One mixture-density batch per epoch is charged to that budget, leaving seven PG batches for the mixture and eight for a direct method. Uniform random actions provide a fixed, untrained reference. All executions have 256 transitions ($K = 257$), a fixed reset without perturbation, and $\gamma = 0.995$; unhealthy dynamics continue without resetting. Certainty PG targets $U \leq 1$ bit. Five held-out training seeds, 200–204, follow separate seed-zero calibration, without per-method hyperparameter search.

Checkpoint-level comparison. Epochs 0, 2, 4, 6, and 8 each receive 128 fresh evaluation executions. Pooled log-occupancy maps show the spatial footprint across all times; colored replica clouds retain elapsed time (Figure 16). A broader footprint can result from executions separating rather than from each execution following a broad route. We therefore report both distinct bins within an execution and the unbinned positional statistic

$$R = \left[\sum_{t=0}^{K-1} w_t \frac{1}{B(B-1)} \sum_{i \neq j} \| Z_t^i - Z_t^j \|_2^2 \right]^{1/2}. \tag{38}$$

The spatial panels use the preselected smallest seed, 200, with fixed axes and common occupancy color scales. Learning and within-execution curves instead aggregate all five training seeds, with 95% seed-bootstrap intervals. No best checkpoint or favorable execution is selected.

Ant: concentration with a smaller footprint. Certainty PG reduces mean replica separation from Pooled PG's 2.51 m to 1.13 m, and snapshot entropy from 4.88 to 3.11 bits (Table 18). The paired seed-bootstrap difference in R is -1.38 m $[-2.08, -0.67]$. The improvement in concentration comes with fewer distinct bins per execution, 8.6 rather than 15.4, and lower pooled entropy, 3.88 rather than 6.19 bits. The two FW variants retain about 6.0 pooled bits and 14.6–14.7 bins per execution, but approximately 4.8 snapshot bits. Uniform actions have the largest pooled entropy in this limited-budget Ant comparison. Thus the learned contraction is visible, but the broad, nearly deterministic tours of the tabular tasks do not emerge here. All five Certainty PG seeds exceed the one-bit target.

Humanoid: spatial certainty can accompany a fall. Pooled PG and FW-entropy increase the spatial footprint relative to random actions, reaching respectively 5.08 and 5.15 pooled bits. Certainty PG instead contracts to $R = 0.25$ m and $U = 1.74$ bits, compared with Pooled PG's 1.05 m and 3.63 bits. Their paired difference in R is -0.80 m $[-0.88, -0.71]$, but distinct bins per execution fall from 8.2 to 4.7. More importantly, Certainty PG is healthy in only 7.9% of recorded states, versus 18.3% for Pooled PG. No final evaluation execution of any method remains healthy throughout. Replaying the first saved execution of seed 200 shows the associated fall-and-settle behavior (Figure 26); it is not a best- or worst-case selection. The tighter clouds therefore indicate more repeatable spatial

Table 18: Continuous-control final-iterate means over 5 training seeds. Entropies are of binned torso position, not full robot state. R is weighted RMS separation of distinct replicas; V counts bins within one execution. Health is the simulator predicate, not a locomotion score. Seed-bootstrap intervals appear in Table 20.

Method	C	U	R (m)	V	Health (%)	$U \leq 1$
Ant-v5						
Uniform random	6.57	5.23	2.98	16.1	94.8	0/5
Pooled PG	6.19	4.88	2.51	15.4	97.4	0/5
Certainty PG	3.88	3.11	1.13	8.6	99.6	0/5
FW-entropy (PG)	5.99	4.83	2.29	14.7	97.7	0/5
FW-uniform-KL (PG)	5.99	4.85	2.26	14.6	98.0	0/5
Humanoid-v5						
Uniform random	4.53	3.40	0.66	5.8	9.4	0/5
Pooled PG	5.08	3.63	1.05	8.2	18.3	0/5
Certainty PG	2.85	1.74	0.25	4.7	7.9	0/5
FW-entropy (PG)	5.15	3.88	0.98	6.7	10.6	0/5
FW-uniform-KL (PG)	4.80	3.53	0.81	6.6	9.8	0/5

outcomes, not stable exploration or improved physical control. The one-bit target is unmet in all five constrained runs here as well.

Together, the epoch plots and independent metrics distinguish the objectives without establishing superiority over maximum-entropy exploration. In these runs the constraint reduces spatial dispersion but does not preserve broad coverage or achieve feasibility. The comparison also identifies a limitation of the spatial representation: occupancy and repeatability can improve while useful physical behavior remains absent. Learned task competence, a stronger planner, and richer density representations remain separate empirical questions.

L Continuous-Control Protocol and All Checkpoints (for Section K)

L.1 Differences from the maximum-entropy protocol of Hazan et al. (2019)

The final published experiments of Hazan et al. (2019), Sections 5.1–5.2 and Figures 2–3, include Ant and Humanoid. Their framework uses conditional-gradient optimization of visitation functionals with an approximate planning oracle. The experimental functional minimizes a smoothed $\mathrm{KL}(\mathrm{Unif}\|d)$, so its ascent linearization is inverse density, rather than the logarithmic score of Shannon entropy. We retain both as separately named comparators.

The source comparison uses the author's public repository `https://github.com/abbyvansoest/maxent`, commit `84a1258efe27e32e41880afcf593ebf9deab78c6`. In particular, the Ant and Humanoid collectors call the component selector inside the action loop. That switching policy need not have the same visitation law as an episode mixture. We use the latter to preserve the affine occupancy interpretation, and do not condition snapshot entropy on component identity. The initial component is uniform random actions. At round $k \geq 1$, the new component gets weight $\eta_k = 2/(k+2)$ and all previous weights are multiplied by $1 - \eta_k$. After k rounds there are $k+1$ components, with weights $2(j+1)/[(k+1)(k+2)]$, $j = 0, \ldots, k$. The neural actor, optimizer and value baseline are warm-started between rounds, while archived components are immutable. The finite PG solve is not a certified planning oracle, and the theoretical guarantee for an accurate oracle is not asserted for these runs.

L.2 Actor, density estimates, and resets

The actor receives the full modern observation, including absolute position: 107 entries for Ant and 350 for Humanoid, with respectively 8 and 17 bounded continuous actions. Each physical entry o is mapped to $\tanh(o/5)$. The clock adds τ, $\sin(2\pi\tau)$, $\cos(2\pi\tau)$, $\sin(4\pi\tau)$ and $\cos(4\pi\tau)$, with $\tau = t/256$. Two 64-unit tanh hidden layers parameterize the Gaussian mean; a learned state-

Table 19: Material differences from the 2019 continuous-control protocol. The new results compare matched adaptations, not published numerical scores.

