CIG: Exploration via Conditional Information Gain

Tim Joseph
FZI
Karlsruhe, Germany
joseph@fzi.de

Marcus Fechner
KIT
Karlsruhe, Germany
marcus.fechner@kit.edu

Philipp Stegmaier
KIT
Karlsruhe, Germany

Karam Daaboul
FZI
Karlsruhe, Germany

J. Marius Zöllner
KIT
Karlsruhe, Germany

Abstract

Intrinsic rewards for exploration in reinforcement learning condition on different contexts: lifelong rewards score each transition against accumulated experience but ignore within-rollout redundancy; episodic rewards penalize intra-trajectory repetition but discard lifelong progress. Hybrid methods combine both signals through heuristic weights or require Gaussian-process dynamics that do not scale beyond low-dimensional state spaces. Trajectory-level information gain decomposes into per-step terms that condition on the lifelong experience and rollout prefix simultaneously, but remains intractable for deep models. We derive the **C**onditional **I**nformation **G**ain (CIG) reward as a tractable surrogate: a log-determinant objective over an ensemble disagreement kernel whose Cholesky factorization yields causal per-step rewards that retain both conditioning sets while scaling to high-dimensional state spaces. We instantiate CIG in a model-based reinforcement learning setting, where rollouts are short and within-rollout corrections remain largely unexplored. Across twelve tasks spanning discrete (MiniGrid) and continuous control (OGBench), in both clean and stochastic-distractor settings, CIG outperforms or matches prior exploration methods while remaining robust to stochastic distractors.

1 Introduction

Reinforcement learning agents in environments with sparse or delayed rewards must explore: they must visit states whose outcomes are not yet predictable from past experience. Intrinsic rewards formalize this pressure by assigning a bonus to transitions that are informative about the environment [1, 2], whether combined with extrinsic reward during task-directed training [3, 4] or used as the sole objective during unsupervised pre-training of a world model [5, 6]. The intrinsic reward controls which transitions the agent collects and therefore which dynamics the world model learns.

In model-based RL the policy is optimised over short imagined rollouts [7–9], and two sources of context are available for scoring a transition within such a rollout (Figure 1): the agent's *lifelong experience*, i.e. the transitions it has executed so far as absorbed by its learned reward model, and the *imagined prefix*, i.e. the steps preceding the transition in the current rollout. Unlike an executed prefix, which enters lifelong experience once the reward model is trained on it, an imagined prefix never does: imagined transitions are not executed, and the reward model is evaluated, not updated, while the policy is optimised. *Lifelong* rewards [3, 4, 10, 5] therefore assign full credit to every imagined transition that probes a model gap, regardless of whether the current rollout has already probed it, so an imagined trajectory can spend every step revisiting the same uncertain transition while neglecting independent transitions elsewhere. Updating the reward model (e.g. with stochastic

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

gradient descent) inside each rollout would remove this redundancy but is impractical, since the policy is trained on thousands of parallel rollouts per update. *Episodic* rewards [11, 12] condition on the current episode instead: they penalise repetition within it but treat a familiar region and a genuinely uncertain one identically whenever both are new within the episode. They were developed for model-free agents whose long executed episodes provide a rich prefix for the redundancy signal; we are not aware of an application to the short imagined rollouts of model-based RL. Existing hybrids either combine the two signals through heuristic weights in the same model-free setting [13, 14] or require Gaussian-process dynamics restricted to low-dimensional state spaces [15].

Trajectory-level information gain provides an alternative that conditions on both the lifelong experience and the prefix simultaneously: its chain-rule decomposition yields per-step rewards in which each term is conditioned on the lifelong experience through the trained ensemble and on the imagined prefix through the preceding steps [16]. Computing these terms, however, requires marginalising over a posterior on model parameters that is intractable for deep networks.

We derive the **Conditional Information Gain (CIG)** reward, a tractable surrogate for this objective. When the posterior over model parameters is approximated by a finite ensemble of one-step predictors, the trajectory-level information gain reduces to a log-determinant over a kernel of pairwise ensemble disagreements (Eq. 6). We identify a rank-saturation bottleneck of the naive surrogate and resolve it through a trace reduction that raises the effective capacity from $M-1$ to $\min(T,\,(M-1)d)$. The Cholesky factorisation of this log-determinant then yields causal per-step rewards: each step's reward equals its total disagreement minus the portion already explained by preceding steps (Eq. 8). CIG plugs into an existing model-based backbone without additional hyperparameters; the only new quantity, the aleatoric scale, is estimated post-hoc from training residuals. Across twelve tasks spanning discrete (MiniGrid) and continuous control (OGBench), in both clean and stochastic-distractor settings, CIG achieves the highest aggregate IQM and is the only method competitive on every task.

We contribute the CIG reward itself (§3), a per-step intrinsic reward derived from a log-determinant surrogate of trajectory-level information gain that conditions on both the lifelong experience and the rollout prefix, together with a rank-saturation analysis that motivates the trace reduction resolving the capacity bottleneck of the finite-ensemble surrogate (§3.2). We evaluate CIG on twelve tasks spanning discrete and continuous-control domains, in both clean and stochastic-distractor variants, using a reward-free protocol that isolates exploration quality from extrinsic reward exploitation; CIG is the only method in the top tier on every task, with an aggregate normalised IQM exceeding the next-best baseline by a clear margin (§4).

2 Background

2.1 Setup

We consider an agent interacting with a reward-free Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \gamma)$, where $\mathcal{S} \subseteq \mathbb{R}^d$ is the state space, $\mathcal{A}$ the action space, $P(s' \mid s, a)$ the unknown transition distribution, and $\gamma \in [0, 1)$ a discount factor. At each step the agent samples $a_t \sim \pi_\theta(\cdot \mid s_t)$, observes $s_{t+1} \sim P(\cdot \mid s_t, a_t)$, and adds the transition to the lifelong experience $\mathcal{D}$ (e.g. by training a model with that transition). The agent's objective is pure exploration: in the absence of any extrinsic reward, it must collect transitions that are maximally informative about the unknown dynamics P. To this end it constructs an intrinsic reward r_t and optimises $J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\big[\sum_{t\geq 0} \gamma^t r_t\big]$.

2.2 Intrinsic Reward Design and Prior Work

The reward r_t must quantify how informative a transition is, yet informativeness depends on context. Two sources of context are available at step t: the lifelong experience $\mathcal{D}$, that determines which dynamics remain uncertain, and the rollout prefix $s_{<t}$, which records what the current trajectory has already probed. In model-based RL, where the policy optimizes over short imagined rollouts ($T \approx 15$), the prefix represents what the agent *plans* to probe rather than what it has already visited in a real episode. Prior intrinsic rewards differ in which of these two contexts they condition on (Figure 1); we summarize the three resulting categories below and provide a comprehensive treatment of related work in Appendix G.

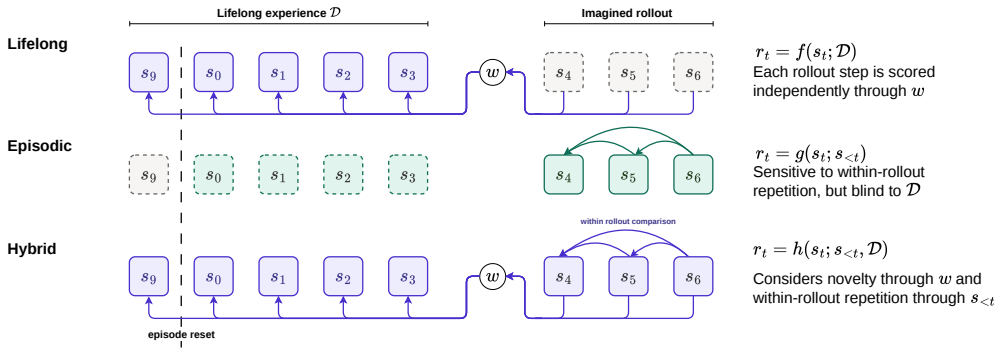


Figure 1: **Conditioning contexts of intrinsic rewards in model-based RL.** The policy plans over short imagined rollouts. Neural network weights w, trained on the lifelong experience $\mathcal{D}$ (e.g. the past transitions), summarize lifelong uncertainty. Solid states and arrows mark the contexts each reward class conditions on; dashed/faded elements are not used as context. In the model-based setting, the rollout prefix $s_{<t}$ is too short to supply a rich episodic signal. CIG (§3) instantiates the hybrid form.