Component	Hazan et al., final paper / released code	This comparison
Simulator	Legacy MuJoCo and modified OpenAI Gym	Gymnasium 1.3.0, MuJoCo 3.13.0, Ant-v5 / Humanoid-v5
Planner	SAC; two 300-unit ReLU layers	Shared sampled PG; two 64-unit tanh layers; warm-started across mixture rounds
Ant density	xy plus five random-projection coordinates; 16^2 spatial bins and 15 bins per projected coordinate	33^2 fixed xy bins and one overflow category
Humanoid density	Epanechnikov KDE, bandwidth 0.10; released code also uses PCA to 32 coordinates	The same fixed spatial counting representation as above, at Humanoid-specific spatial scale
New-policy budget	30×5000 planning steps per epoch	$7 \times 32 \times 256 = 57{,}344$ planning steps per mixture epoch
Mixture estimation	Ant: $10 \times 10{,}000$; Humanoid: $20 \times 50{,}000$ steps per epoch	$32 \times 256 = 8192$ steps per epoch, included in the matched training budget
Mixture semantics	Theory: component sampled per execution; released collectors can resample at each action	Component sampled once per execution, giving a convex mixture of occupancy measures
Evaluation	Long-rollout entropy, spatial occupancy and distinct-bin curves	Fresh 256-step replicas, both entropies, same-time distance, within-execution bins, height and health

independent diagonal log standard deviation starts at -0.7 and is clipped to $[-3, 0.5]$. The Gaussian sample is squashed through tanh and affinely mapped to the environment action bounds. Evaluation samples the same stochastic policy; it does not replace it by the mean action.

Linear layers use PyTorch's default initialization, except that the actor's output weights are scaled by 0.01 and its output bias is zero. The value baseline's output weights and bias are initialized to zero.

The score-function update treats sampled actions as stopped values and uses the squashed density with its Jacobian. Future returns include only states after the action, with the normalized γ^t weights already incorporated. Actor and baseline use separate Adam optimizers at rate 0.003, with $(\beta_1, \beta_2) = (0.9, 0.999)$, denominator offset 10^{-8}, no weight decay, and gradient-norm clipping at 5. The baseline has the same hidden width and is fitted for four full-batch epochs *after* the actor update, so that the baseline used by an update was fitted only to past batches. There is one score update per batch, without PPO clipping, off-policy replay, an action-entropy bonus, or task-reward shaping.

Direct PG uses the whole-trajectory leave-one-out estimates in Section I, now with $n = 1090$, $B = 32$, and total pseudocount one ($\alpha = 1/n$ per category). A fixed uniform-density constant is subtracted from the score to reduce variance; it does not change the population score gradient. Certainty PG starts at $\lambda = 0$, updates it with rate 0.1 from the unsmoothed training-batch snapshot entropy, and projects it to $[0, 20]$. The target is one bit.

For a mixture, the density batch is collected before planning the next component. Weighted counts plus the same total pseudocount produce a frozen density p. The reward table is $-\log_2 p$ or $1/(p+n^{-1/2})$. Its p-weighted mean is subtracted and its p-weighted standard deviation, floored at 10^{-3}, divides the table. This positive affine rescaling does not change the optimizer of the frozen, fixed-horizon linear reward, although finite gradient optimization remains scale-sensitive. The table is not recomputed from the planning trajectories within that round.

All resets use the simulator's unperturbed default initial state (reset-noise scale zero). Ant's action step is 0.05 s and Humanoid's 0.015 s, so 256 transitions correspond to 12.8 s and 3.84 s. These different physical durations are not normalized across robots. Unhealthy termination is disabled and no within-rollout reset is allowed. Unexpected termination or nonfinite state values fail a job rather than silently shortening a trajectory. The usual simulator task reward is discarded throughout. Health is recorded from the environment predicate: Ant includes a finite-state test and torso height in

$[0.2, 1.0]$; Humanoid uses height strictly between 1 and 2 m. These predicates do not certify balance, useful gait, or task competence.

L.3 Evaluation, estimator bias, and all checkpoint figures

Five training seeds 200–204 are evaluated at epochs 0, 2, 4, 6, and 8, with 128 new independent executions per checkpoint. Network initialization uses the training seed directly. Separate SHA-256-derived streams are used for training rollouts, mixture-density collection, and evaluation, indexed by update, mixture round, and checkpoint, respectively. Evaluation never feeds back into learning. Random is re-evaluated at each matched budget marker but consumes no training interactions. Its variation across markers is Monte Carlo variation of a fixed policy. Direct actors begin from the same initialization; mixture comparators begin with their uniform-random component, so epoch-zero values need not match the direct actors.

The spatial grid covers $[-8.25, 8.25]^2$ for Ant and $[-4.125, 4.125]^2$ for Humanoid. The origin lies in the interior of the central bin. Points outside the square use a separate category, rather than being clamped into an edge bin. That category is included in entropy but excluded from distinct in-window coverage. The raw xyz positions, actions, health indicators, reset seeds, and mixture-component indices are retained for every evaluation execution. Positional dispersion uses unbinned coordinates, including any outside the grid.

Entropies use unsmoothed plug-in marginal counts separately for each policy and training seed, then the seed-level estimates are summarized. They are not exact continuous-state entropies. Unlike the tabular evaluation, no transition model is available to remove sampling bias. The nominal support has 1090 categories, while training uses only 32 replicas per time. Consequently the training estimate is bounded above by five bits, and sparse occupancy can substantially bias entropy estimation (Paninski, 2003). The independent 128-replica evaluation helps expose this mismatch but does not eliminate bias or provide a feasibility certificate. The normalized horizon has omitted geometric mass $0.995^{257} = 0.276$, precluding an infinite-horizon feasibility interpretation.

Figure 24 re-evaluates the same final positions at 17, 33 and 65 bins per spatial axis with fixed extents, and at 16, 32, 64 and 128 replicas using nested, prespecified prefixes. No policy is retrained or selected for these analyses. Absolute entropy changes with resolution and sample count, but Certainty PG retains the lower mean spatial uncertainty in these comparisons. This is an estimator/representation sensitivity analysis, not proof of full-state certainty or of statistical consistency at the available batch sizes.

All confidence intervals use 5000 percentile resamples of the five training seeds, never individual time steps as independent replicates. Curve resampling preserves each seed's entire curve. For random actions the seed variability is evaluation variability only. The figures use fixed spatial axes within each robot, common logarithmic probability limits $[10^{-5}, 1]$, and shared within-execution vertical limits across methods. Zero-occupancy cells are blank. Values below the displayed color range retain their actual contribution to the metrics. Spatial panels show seed 200; quantitative curves and tables use all five seeds.

L.4 Compute and audit record

The separate continuous module leaves the frozen tabular source and records unchanged. Internal records cover 50 complete jobs, 250 evaluation batches, neural checkpoints, component weights, training histories and source snapshots. Each learned run uses 524,288 training transitions; the whole deck uses 20,971,520 training transitions, of which 1,310,720 estimate mixture densities, plus 8,192,000 independent evaluation transitions. Rendering replays add 2560 transitions used only to reproduce saved frames. No failed or incomplete seed is excluded from the aggregate. The audit verifies 810 run artifacts and recomputes all checkpoint metrics; the maximum empirical decomposition residual is 2.7×10^{-15}.