Lifelong rewards, $r_t = f(s_t;\, \mathcal{D})$, score each transition against accumulated experience through prediction error [3, 4], ensemble disagreement [10], or density estimates [17]. Each step is scored independently of all other steps in the rollout, so correlations between disagreement patterns at different time steps are invisible to the reward: correlated steps receive full cumulative credit, a structural redundancy that conditioning on $\mathcal{D}$ alone cannot detect. **Episodic rewards**, $r_t = g(s_t;\, s_{<t})$, penalize within-trajectory repetition [18, 11, 19, 12] but condition on the prefix alone, without access to the lifelong uncertainty encoded in $\mathcal{D}$. These methods were developed for model-free RL, where long real episodes supply a rich prefix for the redundancy signal. In the model-based setting we consider, the prefix is a short imagined rollout and carries far less information, limiting the episodic correction. **Hybrid rewards** target the joint form $r_t = h(s_t;\, s_{<t}, \mathcal{D})$. Several methods multiply or gate an episodic indicator with a lifelong bonus [13, 20, 21, 14], though all set the combination weights heuristically rather than deriving them from a unified objective. Plan2Explore [5] maximizes per-step ensemble disagreement in a model-based setting but scores each step independently, placing it in the lifelong category by conditioning structure despite operating over planned rollouts. TIP [15] derives per-step rewards from a trajectory-level information-gain objective, conditioning on both $\mathcal{D}$ and $s_{<t}$, but relies on Gaussian-process dynamics and is restricted to low-dimensional state spaces. We share TIP's objective applied to the dynamics; the next subsection formalises it, and §3 derives a tractable surrogate that scales to high-dimensional latent spaces.

2.3 Information Gain as a Unifying Objective

A principled objective that conditions on both $\mathcal{D}$ and $s_{<t}$ follows from treating pure exploration as information gain about the dynamics model. Let w parameterise a probabilistic model of the unknown dynamics P, and let $p(w \mid \mathcal{D})$ denote the posterior after observing $\mathcal{D}$. We seek a policy $\pi^\star$ maximising the trajectory-level information gain $I(w;\, s_{1:T} \mid \mathcal{D})$. This objective decomposes via the chain rule of mutual information into per-step terms that condition on both contexts simultaneously:

$$I(w;\, s_{1:T} \mid \mathcal{D}) \;=\; \sum_{t=1}^{T} I\big(w;\, s_t \mid s_{<t},\, \mathcal{D}\big). \tag{1}$$

Each summand scores step t against both the lifelong experience $\mathcal{D}$ and the prefix $s_{<t}$, so the two conditioning contexts that §2.2 identified as complementary appear as joint requirements of a single objective. This sequential decomposition parallels the chain-rule factorisation of expected information gain in Bayesian optimal experimental design [16, 22, 23]. The obstacle is tractability: $p(w \mid \mathcal{D})$ admits no closed form for deep models, and each per-step term requires marginalising over w to obtain the predictive distribution over s_t. The next section derives a tractable surrogate that retains both conditioning sets while scaling beyond the low-dimensional settings to which TIP is confined.

3 Method

We derive the **Conditional Information Gain (CIG)** reward as a tractable surrogate for the per-step information gain $I(w;\, s_t \mid s_{<t},\, \mathcal{D})$ identified in §2.3, preserving its conditioning on both the lifelong experience and the rollout prefix while scaling to high-dimensional latents. Following Plan2Explore [5], we instantiate the dynamics model as an ensemble of one-step predictors over the latent of a learned world model [7–9]; from here on s_t refers to this latent, and all downstream quantities are computed in the latent space. Backbone details are deferred to §4.1.

The derivation has three steps. We first replace the intractable posterior with a finite ensemble, reducing trajectory information gain to a marginal-entropy problem (§3.1). A Gaussian bound on the mixture entropy exposes a rank bottleneck of the finite ensemble, which a trace reduction resolves to yield the CIG surrogate (§3.2). The Cholesky factorisation of the surrogate then decomposes it into causal per-step rewards (§3.3).

3.1 From Information Gain to Marginal Entropy

Two standard approximations make the trajectory information gain tractable. **A1 (ensemble posterior)** replaces the posterior $p(w \mid \mathcal{D})$ with a uniform distribution over M independently trained ensemble members [24, 5], indexed by k, so that the lifelong conditioning is absorbed into training and dropped from the notation. **A2 (Markov-Gaussian dynamics)** factorises each member's trajectory into Gaussian one-step transitions with mean $\mu_k(s_t, a_t)$ and isotropic aleatoric covariance $\sigma^2 I_d$ shared across members. This is the noise model implicit in Plan2Explore's MSE-trained ensemble [5]; adopting it unchanged keeps the ensemble's training identical, so CIG differs from Plan2Explore only in the reward. Under A2, evaluating each member along a fixed imagined trajectory gives a joint predictive that factorises across steps:

$$p(s_{1:T} \mid k) = \mathcal{N}\big(\boldsymbol{\mu}_k,\, \sigma^2 I_{Td}\big), \qquad \boldsymbol{\mu}_k = [\mu_k(s_0, a_0);\, \ldots;\, \mu_k(s_{T-1}, a_{T-1})] \in \mathbb{R}^{Td}. \tag{2}$$

The conditional entropy $H(s_{1:T} \mid k)$ is therefore a constant in the policy, depending only on the aleatoric variance. Since $I(k;\, s_{1:T}) = H(s_{1:T}) - H(s_{1:T} \mid k)$, maximising information gain reduces to maximising the marginal entropy of a Gaussian mixture:

$$\arg\max_{\pi} I(k;\, s_{1:T}) = \arg\max_{\pi} H(s_{1:T}), \qquad p(s_{1:T}) = \frac{1}{M}\sum_{k=1}^{M} \mathcal{N}\big(\boldsymbol{\mu}_k,\, \sigma^2 I_{Td}\big). \tag{3}$$

3.2 Gaussian Surrogate and Capacity Saturation

The mixture entropy in Eq. (3) has no closed form. By approximation **A3 (moment-matched Gaussian)**, we replace the mixture with a Gaussian of matching mean and covariance Σ (see Appendix B). Since the Gaussian maximises entropy among distributions with covariance Σ:

$$H(s_{1:T}) \;\le\; H\big(\mathcal{N}(\,\cdot\,,\, \Sigma)\big) \;=\; \tfrac{1}{2}\log\det(2\pi e\, \Sigma), \qquad \Sigma = C + \sigma^2 I_{Td}. \tag{4}$$

The epistemic part $C \in \mathbb{R}^{Td \times Td}$ has $d \times d$ blocks

$$C_{jt} = \frac{1}{M}\sum_{k=1}^{M} \boldsymbol{\delta}_k^{(j)} \big(\boldsymbol{\delta}_k^{(t)}\big)^{\top}, \qquad \boldsymbol{\delta}_k^{(t)} = \mu_k(s_{t-1}, a_{t-1}) - \bar{\mu}(s_{t-1}, a_{t-1}), \tag{5}$$

collecting the deviations of each member from the ensemble mean $\bar{\mu} = \frac{1}{M}\sum_k \mu_k$.

Maximising this bound directly, which we call the *exact objective*, underperforms in our ablations (§4.5), and Appendix C traces this to a lack of samples. The bound depends on the full covariance of the rollout, yet C is the sample covariance of only M members whose deviations sum to zero, so $\operatorname{rank}(C) \le M - 1$ regardless of the latent dimension d. Once these $M - 1$ directions of disagreement are open, typically within the first step, every further step can only lengthen them, and a step that probes new unknowns no longer systematically earns more than one that repeats the prefix. The exact objective thus loses the redundancy signal that the chain rule of Eq. (1) is meant to supply. The fix is design choice **D1 (trace reduction)**: we collapse each $d \times d$ block of C to its trace, replacing the full epistemic covariance with the kernel

$$K_{jt} = \operatorname{tr}(C_{jt}) = \frac{1}{M}\sum_{k=1}^{M} \big(\boldsymbol{\delta}_k^{(j)}\big)^{\top} \boldsymbol{\delta}_k^{(t)}. \tag{6}$$

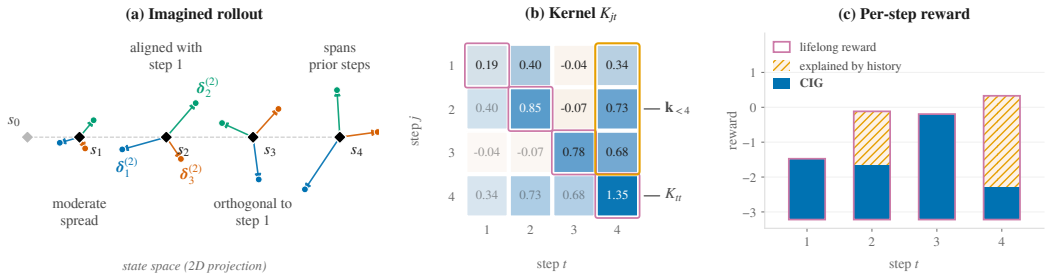


Figure 2: **Mechanism of the CIG reward on a four-step imagined rollout** (T=4, M=3)**. (a)** Disagreement vectors $\boldsymbol{\delta}_k^{(t)}$ at each step. **(b)** The kernel K_{jt} (Eq. 6) built from their inner products; the column $\mathbf{k}_{<4}$ links step 4 to its prefix. **(c)** Eq. 8 subtracts the prefix-explained portion from the lifelong term: step 3, nearly orthogonal to the prefix, retains most of its bonus; step 4, largely spanned by the prefix $s_{<4}$, is sharply discounted.