Runs use the same four-worker CPU machine as the tabular study, with PyTorch 2.14.0+cpu, Gymnasium 1.3.0 and MuJoCo 3.13.0; CUDA is not used. Elapsed times are retained exactly as recorded. Median per-job elapsed time was about 162 s for Ant and 363 s for Humanoid, but four late Humanoid jobs recorded approximately 7.5 h after a long execution gap. That gap was not separately instrumented, so these elapsed values are not used as algorithmic compute-efficiency measure-

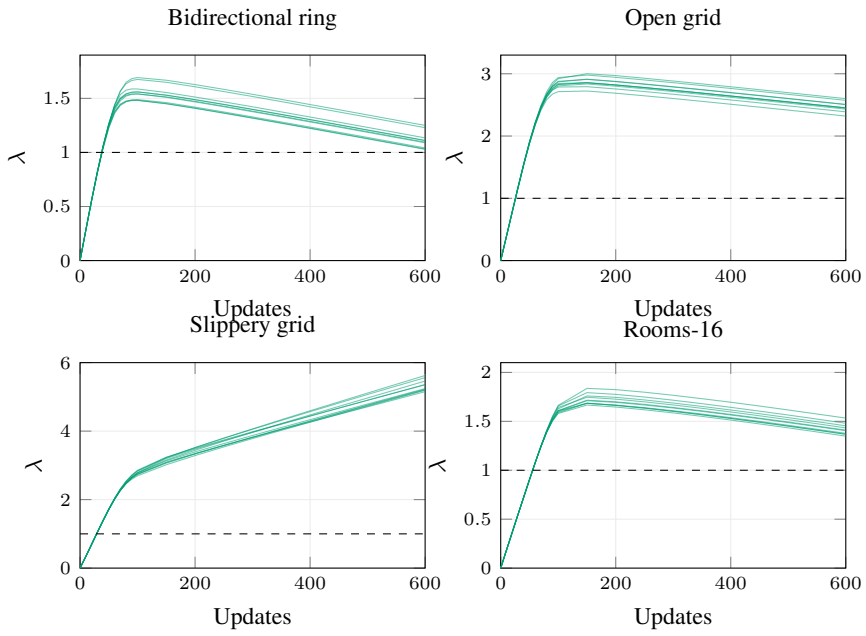


Figure 12: **Multipliers in the original failure-task runs.** Every line is one seed, 100–109; the dashed line is $\lambda = 1$. For clarity, the plot uses every logged update through 100 and every fifth logged update thereafter, with linear connections and no smoothing. Axes differ vertically to show seed variation. The crossing is observed in all seeds, but does not certify a local optimum or establish causation.

Table 20: Continuous-control means and 95% percentile-bootstrap intervals across 5 independent training seeds (5,000 resamples). All evaluation uses 128 fresh replicas of the final policy; no checkpoint selection.

Method	C (bits)	U (bits)	R (m)	Bins / execution	Health (%)
Ant-v5					
Uniform random	6.57 [6.54, 6.61]	5.23 [5.19, 5.26]	2.98 [2.90, 3.03]	16.1 [15.8, 16.4]	94.8 [94.6, 95.1]
Pooled PG	6.19 [5.92, 6.46]	4.88 [4.71, 5.06]	2.51 [2.24, 2.78]	15.4 [14.0, 16.8]	97.4 [96.0, 98.8]
Certainty PG	3.88 [2.85, 4.84]	3.11 [2.24, 3.92]	1.13 [0.69, 1.60]	8.6 [5.6, 11.6]	99.6 [99.1, 99.9]
FW-entropy (PG)	5.99 [5.95, 6.04]	4.83 [4.81, 4.86]	2.29 [2.22, 2.36]	14.7 [14.3, 15.3]	97.7 [97.3, 98.1]
FW-uniform-KL (PG)	5.99 [5.87, 6.10]	4.85 [4.79, 4.90]	2.26 [2.13, 2.40]	14.6 [14.1, 15.1]	98.0 [97.7, 98.2]
Humanoid-v5					
Uniform random	4.53 [4.45, 4.59]	3.40 [3.33, 3.45]	0.66 [0.64, 0.68]	5.8 [5.8, 5.9]	9.4 [9.4, 9.5]
Pooled PG	5.08 [4.94, 5.22]	3.63 [3.35, 3.84]	1.05 [0.90, 1.19]	8.2 [7.5, 8.8]	18.3 [11.5, 25.7]
Certainty PG	2.85 [2.39, 3.40]	1.74 [1.32, 2.23]	0.25 [0.17, 0.36]	4.7 [4.1, 5.4]	7.9 [6.9, 9.0]
FW-entropy (PG)	5.15 [4.93, 5.37]	3.88 [3.74, 4.04]	0.98 [0.87, 1.11]	6.7 [6.1, 7.3]	10.6 [7.7, 13.6]
FW-uniform-KL (PG)	4.80 [4.40, 5.15]	3.53 [3.14, 3.86]	0.81 [0.69, 0.94]	6.6 [5.4, 7.9]	9.8 [7.6, 12.0]

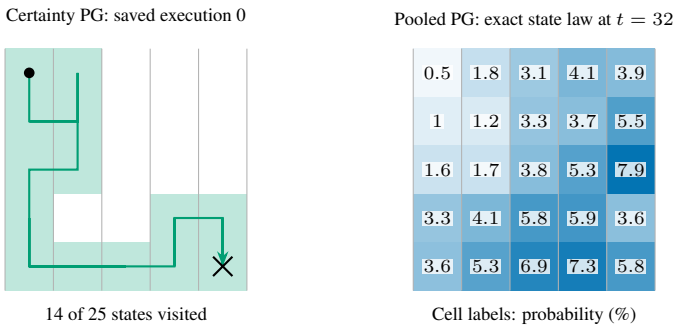


Figure 13: **A repeatable partial route on the open grid.** Both panels use the prespecified original seed 100. Left: the first saved independent evaluation execution; consecutive waits are omitted from the line, and repeated moves overlap. The dot is the reset and the cross the final state. At $t = 32$, the exact Certainty PG state law puts 99.97% on that bottom-right state. Right: Pooled PG's exact law at the same time remains spread over the grid. One execution and a population law are different objects; the panels illustrate, rather than establish causally, the concentration failure.