In each latent coordinate, the members' deviations over the T steps form M samples of a T-dimensional vector, and K adds up their sample covariances over the d coordinates. D1 thus counts the latent coordinates as samples rather than as dimensions, so that K is built from Md samples instead of M. Whereas $\mathrm{rank}(C) \leq M-1$, the kernel satisfies $\mathrm{rank}(K) \leq \min(T, (M-1)d)$, so every step of the rollout can open a direction of its own once $(M-1)d \geq T$ (Appendix D). This comes at the cost of being blind to correlations between different latent coordinates.

Substituting K into the bound and dropping policy-independent constants gives the **CIG surrogate**

$$\tilde{J}(s_{1:T}) = \log\det \tilde{K}, \qquad \tilde{K} \coloneqq K + \sigma^2 d \cdot I_T, \tag{7}$$

in which the ridge $\sigma^2 d$ aggregates the aleatoric contribution across the latent dimensions.

3.3 Per-Step Decomposition: The CIG Reward

The trajectory objective $\tilde{J}$ admits an exact per-step decomposition via the Cholesky factorisation, making it directly compatible with actor-critic training without trajectory-level value heads or high-variance estimators. Writing $\tilde{K} = LL^\top$ gives $\log\det \tilde{K} = 2\sum_t \log L_{tt}$, with each L_{tt}^2 equal to the Schur complement of the leading $(t-1)\times(t-1)$ submatrix. Setting $r_t = 2\log L_{tt}$ recovers $\tilde{J}(s_{1:T}) = \sum_t r_t$ exactly, with r_t depending only on steps $1,\dots,t$:

$$r_t = \log\Big(\underbrace{K_{tt}}_{\text{Lifelong}} + \sigma^2 d \; - \; \underbrace{\mathbf{k}_{<t}^\top \tilde{K}_{<t}^{-1} \mathbf{k}_{<t}}_{\text{Prefix Redundancy}} \Big), \tag{8}$$

where $\mathbf{k}_{<t}$ collects the kernel entries linking step t to its prefix, and $\tilde{K}_{<t}$ is the prefix submatrix. The underlying log-determinant surrogate recovers the form of D-optimal design [25] for deep ensembles.

Eq. (8) composes three mechanisms. **Lifelong disagreement:** the diagonal K_{tt} is the per-step ensemble disagreement, the same signal as Plan2Explore, carrying the lifelong conditioning through the trained ensemble. **Aleatoric compression:** the ridge $\sigma^2 d$ damps small disagreements toward a constant floor $\log(\sigma^2 d)$, concentrating credit on transitions whose disagreement clearly exceeds the predictor's own noise scale; this is a substantive departure from Plan2Explore, whose bonus scales with K_{tt} alone. **Prefix redundancy:** the subtracted quadratic form removes the portion of step-t disagreement already explained by the prefix, recovering the chain-rule conditioning on $s_{<t}$; when the step-t disagreement is orthogonal to the prefix, r_t keeps the full lifelong bonus, and when it lies in the prefix span, r_t falls to the aleatoric floor (Figure 2c; Proposition 3). The two failure modes of §2.2 are addressed symmetrically, with the prefix term preventing the overcounting that lifelong rewards suffer on correlated frontiers and the diagonal preventing the uncertainty-blindness of episodic rewards in familiar-but-uncertain regions. Appendix E.2 characterises this selectivity quantitatively: the correction concentrates on frontier steps where K_{tt} is large, leaves low-disagreement steps untouched, and sharpens as training narrows the frontier.

3.4 Reward Computation

Aleatoric scale. The ensemble is trained by MSE and carries no explicit noise parameter. We estimate σ^2 post-hoc from ensemble-mean residuals on training transitions and substitute the estimate into Eq. (8) at reward-computation time (Appendix A.5). The ensemble training protocol is identical to Plan2Explore [5], so that differences against this baseline isolate the effect of the reward rather than changes to the learned representation.

Complexity. The per-step rewards drive a standard actor-critic on imagined rollouts (§A.2). After the ensemble forward passes, assembling the $T \times T$ kernel K from the deviation vectors costs $\mathcal{O}(T^2Md)$ operations, and the Cholesky factorisation of $\tilde{K}$ costs $\mathcal{O}(T^3)$; the per-step rewards are then read from the diagonal of L. Both costs are negligible relative to the $\mathcal{O}(TMP)$ ensemble forward passes for the rollout length we consider ($T = 15$ and P is the cost of an MLP forward pass). Pseudocode is given in Algorithm 1 (see Appendix A.1).

4 Experiments

We evaluate CIG on twelve tasks across discrete (Minigrid [26]) and continuous-control (OG-Bench [27]) domains, including both clean environments and Noisy-TV variants with action-dependent stochastic distractors, to test three claims: that CIG improves exploration coverage over both lifelong and episodic baselines under matched compute and tuning budget; that the ensemble-grounded reward is robust to distractors that degrade count-based and entropy-based signals; and that each component of the CIG reward, the trace reduction D1, the prefix correction of Eq. (8), and the aleatoric ridge $\hat{\sigma}^2$, contributes to the observed gains. Sections 4.2–4.3 address the first two claims per domain, §4.4 reports aggregate statistics, and §4.5 isolates individual components.

4.1 Experimental Setup

All methods are implemented on top of DreamerV2 [9] so that the intrinsic reward is the only axis of variation. Ensemble and MLP architectures, optimizer, and optimizer hyperparameters are identical across methods; baseline-specific parameters (e.g., ICM forward-model coefficient, E3B ridge scale) use the defaults reported in the original publications without further tuning. CIG introduces no additional hyperparameters beyond those of the shared ensemble: the aleatoric scale $\hat{\sigma}^2$ is estimated post-hoc from ensemble-mean residuals on training transitions (§3.4), not tuned. The tuning budget is therefore identical across CIG and P2E; any performance difference is attributable to the reward computation alone. We compare against six baselines spanning the lifelong/episodic axis of §2.2: P2E [5], RND [4], ICM [3], APT [28], E3B [11] and E3B $\times$ P2E [12] . We evaluate on twelve tasks across two domains: six MiniGrid [26] tasks (three clean, three Noisy-TV) testing procedurally generated combinatorial exploration, and six continuous-control tasks (four clean, two Noisy-TV) testing spatial coverage, multi-object manipulation, and combinatorial configuration coverage. The Noisy-TV (NT) variants augment observations with an action-dependent stochastic distractor. Each (method, task) pair is run with five seeds; we report the inter-quartile mean (IQM) with 95 % stratified bootstrap confidence intervals [29]. Full hyperparameter tables, environment details, baseline implementations, and compute budget are given in Appendix A.

4.2 MiniGrid

MiniGrid [26] provides procedurally generated gridworlds in which layouts and object positions are resampled every episode, and solving a task requires interacting with objects in a specific order. We evaluate on three distinct tasks (MultiRoom N7S8, KeyCorridor S4R3, and ObstructedMaze 2DlhB), each in a clean and a Noisy-TV variant, and report cumulative successes over training. Environment details are in Appendix A.4.1.

Clean variants. Figure 3 (top row) shows cumulative successes on the three clean tasks. CIG and P2E share the same ensemble, so the gap between them is attributable to the reward computation alone. On MultiRoom N7S8, CIG accumulates an order of magnitude more successes than P2E by the end of training, and its success rate continues to accelerate where P2E plateaus. On KeyCorridor S4R3, CIG and APT accumulate successes at a comparable rate throughout training, with P2E trailing

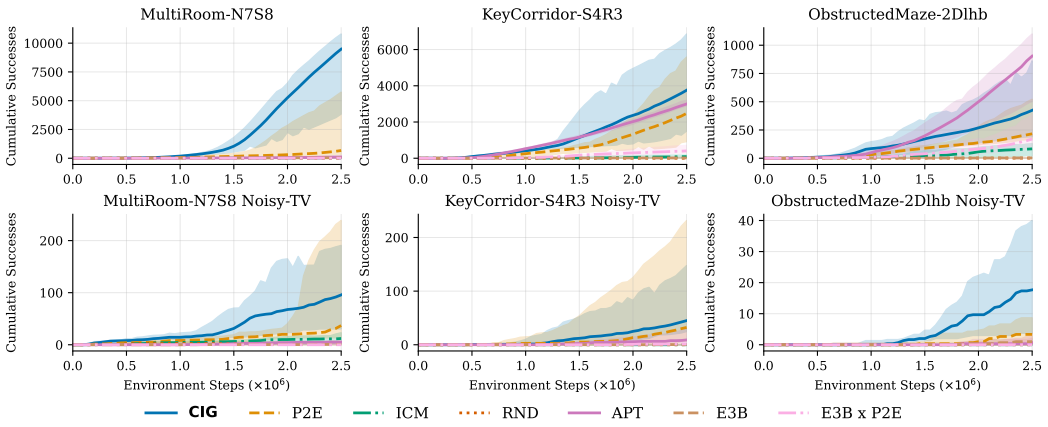


Figure 3: Cumulative successes (total completed episodes) on three MiniGrid tasks (columns) in clean (top) and Noisy-TV (bottom) variants. Lines show the IQM across five seeds; shaded regions are 95 % stratified bootstrap confidence intervals. Under Noisy-TV distractors, non-ensemble methods drop to near-zero success rates while CIG and P2E accumulate successes throughout training.

both but continuing to improve; the three methods' confidence intervals overlap toward the end of the budget. On ObstructedMaze 2DlhB, APT accumulates roughly twice the successes of CIG: its state-entropy proxy produces high intra-episode state diversity, which is effective when uniform coverage aligns with the task structure. A per-method entropy analysis (Appendix F.2) shows that high visit-distribution entropy alone does not predict downstream performance: APT and E3B achieve the highest entropy across tasks, yet only APT on ObstructedMaze translates this into a clear advantage over other baselines, while CIG outperforms with lower entropy on the other two tasks, pointing to targeted uncertainty reduction as the driving factor. E3B $\times$ P2E trails standalone P2E on MultiRoom and KeyCorridor, indicating that naïvely multiplying an episodic indicator with a lifelong bonus can suppress the lifelong signal rather than refine it.

Noisy-TV variants. Ensemble disagreement vanishes for irreducible stochastic transitions, so CIG and P2E are largely robust to the distractor by construction. RND and APT score every novel observation regardless of source; ICM discards action-independent variation but not the action-dependent noise this variant introduces. Figure 3 (bottom row) confirms these predictions: RND, ICM, APT, E3B, and E3B $\times$ P2E drop to near-zero successes on all three tasks. APT's clean-task advantage on ObstructedMaze does not survive the distractor. Only CIG and P2E retain non-trivial success counts, with CIG outpacing P2E on every task. Because both methods already filter irreducible noise through the ensemble, this persistent gap reflects the additional structure CIG's reward formulation captures beyond per-step disagreement.

4.3 Continuous Control

We evaluate on four continuous-control tasks that pose complementary exploration challenges: spatial coverage under quadrupedal locomotion (AntMaze), where exploration is bottlenecked by learning an 8-DoF gait before coverage can accumulate; continuous multi-object configuration coverage (Cube-Triple); coverage of a combinatorial discrete configuration space where each button press toggles itself and its neighbours (Puzzle 3×3); and coverage over a heterogeneous set of manipulation objects (Scene). AntMaze and Puzzle 3×3 additionally admit Noisy-TV variants. Task-specific coverage metrics are defined in Appendix A.4.2.

Clean variants. Figure 4 (top row and bottom right) shows coverage on the four clean tasks. On AntMaze, CIG saturates coverage first; E3B $\times$ P2E follows closely, and P2E and ICM trail with overlapping confidence intervals. APT, E3B, and RND plateau early at low coverage, never progressing beyond the initial region of the maze. On Cube-Triple, CIG opens the widest margin over all baselines: P2E is the nearest competitor, and E3B $\times$ P2E falls below standalone P2E. Both AntMaze and Cube-Triple exhibit high within-rollout redundancy (consecutive steps often probe

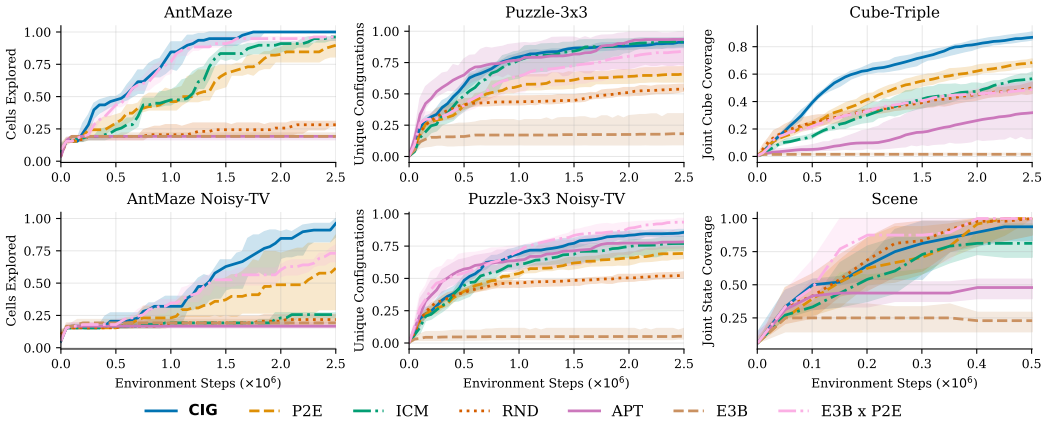


Figure 4: Coverage curves on six continuous-control tasks: four clean (top row and bottom right) and two Noisy-TV (bottom left). Lines show the IQM across five seeds; shaded regions are 95 % stratified bootstrap confidence intervals. CIG is the only method that remains competitive across all six panels; other methods match or exceed CIG on individual tasks but degrade on at least one.

the same locomotion gap or move the robot instead of the cubes), and CIG's gains over P2E are largest in these settings. On Puzzle 3×3, CIG, APT, and ICM reach comparable final coverage with overlapping confidence intervals; E3B × P2E is close behind, and P2E trails all four. Each button press tends to yield a distinct configuration and the buttons must be pressed in the right combination to reach unexplored configurations. APT reaches similar coverage through a different mechanism: it accumulates roughly an order of magnitude more button flips than CIG (Appendix F.3), relying on interaction volume rather than selectivity. On Scene, the small configuration space is covered quickly by most methods. P2E and RND reach full coverage before CIG, bounding CIG's advantage to tasks where within-rollout correlation is non-trivial; when the exploration problem is already easy, CIG's more structured reward provides no additional benefit.

Noisy-TV variants. Figure 4 (bottom left two panels) extends the distractor analysis of §4.2 to continuous control. On AntMaze Noisy-TV, CIG retains nearly all of its clean-task coverage; P2E and E3B × P2E both degrade but remain above the plateau of RND, ICM, APT, and E3B, which stall at or near their initial coverage. CIG's lead over P2E on this panel mirrors the MiniGrid pattern: both methods filter the distractor through the ensemble, but CIG's reward formulation extracts a stronger signal from the remaining epistemic uncertainty. E3B × P2E degrades more from its clean-task level than CIG or P2E do, but it does not collapse; its performance falls between P2E and CIG. On Puzzle 3×3 Noisy-TV, the distractor has a smaller effect than on other tasks: CIG, P2E, and RND retain most of their clean-task coverage, while APT and ICM degrade moderately. E3B × P2E is the exception in both directions: it is the only method whose coverage increases relative to its clean variant, reaching the highest final coverage of any method on this panel. We do not have a clear explanation for this improvement; it is the only panel on which E3B × P2E surpasses CIG. CIG's coverage drops modestly between variants, a smaller relative degradation than APT or ICM, confirming that CIG's mechanisms compose robustly without task-specific calibration.