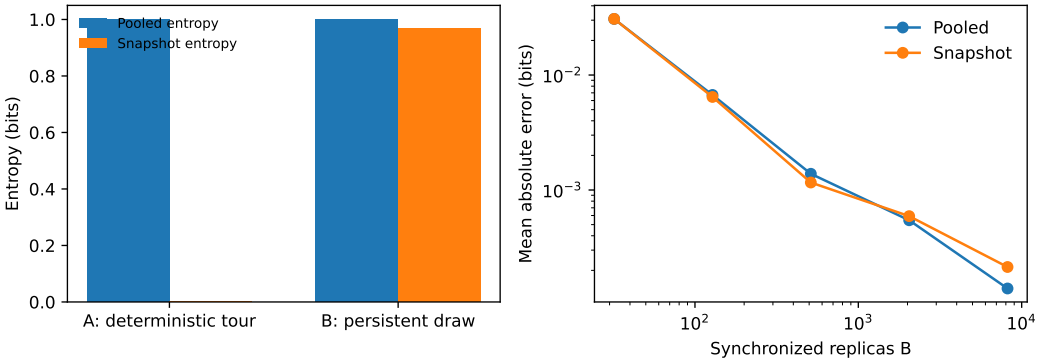


Figure 14: Analytic two-state separation and empirical estimator convergence. The error curve averages 32 independent synchronized batches per size.

ments. Reproduction is per completed seed; the checkpoints contain actors and mixture weights, not mid-update optimizer states.

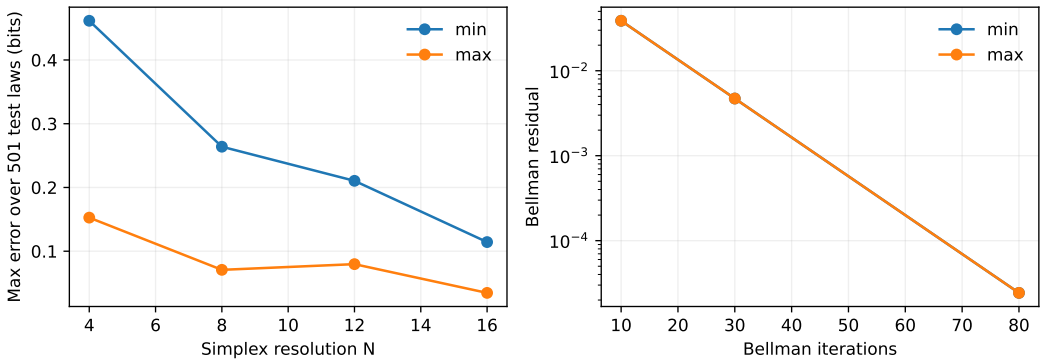


Figure 15: Grid value iteration: interpolation error on fixed test laws and Bellman residual. A small Bellman residual alone does not certify small grid error.

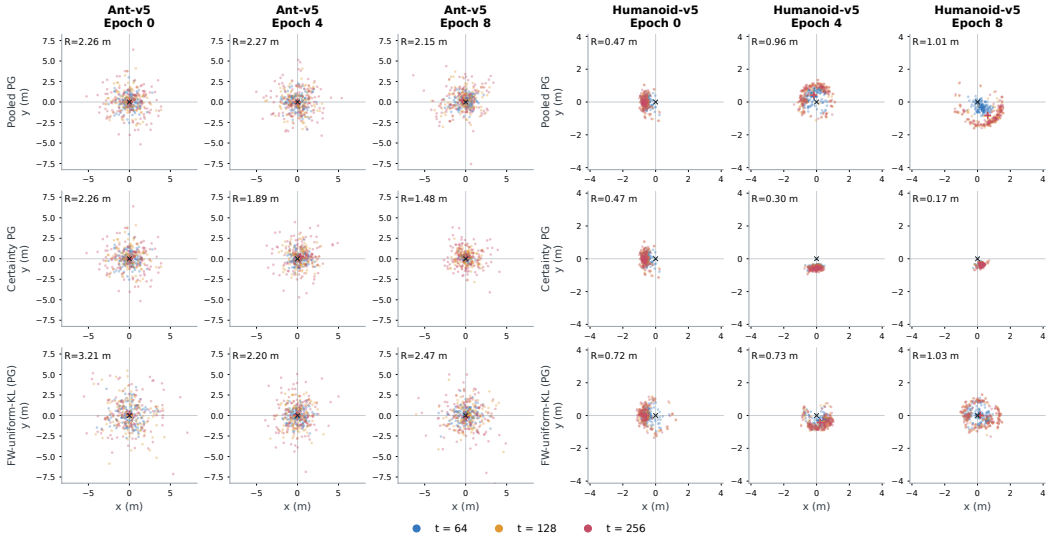


Figure 16: **Dispersion across training epochs.** Each panel overlays 128 torso positions at steps 64 (blue), 128 (orange), and 256 (red); colored crosses mark their means and the black cross marks the reset. Columns compare epochs 0, 4, and 8, corresponding to 0, 262,144, and 524,288 learning interactions. R uses all times, not just the displayed three. Certainty PG contracts the clouds, particularly on Humanoid, but contraction alone does not establish useful exploration. Pooled PG and Certainty PG have identical initialization; the mixture starts from uniform random actions. Axes are fixed within each robot, not shared across robots. All methods and all five checkpoints appear in Section L.

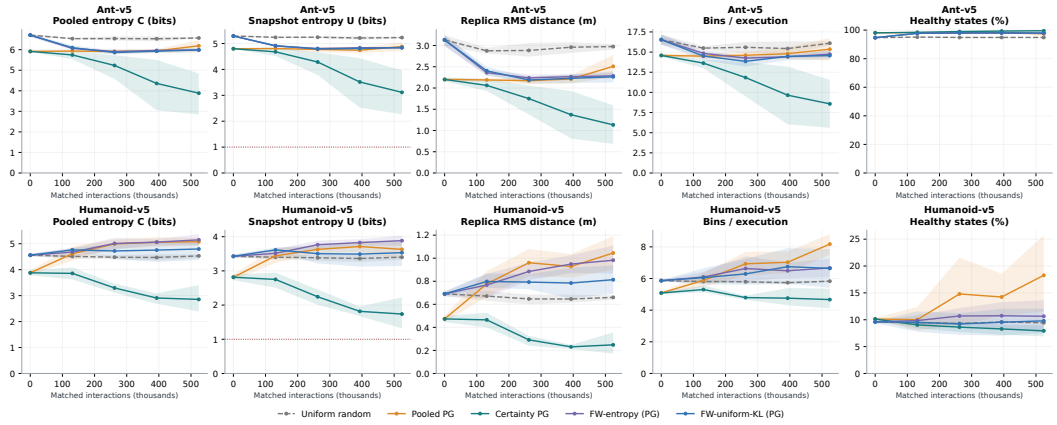


Figure 17: **Continuous-control learning at matched interaction budgets.** Means and 95% seed-bootstrap bands over all five seeds. Each learned method uses the same total environment interactions, including mixture-density batches. Random is a fixed-policy reference drawn at matched markers, not a trained agent. The dotted uncertainty line is the one-bit target. Reduced dispersion under Certainty PG coincides with lower coverage; Humanoid health remains poor.