4.4 Aggregate analysis

The per-domain results of §4.2–4.3 show that CIG outpaces baselines on most individual tasks; we now ask whether this advantage is consistent across the full suite. Figure 5 (left) reports normalised scores across all twelve tasks. CIG leads on all three statistics with non-overlapping IQM confidence intervals against every other method. CIG's median and IQM are closely aligned, confirming that the advantage is not driven by a few high-scoring tasks; its lower mean reflects the tasks where CIG does not lead (ObstructedMaze clean, Puzzle 3×3, Scene; §4.2–4.3). P2E's three statistics cluster tightly, reflecting moderate but uniform performance. APT's mean exceeds its median, consistent with strong wins on a subset of tasks that do not transfer across the suite. E3B × P2E falls below standalone P2E in aggregate, confirming that heuristic combination does not compensate for the hybrid's failure

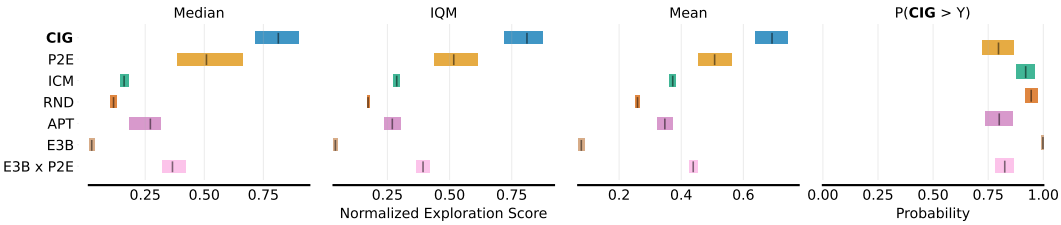


Figure 5: Aggregate normalised exploration scores across all twelve tasks (six MiniGrid, six continuous control). Per-task scores are normalised to $[0, 1]$ by the best score on that task. Left: median, IQM, and mean with 95 % stratified bootstrap confidence intervals over five seeds per (method, task) pair [29]. Right: probability that CIG outperforms each baseline on a randomly sampled task. CIG leads on all three aggregate statistics; its IQM confidence interval does not overlap with that of any other baseline.

modes on Cube-Triple and AntMaze Noisy-TV. E3B, the only purely episodic baseline, ranks last in aggregate across all three statistics, consistent with the short-rollout limitation identified in §2.2: with $T \approx 15$ imagined steps the prefix carries too little information for the episodic correction to produce a useful signal. Figure 5 (right) reports the probability of improvement [29], the probability that CIG outscores a given baseline on a uniformly sampled task (0.5 = parity). All values exceed 0.79 with CIs bounded away from 0.5, confirming that CIG's advantage holds at the level of individual tasks.

4.5 Ablations

We ablate CIG at two levels. On AntMaze (Figure 6), we remove each component in turn: the trace reduction D1, which changes the surrogate itself, and the two terms of the reward in Eq. (8), the prefix correction and the aleatoric ridge. Across the suite (Table 1), we then test whether the contributions of the two reward terms carry over to other tasks.

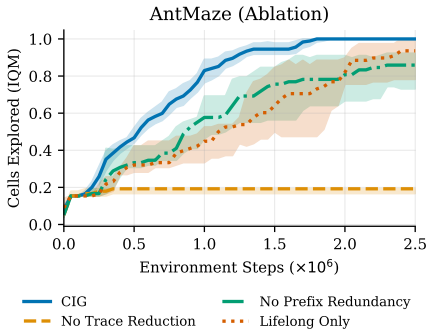


Figure 6: Ablation on AntMaze. Variants remove the trace reduction D1, the prefix term of Eq. (8), or both reward terms (*Lifelong Only*). Lines show the IQM; shaded regions are 95 % stratified bootstrap confidence intervals.

Component analysis on AntMaze. Removing the trace reduction (*No Trace Reduction*) is the most severe intervention: coverage flatlines early in training, with CIs that separate from all other variants and never recover. This confirms the rank-$(M-1)$ capacity saturation derived in §3.2: the surrogate's limited rank renders the reward insensitive to meaningful variation across steps, and the trace reduction restores sufficient fidelity for the policy to distinguish explored from unexplored regions. The bound behind this saturation, $\mathrm{rank}(C) \leq M - 1$, depends only on the ensemble size and not on the task (Appendix C). The prefix correction is the main source of sample efficiency on AntMaze. Both variants that remove it learn steadily but converge more slowly, with CIG's lower CI above both upper CIs through mid-training: where consecutive steps revisit the same locomotion frontier, the prefix correction concentrates credit on steps that probe new directions in disagreement space. The ridge is not separable on this task: *No Prefix Redundancy* and *Lifelong Only*, which differ only by the ridge, have overlapping CIs at every checkpoint.

Reward terms across the suite. Table 1 extends the two reward-term ablations to all six continuous-control tasks and five MiniGrid tasks. Removing the prefix correction lowers the IQM by a similar relative margin in both domains (17 % in continuous control, 18 % in MiniGrid). In continuous control the CIs separate clearly, consistent with the high within-rollout redundancy of AntMaze and Cube-Triple (§4.3); in MiniGrid, the wider CIs leave them overlapping. The ridge, not separable on AntMaze alone, has a strongly domain-dependent effect: removing it costs little in continuous control, where the IQM CIs of *No Prefix Redundancy* and *Lifelong Only* overlap, but cuts the MiniGrid IQM

Table 1: Reward-term ablations across the suite. Normalised scores over five seeds per (variant, task) pair with stratified bootstrap CIs [29], computed separately over the six continuous-control tasks and five MiniGrid tasks (all except KeyCorridor S4R3 clean). *No Prefix Redundancy* drops the subtracted term of Eq. (8); *Lifelong Only* additionally drops the ridge, so each row removes one further term. Per-task normalisation follows Figure 5. Removing either reward term lowers the IQM in both domains; the prefix correction matters most in continuous control, the ridge in MiniGrid.

Variant	Continuous control IQM [95 % CI]	MiniGrid IQM [95 % CI]
CIG	**0.972** [0.953, 0.987]	**0.472** [0.259, 0.705]
No Prefix Redundancy	0.812 [0.778, 0.844]	0.385 [0.269, 0.515]
Lifelong Only	0.769 [0.740, 0.799]	0.104 [0.062, 0.203]

by nearly three quarters, with non-overlapping CIs. This is consistent with the role of the ridge in Eq. (8), which damps small disagreements toward the floor $\log(\sigma^2 d)$ instead of letting the logarithm amplify differences among them. The two terms are therefore complementary: the prefix correction accounts for most of the gap between CIG and *Lifelong Only* in continuous control, the ridge for most of it in MiniGrid, and the full reward ranks first on every statistic in both domains (median and mean in Appendix Table 5).

5 Conclusion

Lifelong and episodic exploration rewards each discard half of the available context: the former ignores what has already been observed within a rollout, while the latter forgets what was learned across rollouts. Prior attempts to combine both signals have relied on heuristic products or Gaussian-process dynamics that do not scale. CIG shows that the two conditioning sets need not be stitched together post hoc: they emerge jointly from a single log-determinant surrogate of trajectory-level information gain, whose Cholesky factorization produces causal per-step rewards. The practical consequence is robustness across problem structure. On tasks where count- and entropy-based baselines collapse (all Noisy-TV variants), the shared ensemble posterior filters irreducible noise; on tasks where most methods already perform well (Puzzle 3×3, Scene), CIG reaches comparable coverage. Across the full twelve-task suite, CIG achieves the highest aggregate IQM with non-overlapping confidence intervals against every baseline (§4.4). No other method avoids degradation across all conditions we tested.

Limitations and future work. CIG does not dominate on every task: APT outperforms it on ObstructedMaze (clean), and E3B$\times$P2E surpasses it on Puzzle 3×3 Noisy-TV for reasons we do not yet fully understand. Characterising the problem structure under which CIG's surrogate is loose enough for simpler proxies to win is an open question. The evaluation is restricted to model-based RL with short imagined rollouts (DreamerV2, $T \approx 15$). In model-free settings with episodes spanning hundreds of steps, the $\mathcal{O}(T^3)$ Cholesky cost and $T \times T$ kernel storage may become prohibitive, and the posterior signal from a small ensemble ($M = 5$) would compound errors over longer horizons; sliding-window or low-rank kernel approximations are a natural next step. More broadly, the trace reduction discards directional structure in the epistemic covariance; retaining partial directional information without reintroducing the rank-saturation bottleneck of §3.2 remains open.

Acknowledgments and Disclosure of Funding

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project "NXT GEN AI METHODS – Generative Methoden für Perzeption, Prädiktion und Planung". The authors would like to thank the consortium for the successful cooperation.

The authors declare no competing interests.

References

[1] Jürgen Schmidhuber. Curious model-building control systems. In *Proceedings of the 1991 IEEE International Joint Conference on Neural Networks (IJCNN '91), Singapore*, volume 2, pages 1458–1463. IEEE, 1991. doi: 10.1109/IJCNN.1991.170605.

[2] Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics*, Volume 1 - 2007, 2007. ISSN 1662-5218. doi: 10.3389/neuro.12.006.2007.

[3] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-Driven Exploration by Self-Supervised Prediction. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 2778–2787, Sydney, NSW, Australia, 2017. JMLR.org.

[4] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

[5] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to Explore via Self-Supervised World Models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8583–8592. PMLR, July 2020.

[6] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, April 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08744-2.

[7] David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[8] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565. PMLR, June 2019.

[9] Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations*, 2021.

[10] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-Supervised Exploration via Disagreement. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5062–5071. PMLR, June 2019.