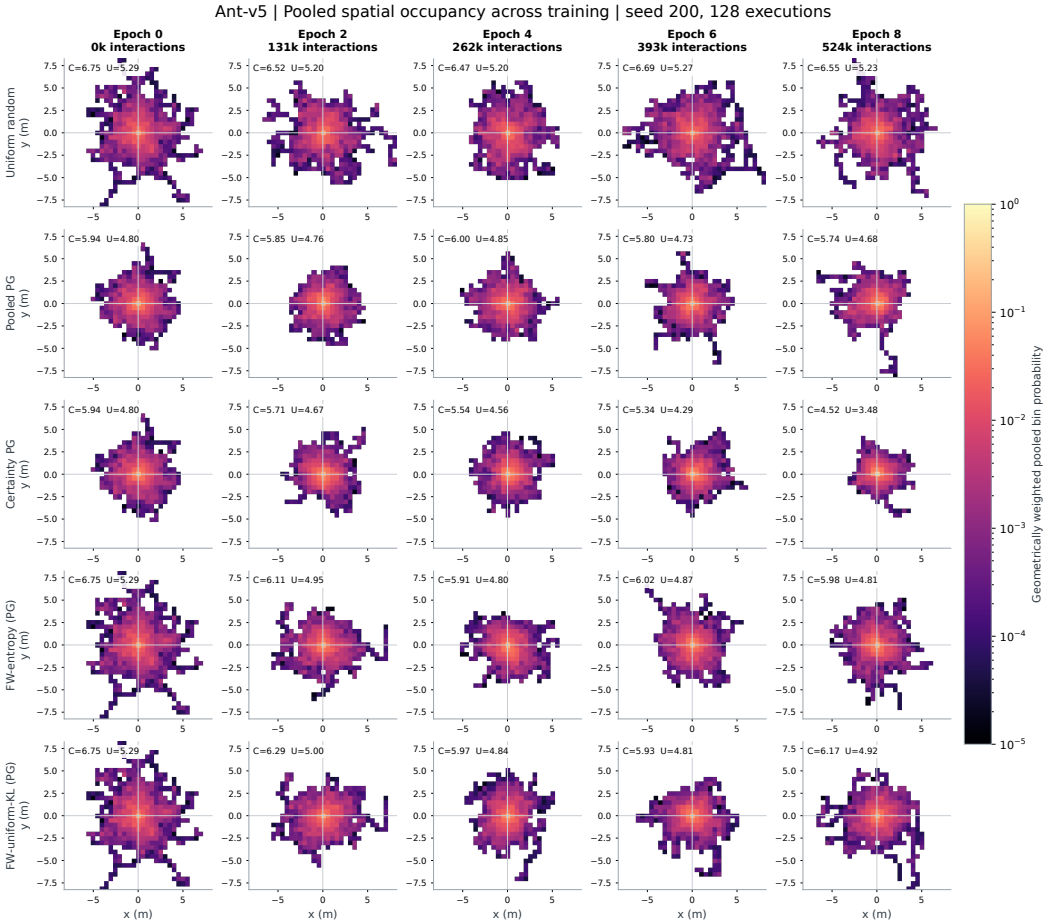


Figure 18: **Ant pooled occupancy across epochs.** Geometrically weighted bin probabilities from 128 independent executions of seed 200. Rows are methods, columns are matched checkpoints. Spatial axes and logarithmic color normalization are fixed. Uniform random is freshly evaluated at each marker without training. Certainty PG's contracting footprint accompanies the loss in distinct bins per execution.

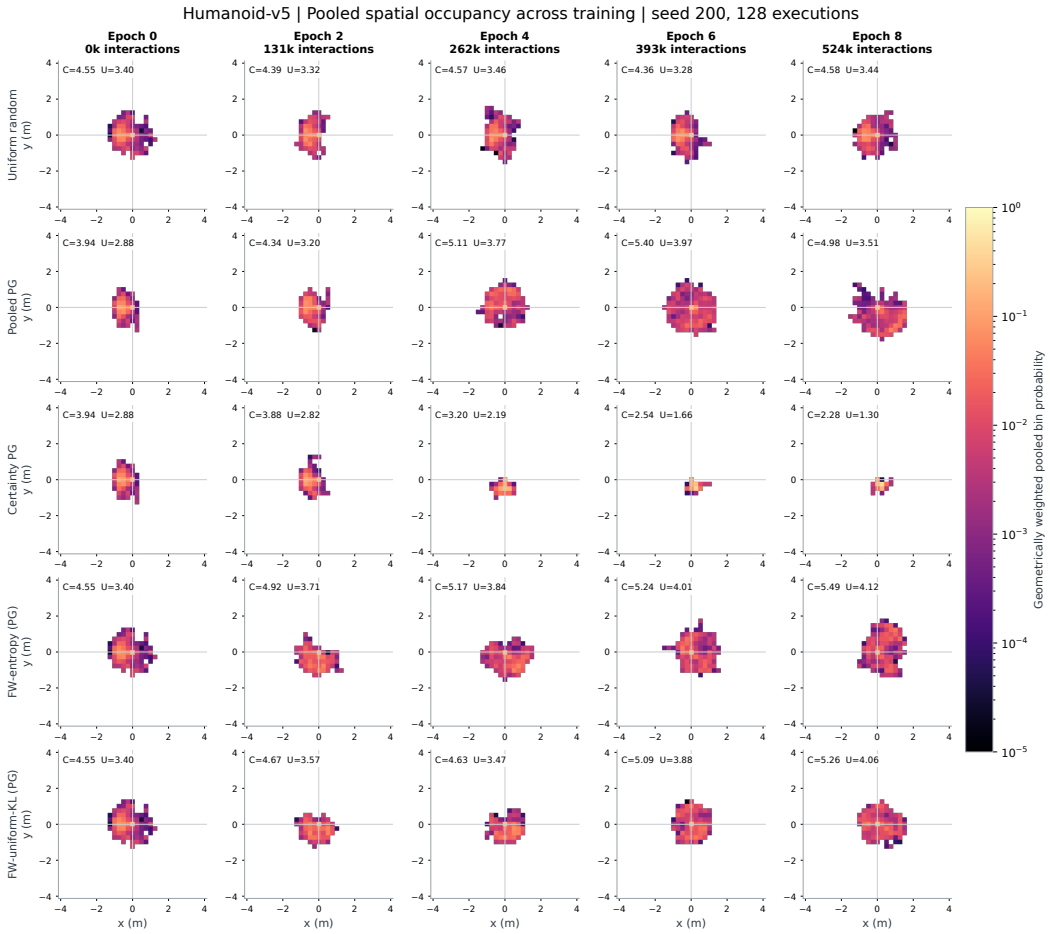


Figure 19: **Humanoid pooled occupancy across epochs.** Same plotting convention and preselected seed as Figure 18, with the fixed Humanoid grid. An expanding footprint is not by itself evidence of locomotion: the height and health records retain falls and post-fall motion. All plotted mass comes from actual saved simulator executions.