[11] Mikael Henaff, Roberta Raileanu, Minqi Jiang, and Tim Rocktäschel. Exploration via Elliptical Episodic Bonuses. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[12] Mikael Henaff, Minqi Jiang, and Roberta Raileanu. A Study of Global and Episodic Bonuses for Exploration in Contextual MDPs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 12972–12999. PMLR, July 2023.

[13] Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andew Bolt, and Charles Blundell. Never Give Up: Learning Directed Exploration Strategies, February 2020.

[14] Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E. Gonzalez, and Yuandong Tian. NovelD: A Simple yet Effective Exploration Criterion. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[15] Viraj Mehta, Ian Char, Joseph Abbate, Rory Conlin, Mark Boyer, Stefano Ermon, Jeff Schneider, and Willie Neiswanger. Exploration via Planning for Information about the Optimal Trajectory. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 28761–28775. Curran Associates, Inc., 2022.

[16] D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956. doi: 10.1214/aoms/1177728069.

[17] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying Count-Based Exploration and Intrinsic Motivation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[18] Nikolay Savinov, Anton Raichuk, Damien Vincent, Raphael Marinier, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic Curiosity through Reachability. In *International Conference on Learning Representations*, 2019.

[19] Daejin Jo, Sungwoong Kim, Daniel Nam, Taehwan Kwon, Seungeun Rho, Jongmin Kim, and Donghoon Lee. LECO: Learnable Episodic Count for Task-Specific Intrinsic Reward. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30432–30445. Curran Associates, Inc., 2022.

[20] Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the Atari Human Benchmark. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 507–517. PMLR, July 2020.

[21] Roberta Raileanu and Tim Rocktäschel. RIDE: Rewarding Impact-Driven Exploration for Procedurally-Generated Environments. In *International Conference on Learning Representations*, 2020.

[22] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995. doi: 10.1214/ss/1177009939.

[23] Tom Rainforth, Adam Foster, Desi R. Ivanova, and Freddie Bickford Smith. Modern bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024. doi: 10.1214/23-STS915.

[24] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[25] Friedrich Pukelsheim. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics, 2006. doi: 10.1137/1.9780898719109.

[26] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo De Lazcano Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and J K Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[27] Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking Offline Goal-Conditioned RL. In *International Conference on Learning Representations (ICLR)*, 2025.

[28] Hao Liu and Pieter Abbeel. Behavior From the Void: Unsupervised Active Pre-Training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[29] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep Reinforcement Learning at the Edge of the Statistical Precipice. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29304–29320. Curran Associates, Inc., 2021.

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR (Poster)*, 2015.

[31] Georg Ostrovski, Marc G. Bellemare, Aäron van den Oord, and Rémi Munos. Count-Based Exploration with Neural Density Models. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2721–2730. PMLR, August 2017.

[32] Sam Lobel, Akhil Bagaria, and George Konidaris. Flipping Coins to Estimate Pseudocounts for Exploration in Reinforcement Learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22594–22613. PMLR, July 2023.

[33] Kai Yang, Jian Tao, Jiafei Lyu, and Xiu Li. Exploration and Anti-Exploration with Distributional Random Network Distillation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 56397–56421. PMLR, July 2024.

[34] Augustine Mavor-Parker, Kimberly Young, Caswell Barry, and Lewis Griffin. How to Stay Curious while avoiding Noisy TVs using Aleatoric Uncertainty Estimation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15220–15240. PMLR, July 2022.

[35] Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State Entropy Maximization with Random Encoders for Efficient Exploration. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9443–9454. PMLR, July 2021.

[36] Yiming Wang, Kaiyan Zhao, Furui Liu, and Leong Hou U. Rethinking Exploration in Reinforcement Learning with Effective Metric-Based Exploration Bonus. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[37] Rein Houthooft, Xi Chen, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. VIME: Variational Information Maximizing Exploration. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[38] Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-Based Active Exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5779–5788. PMLR, June 2019.

[39] Pietro Mazzaglia, Ozan Catal, Tim Verbelen, and Bart Dhoedt. Curiosity-Driven Exploration via Latent Bayesian Surprise. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36 (7):7752–7760, June 2022. doi: 10.1609/aaai.v36i7.20743.

[40] Yuhua Jiang, Qihan Liu, Yiqin Yang, Xiaoteng Ma, Dianyu Zhong, Hao Hu, Jun Yang, Bin Liang, Bo XU, Chongjie Zhang, and Qianchuan Zhao. Episodic Novelty Through Temporal Distance. In *The Thirteenth International Conference on Learning Representations*, 2025.

[41] Yao Fu, Run Peng, and Honglak Lee. Go Beyond Imagination: Maximizing Episodic Reachability with World Models, 2023.

[42] Daochen Zha, Wenye Ma, Lei Yuan, Xia Hu, and Ji Liu. Rank the Episodes: A Simple Approach for Exploration in Procedurally-Generated Environments. In *International Conference on Learning Representations*, 2021.

[43] Tianjun Zhang, Paria Rashidinejad, Jiantao Jiao, Yuandong Tian, Joseph E. Gonzalez, and Stuart Russell. MADE: Exploration via Maximizing Deviation from Explored Regions. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[44] Yiding Jiang, J. Zico Kolter, and Roberta Raileanu. On the Importance of Exploration for Generalization in Reinforcement Learning. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

[45] Alen Alexanderian. A brief note on the Bayesian D-optimality criterion, 2023.

[46] Viraj Mehta, Biswajit Paria, Jeff Schneider, Willie Neiswanger, and Stefano Ermon. An experimental design perspective on model-based reinforcement learning. In *International Conference on Learning Representations*, 2022.

[47] Alberto Caron, Vasilios Mavroudis, and Chris Hicks. On efficient bayesian exploration in model-based reinforcement learning. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Algorithm 1 Conditional Information Gain reward (Algorithm referenced in §3.4).

Require: Imagined latent rollout $(s_0, a_0, \ldots, s_{T-1}, a_{T-1})$; ensemble of one-step predictors $\{\mu_k\}_{k=1}^{M}$; aleatoric estimate $\hat{\sigma}^2$; latent dimension d.
Ensure: Per-step intrinsic rewards $r_{1:T}$.
1: **for** $t = 1, \ldots, T$ **do**
2: **for** $k = 1, \ldots, M$ **do**
3: $m_k^{(t)} \leftarrow \mu_k(s_{t-1}, a_{t-1})$ ▷ ensemble forward pass (A2)
4: **end for**
5: $\bar{m}^{(t)} \leftarrow \frac{1}{M} \sum_{k=1}^{M} m_k^{(t)}$
6: $\boldsymbol{\delta}_k^{(t)} \leftarrow m_k^{(t)} - \bar{m}^{(t)}$ for $k = 1, \ldots, M$ ▷ centred deviations, Eq. (5)
7: **end for**
8: $K_{jt} \leftarrow \frac{1}{M} \sum_{k=1}^{M} \big(\boldsymbol{\delta}_k^{(j)}\big)^{\top} \boldsymbol{\delta}_k^{(t)}$ for $1 \le j, t \le T$ ▷ trace-reduced Gram (D1), Eq. (6)
9: $\tilde{K} \leftarrow K + \hat{\sigma}^2 d \cdot I_T$ ▷ aleatoric ridge, Eq. (7)
10: $L \leftarrow \mathrm{Cholesky}(\tilde{K})$ ▷ $\tilde{K} = LL^{\top}$
11: **for** $t = 1, \ldots, T$ **do**
12: $r_t \leftarrow 2 \log L_{tt}$ ▷ Schur complement = Eq. (8)
13: **end for**
14: **return** $r_{1:T}$

A Implementation

A.1 Pseudocode

Algorithm 1 consolidates the reward computation of §3 into explicit pseudocode and computes the per-step rewards $r_{1:T}$ for a single imagined rollout. The three substantive steps mirror §3: lines 1–7 evaluate the ensemble and form the centred deviations $\boldsymbol{\delta}_k^{(t)}$ of Eq. (5) (assumptions A1–A2, §3.1); line 8 contracts each $d \times d$ block of the epistemic covariance to its trace to obtain the kernel K of Eq. (6) (design choice D1, §3.2); lines 9–13 recover the per-step rewards from the Cholesky factorisation of the ridged kernel $\tilde{K} = K + \hat{\sigma}^2 d \cdot I_T$. The Schur-complement identity discussed in §3.3 guarantees that $r_t = 2 \log L_{tt}$ realises Eq. (8) exactly.

Algorithm 2 maintains the aleatoric estimate $\hat{\sigma}^2$ used as the ridge in Algorithm 1. As described in §3.4, we estimate it post-hoc from the MSE residuals of the ensemble mean on training transitions, and update it with an exponential moving average with each training step; the running buffer is initialised to zero, in which case the first batch value is taken directly to avoid biasing the estimator toward the initialisation.