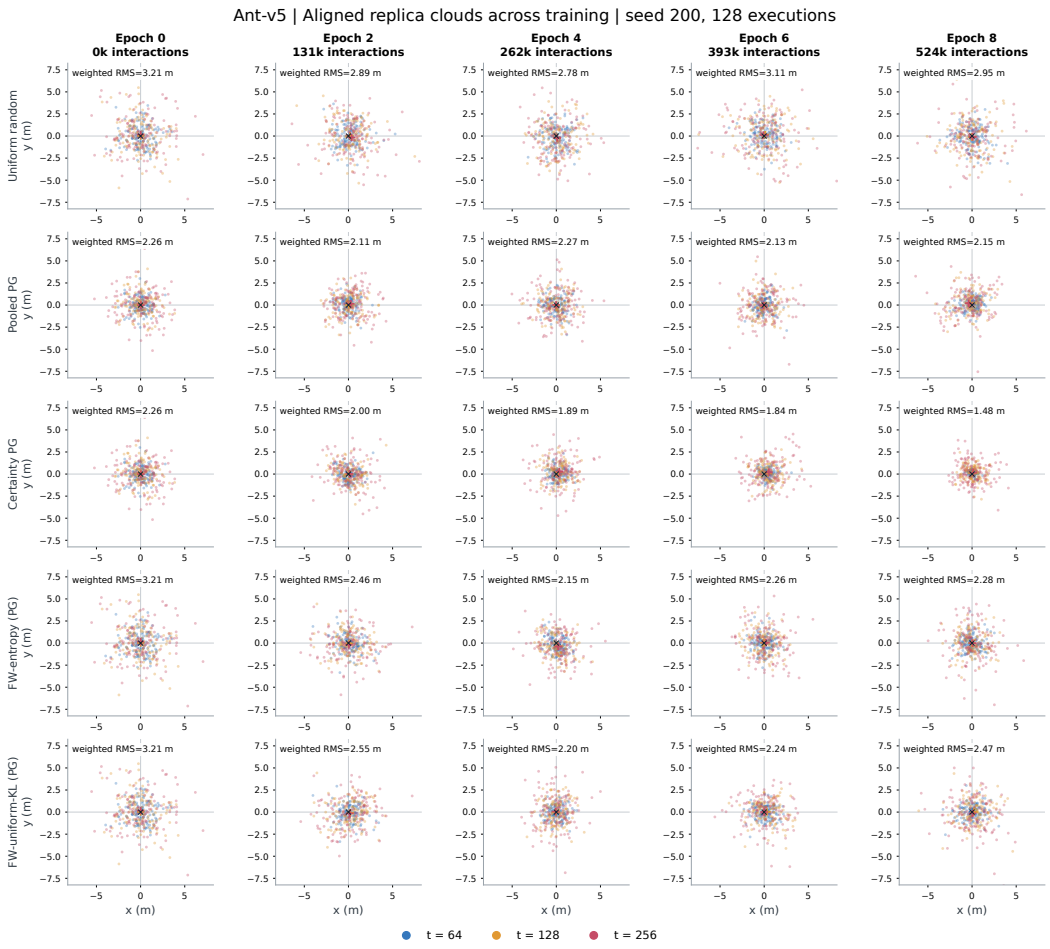


Figure 20: **Ant aligned clouds at all checkpoints.** The three colors denote steps 64, 128, and 256 of independent executions, not training epochs. The annotated RMS distance is weighted over all 257 state times. It averages over component randomness for both mixture comparators. This figure complements pooled occupancy by retaining elapsed time.

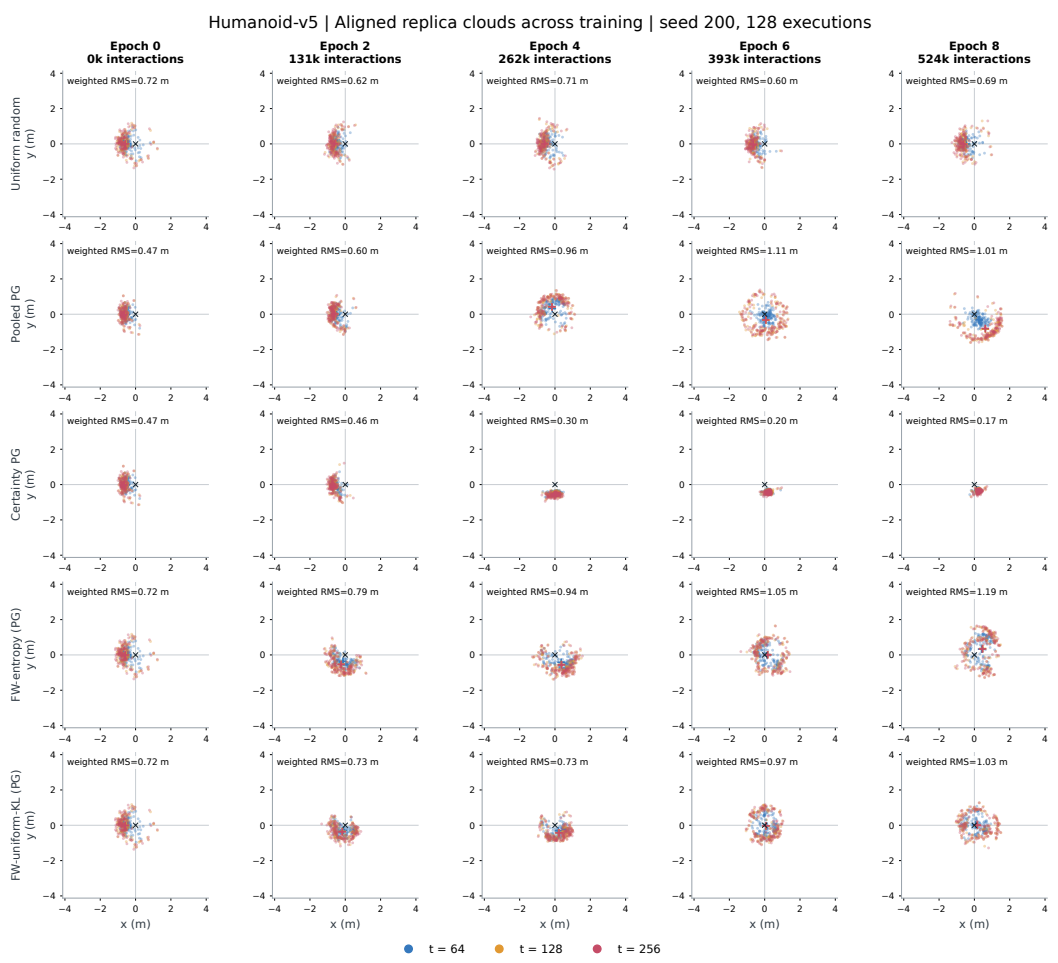


Figure 21: **Humanoid aligned clouds at all checkpoints.** The smaller Certainty PG clouds are spatially repeatable outcomes under this projected metric. They do not establish healthy behavior; the saved posture replay in Figure 26 shows why the distinction matters.