A.2 Architecture and Training

CIG and all baselines share a common DreamerV2 backbone [9] and differ only in the form of the intrinsic reward. CIG introduces no architectural changes to the backbone.

Architecture. The world model follows the Recurrent State-Space Model (RSSM) [8, 9] with a discrete 32×32 stochastic state space. A convolutional encoder maps pixel observations to the latent state; a convolutional decoder reconstructs observations from the concatenation of the deterministic and stochastic components. Since we operate in a reward-free setting, no reward predictor is attached. The actor and critic are MLPs conditioned on the concatenated latent: the actor outputs a truncated Normal distribution for continuous action spaces and a categorical distribution for discrete action spaces; the critic outputs a scalar state value. The ensemble consists of $M = 5$ independently initialised MLPs operating in the latent space of the world model. All members receive the same training data and differ only in their random initialisation; we use no bootstrap masks, diversity penalties, or architectural variation between members. The ensemble is used by CIG, Plan2Explore and E3B $\times$ P2E; remaining baselines do not require it, but instead have different architectures that include MLPs of the same depth and width as the ensemble MLPs.

Algorithm 2 Aleatoric scale update (Appendix A.5).

Require: Mini-batch MSE residuals $\{\epsilon_i\}_{i=1}^{B}$ between ensemble means and targets; decay $\beta \in [0, 1)$; running estimate $\hat{\sigma}^2$ (initialised to 0).

$\bar{\epsilon} \leftarrow \frac{1}{B} \sum_{i=1}^{B} \epsilon_i$
if $\hat{\sigma}^2 = 0$ **then** ▷ first update: take the batch value directly
    $\hat{\sigma}^2 \leftarrow \bar{\epsilon}$
else
    $\hat{\sigma}^2 \leftarrow \beta\, \hat{\sigma}^2 + (1 - \beta)\, \bar{\epsilon}$
end if

Training. All components are trained with Adam [30]. The world model is trained on replay-buffer sequences by jointly optimising the reconstruction, KL-regularisation, and discount losses of DreamerV2. The ensemble is trained alongside the world model via per-member mean-squared error on replay-buffer transitions encoded to latent states; the reward path detaches the ensemble outputs, so the intrinsic reward does not backpropagate into the ensemble parameters. The actor and critic are trained on imagined rollouts of the world model: starting from replay-sampled latents, the actor is rolled out through the RSSM transition model for a fixed horizon, and the intrinsic reward is computed at each imagined step. The critic is regressed onto λ-returns; the actor is updated through the world-model dynamics via reparameterisation gradients (continuous action spaces) or REINFORCE (discrete action spaces). The behaviour policy is the actor itself, with no additional noise other than that induced by the entropy bonus that is part of DreamerV2, thus exploration is mainly driven by the intrinsic reward.

Baselines. All baselines use the same DreamerV2 world model, actor–critic architecture, and training procedure described above. Baselines that require ensemble disagreement (Plan2Explore) use the same ensemble. Each baseline reward is computed on imagined rollouts in the same latent space, isolating the reward as the sole variable of comparison. The specific baseline rewards are listed in §F.1.

A.3 Hyperparameters

CIG is built on top of a DreamerV2 world model [9] augmented with a Plan2Explore [5] ensemble. We follow the default DreamerV2 backbone configuration of Hafner et al. [9] unchanged and refer the reader there for full detail; the most important values are restated in Table 2 alongside the hyperparameters introduced by CIG and the Plan2Explore ensemble.

A.4 Environments

All environments operate in a fully reward-free setting: the agent never observes extrinsic reward during pretraining. Instead, we track a domain-specific coverage proxy for each environment that measures how much of the reachable state space the agent has visited. Every environment provides 64×64 RGB (see Figure 7). Table 3 gives an overview; the paragraphs that follow supply the details needed for reproduction.

A.4.1 MiniGrid

All MiniGrid environments use the standard seven discrete actions (turn left, turn right, move forward, pick up, drop, toggle, done) and render a 64×64 RGB egocentric partial view. Environments are procedurally generated, so the agent encounters a new layout variant on every reset. Episodes are truncated after the per-task default step budget defined by Chevalier-Boisvert et al. [26].

The exploration metric shared by all MiniGrid tasks is *cumulative successes*: at every environment step we check whether the agent has reached a rewarding state (the goal tile, target object, etc.) and, if so, record a single success for that episode. Visiting the rewarding state more than once within the same episode does not increment the counter further. We adopt this metric because the agent never observes the extrinsic signal, so cumulative successes measure how often reward-free exploration incidentally solves the task, which is the quantity that ultimately matters for any downstream exploitation phase.

Table 2: Hyperparameters for the CIG experiments. Backbone settings not listed here follow the DreamerV2 defaults of Hafner et al. [9].

Hyperparameter	**Symbol**	**Value**
CIG reward (Eq. 7, Eq. 8)		
Ridge scale multiplier	—	1.0
$\hat{\sigma}^2$ EMA momentum	β_{σ^2}	0.99
Plan2Explore ensemble (carries A1)		
Ensemble size	M	5
Target dimensionality	d	200 (latent, MiniGrid) / 1536 (encoding, OGBench)
Head architecture	—	shared MLP (below)
Shared MLP architecture and optimization		
Hidden layers	—	4
Hidden units per layer	—	400
Activation	—	ELU
Initialization	—	Glorot
Optimizer	—	Adam, $\epsilon = 10^{-5}$, wd 10^{-6}, grad-clip 100
DreamerV2 world model		
Stochastic latent (categorical)	—	32×32
Deterministic GRU state	—	600
KL balancing	α	0.8
Free nats	—	0.0
CNN channel multiplier	—	48
Model learning rate	—	2×10^{-4}
Actor–critic over imagined rollouts		
Imagination horizon	T	15
Discount	γ	0.99
λ-return	λ	0.95
Actor learning rate	—	8×10^{-5}
Critic learning rate	—	8×10^{-5}
Slow-critic update	—	hard, every 100 gradient steps
Replay buffer and training schedule		
Replay capacity	—	10^6 transitions
Sequence length	—	50
Batch size (trajectories)	—	50
Random prefill	—	5,000 env steps
Train every	—	100 env steps
Gradient updates per train call	—	10
Reward normalization		
Per-step reward normalization	—	z-score
Running statistics EMA momentum	—	0.99
Return-level normalization	—	Welford running return std

KeyCorridorS4R3. KeyCorridorS4R3 requires picking up a target object behind a locked door. The key is placed behind a separate, closed but unlocked door, so the agent must enter a side room, collect the key, return to the corridor, and unlock the target door. The layout consists of three rows of 4×4 rooms connected by a central corridor. The task tests multi-step credit assignment under sparse structure: the agent must commit to a long prerequisite detour before any successful trajectory becomes possible, so it reveals whether exploration can discover and follow a chained prerequisite structure rather than greedily pursuing local novelty.

MultiRoom-N7-S8. MultiRoom-N7-S8 arranges seven rooms of randomised size (up to 8×8) in sequence, each connected by a closed door the agent must toggle open. The agent starts in the first room and must reach a green goal tile in the last. The task tests long-horizon directed exploration:

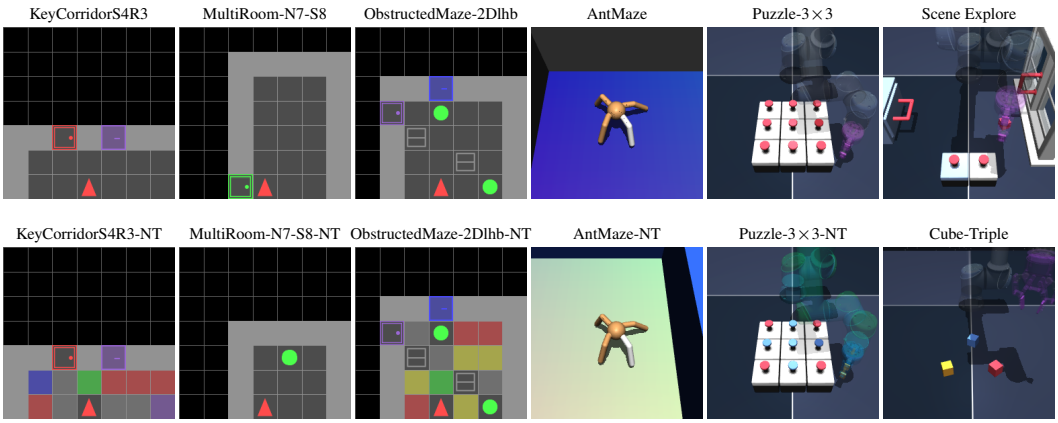


Figure 7: RGB observations from the evaluation environments (the agent sees 64×64 downsampled versions).

the agent must keep pushing into unseen rooms rather than collapsing onto the well-explored neighbourhood of its starting position.