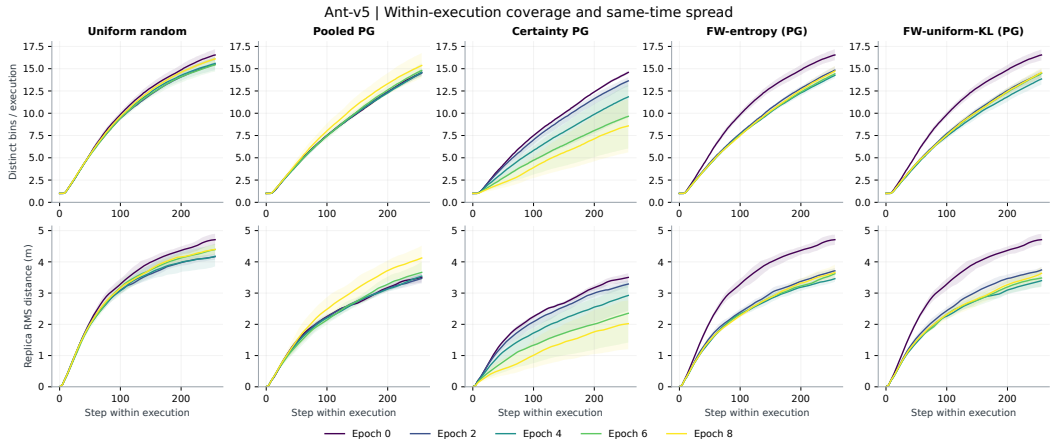


Figure 22: **Ant coverage and dispersion within an execution.** Columns identify methods; colors identify training checkpoints. Top: number of distinct spatial bins visited so far, averaged within each training seed before aggregation. Bottom: same-time RMS separation of replicas. Shading resamples whole training seeds. All methods share vertical limits within each row, including every confidence band.

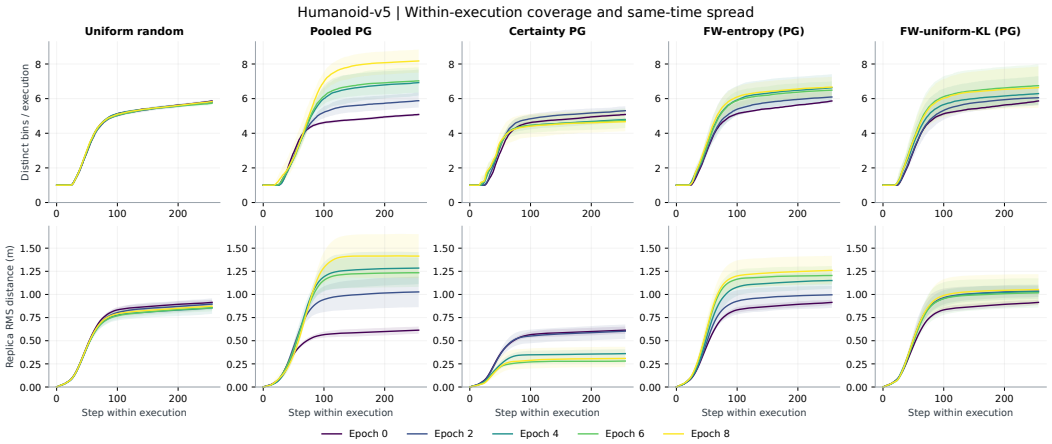


Figure 23: **Humanoid coverage and dispersion within an execution.** Same convention as Figure 22. Pooled PG and the mixture comparators broaden visitation, while Certainty PG contracts replica separation without expanding per-execution coverage. The plateau in the curves should be interpreted alongside the unhealthy outcomes, not as completion of a useful exploration task.

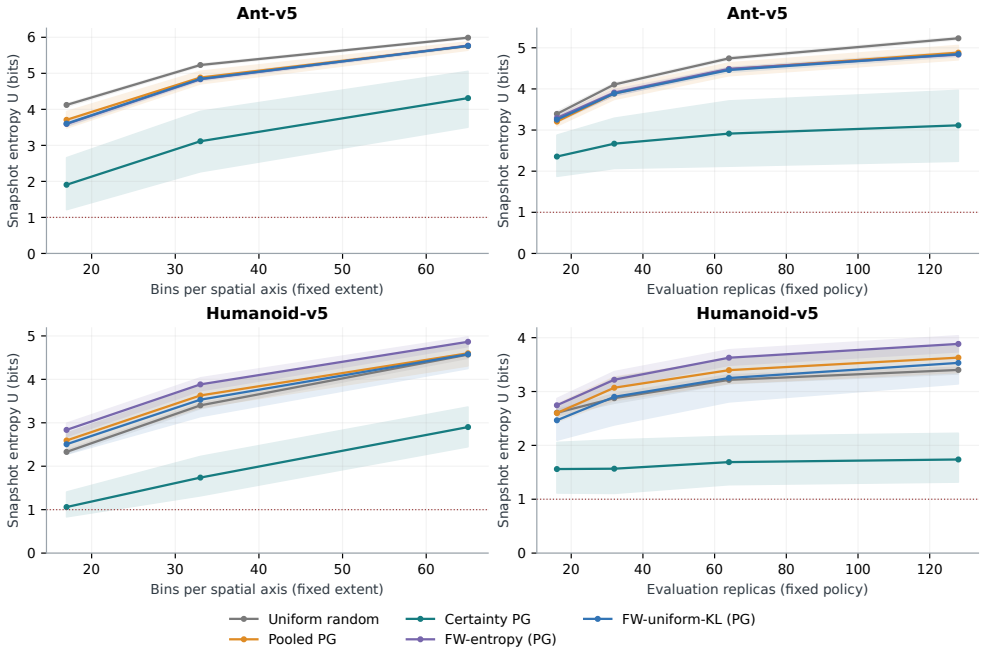


Figure 24: **Representation and estimator sensitivity.** Left: rebinning fixed final positions changes the categorical target. Right: increasing the number of aligned replicas changes the plug-in estimate of the same target. Bands summarize training seeds, not independent repeats of the nested subsample procedure. The one-bit line refers to the original training target; changing bins does not retroactively change that target. Unbinned replica distance supplies a complementary measure insensitive to bin width.

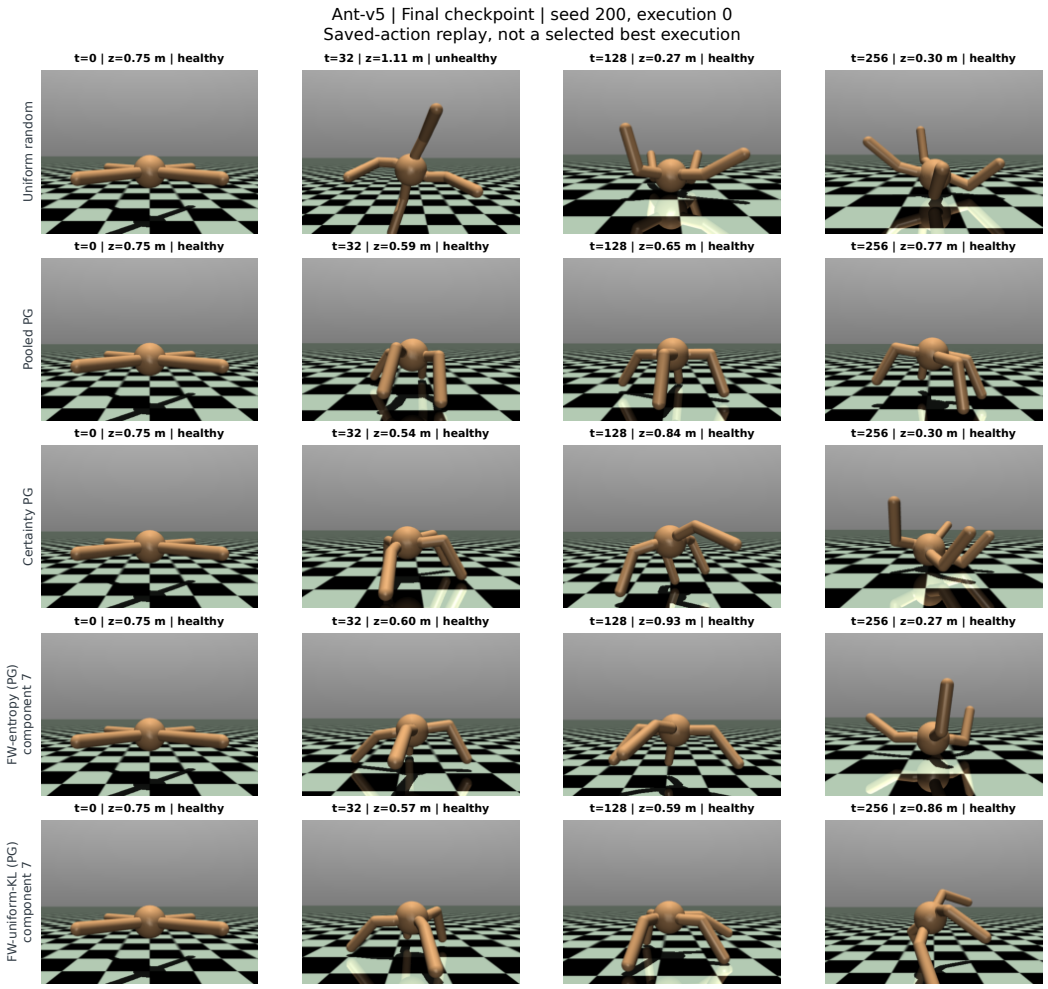


Figure 25: **Ant posture check.** Saved-action replay of execution 0, seed 200, final checkpoint, for every method. Frames are at prespecified steps 0, 32, 128 and 256. The simulator's height-based health indicator does not certify an upright orientation or useful gait. Neither the seed nor execution was selected by reward, coverage, or health.

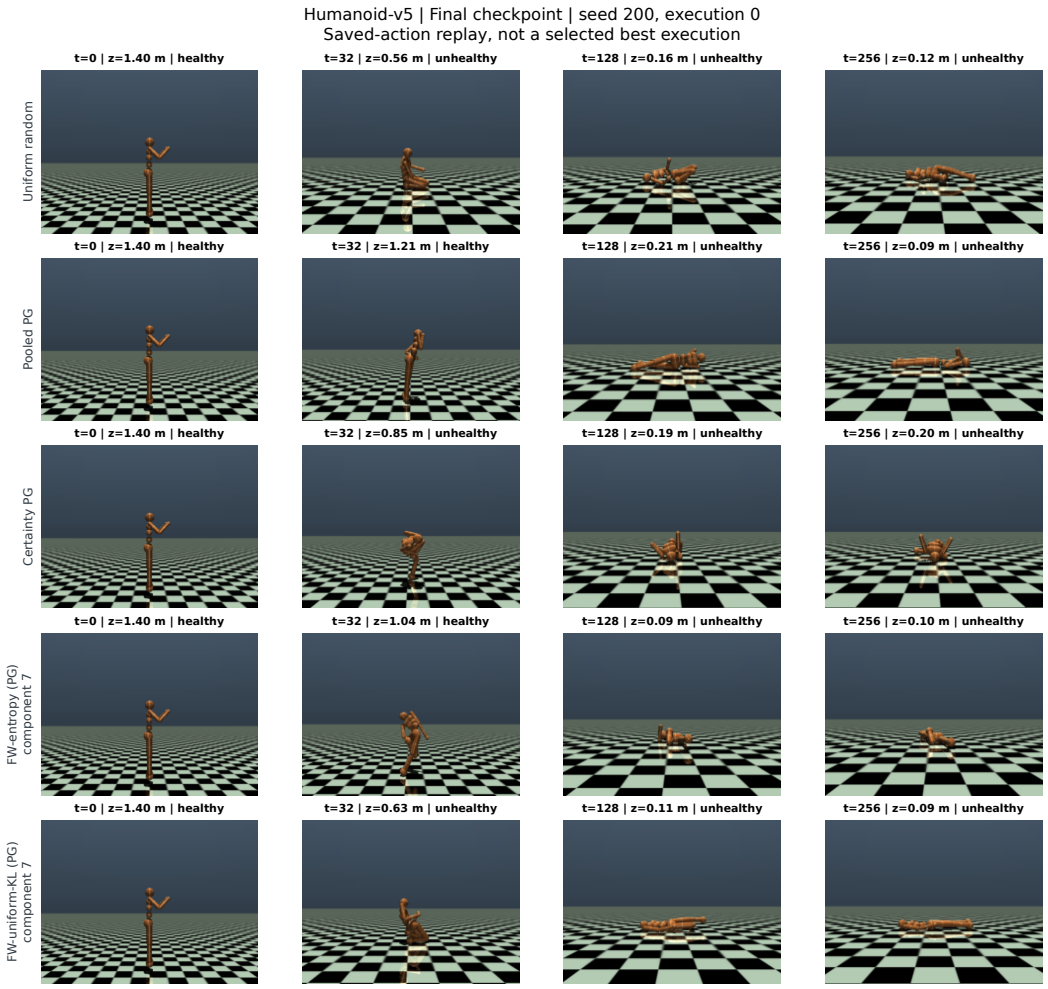


Figure 26: **Humanoid posture check.** The first saved execution of seed 200, final checkpoint, replayed from its recorded reset seed and actions. Positions match the saved trajectory exactly. Every method in this filmstrip falls; all final evaluation batches also have zero fully healthy executions. Tight spatial clouds can therefore arise without healthy locomotion.