ObstructedMaze-2Dlhb. ObstructedMaze-2Dlhb is the second tier of the Obstructed Maze suite (two rooms across) and the hardest of our three MiniGrid base tasks. The suffix indicates that doors are simultaneously locked (require a coloured key), hidden (the key is concealed inside a box), and blocked (a ball obstructs the door and must be moved aside). Reaching the target therefore requires chaining several object interactions over a long horizon. The task tests compositional object interaction: each subgoal decomposes into distinct manipulation primitives, so the environment reveals whether exploration scales to settings that require correctly composing multiple interactions.

A.4.2 OGBench

All OGBench environments provide 64×64 RGB observations and continuous actions. Episodes run for a fixed 1000 steps and are truncated at the end of the budget; there is no early termination. Initial configurations are pinned with a fixed reset seed across episodes. Each environment defines a coverage metric following the same template: a continuous state quantity is discretised into cells or configuration identifiers, each identifier is added to a visited set upon first encounter, and the reported metric is the fraction of reachable identifiers visited at least once (expressed as a percentage). The visited set is monotone non-decreasing within each evaluation window. Per-environment details of the discretisation and state quantity are given below.

Scene Explore. The Scene environment is a tabletop manipulation task in which a 7-DoF arm interacts with two buttons, a drawer, a window, and a single block. Both buttons start locked, the drawer and window start fully closed, and the block is placed at a fixed location. The task tests breadth of interaction across heterogeneous affordances: buttons toggle, the drawer and window slide, and the block can be lifted and translated, so it reveals whether the policy discovers all available affordances rather than fixating on the easiest one. We enumerate a fixed set of sub-tasks defined by the scene's manipulable elements (each button unlocked, drawer opened, window opened, block displaced) and mark a sub-task as solved once its ground-truth state predicate fires for the first time. The metric is the fraction of sub-tasks solved.

AntMaze. The medium variant of the OGBench antmaze: an 8-DoF quadruped navigates a fixed maze layout from a top-down view. The agent is always reset at the same cell. The task tests long-horizon spatial exploration coupled with high-dimensional continuous control from pixels: the agent must acquire competent locomotion of an 8-DoF body and use that competence to reach spatially novel regions of the maze. The maze is partitioned into a grid of cells defined by the underlying layout. At every step the ant's torso (x, y) position is snapped to the enclosing cell and added to the visited set. The metric is the fraction of reachable cells visited.

Table 3: Summary of evaluation environments. "Distractor" entries refer to the visual-distractor variants described in §A.4.3.

Environment	Action space	Ep. length	Capability tested	Metric
MiniGrid (discrete, 7 actions)				
KeyCorridorS4R3	discrete	default	multi-step credit assignment	cum. successes
MultiRoom-N7-S8	discrete	default	long-horizon directed expl.	cum. successes
ObstructedMaze-2Dlhb	discrete	default	compositional object interact.	cum. successes
+ distractor variants	(same metrics; see §A.4.3)			
OGBench (continuous)				
Scene Explore	7-DoF arm	1000	heterogeneous affordances	sub-task %
AntMaze	8-DoF ant	1000	spatial expl. + locomotion	cell %
Puzzle-3×3	7-DoF arm	1000	combinatorial coverage	config. %
Cube-Triple	7-DoF arm	1000	contact-rich rearrangement	config. %
+ distractor variants	(same metrics; see §A.4.3)			

Puzzle-3×3. A 3×3 lights-out style button puzzle. A 7-DoF arm presses buttons arranged in a grid; pressing one toggles it and its orthogonal neighbours, yielding $2^9 = 512$ reachable configurations. All nine buttons start in the off state. The task tests combinatorial state-space coverage: the 512 configurations are accessible only via long action chains under adjacent-toggle dynamics, so it reveals whether the agent systematically traverses a discrete combinatorial space rather than visiting only configurations close to the initial state. The ground-truth on/off state of all nine buttons is read at every step and treated as a 9-bit configuration identifier. The metric is the fraction of the 512 reachable configurations visited.

Cube-Triple. The three-cube tabletop task: a 7-DoF arm interacts with three cubes initialised in a fixed row. The task tests configurational exploration in a continuous, contact-rich domain: the agent must grasp, lift, and rearrange multiple objects, so it reveals whether the method discovers diverse multi-object configurations rather than collapsing onto single-object behaviour. The workspace is partitioned into a 3D grid (using the cube spawn ranges in x, y, z at the spacing used to generate goal positions). Each cube's position is snapped to its enclosing cell, and the three cell indices are concatenated into a configuration identifier. The metric is the fraction of unique discretised configurations visited.

A.4.3 Visual-Distractor Variants

For each domain we construct distractor variants that instantiate the *noisy-TV* problem with action-dependent noise: a region of the observation is overwritten with content that is sampled from a stochastic process the agent cannot predict, but whose updates the agent itself triggers through a designated action. The agent therefore has full control over *when* the distractor refreshes, and no control over *what* appears, which is exactly the setting that traps prediction error driven exploration, since each refresh produces genuinely novel pixels that no learned model can anticipate. These variants test whether the method can separate genuine state-space novelty from this self-induced but irreducible pixel entropy. The underlying task, action space, episode budget, and exploration metric of each base environment are left unchanged; only the visual observation is modified. Table 4 summarises the variants; details follow.

MiniGrid distractor variants. In KeyCorridorS4R3 and ObstructedMaze-2Dlhb, the floor within the agent's field of view is overwritten with random colours; the colours are independently re-sampled whenever the agent issues the `drop` action. In MultiRoom-N7-S8 the floor cannot be modified because the green goal tile must remain visually identifiable; instead, we add an orb to the observation whose colour is re-sampled on the same `drop` action. In all three cases the trigger is task-irrelevant: `drop` has no effect on the underlying MiniGrid task in our configurations, so the agent can summon a fresh, unpredictable visual sample at any time without progressing toward the goal. These variants test whether the method still discovers the prerequisite structures of each base task (key fetching, room traversal, compositional object interaction) when an action-gated noisy TV is freely available as an alternative.

Table 4: Noisy-TV distractor variants with action-dependent noise. Each row modifies only the observation of its base environment; all other aspects (task, actions, episode length, metric) are identical. The agent triggers each refresh through the listed action, but the sampled content is unpredictable, so an agent that maximises naive prediction error can in principle drive unbounded reward through its own behaviour without ever exploring the task.

Base environment	**Distractor target**	**Trigger**
KeyCorridorS4R3	Floor texture	`drop` action
MultiRoom-N7-S8	Colour-changing orb	`drop` action
ObstructedMaze-2Dlhb	Floor texture	`drop` action
AntMaze	Wall colours	Action-dim threshold
Puzzle-3×3	Robot appearance	Centre-button press

AntMaze with action-gated distractor. The maze wall colours are re-sampled whenever a designated action dimension exceeds a threshold; the floor texture, ant body colours, and lighting are left untouched. The distractor is therefore controllable: the agent can summon arbitrarily many fresh wall colourings without changing its (x, y) position in the maze, but the sampled colours themselves are unpredictable. This variant tests whether the method separates spatial novelty (reaching a new maze region) from self-induced surface-level pixel change.

Puzzle-3×3 with action-gated robot distractor. The robot's rendered appearance (the UR5e arm and Robotiq gripper materials) is re-sampled whenever the agent presses the centre button; the puzzle board and surrounding scene are left clean. The trigger is itself a puzzle action, but the colour change carries no task-relevant information: the on/off button configuration that defines the metric is unaffected, so an agent that maximises naive prediction error can exploit the distractor by repeatedly toggling the centre button without exploring the rest of the combinatorial configuration space. The variant tests whether the method resists such self-induced novelty and continues to explore the full button-configuration space.

A.5 Aleatoric Variance Estimation

Ensemble training dynamics Assumption A2 posits $s_{t+1} \mid k, s_t, a_t \sim \mathcal{N}(\mu_k(s_t, a_t), \sigma^2 I_d)$ with a shared, isotropic, non-learned covariance. We adopt this generative model directly from Plan2Explore [5] for two reasons. First, it is the probabilistic model implicit in the P2E training protocol, and holding it fixed preserves the P2E ensemble training dynamics exactly; Conditional Information Gain (CIG) then differs from P2E only at reward-computation time and in the resulting distribution of training data, so any downstream comparison attributes performance differences to the reward itself. Second, Plan2Explore establishes empirically that this noise model is sufficient for effective ensemble-based exploration in the Dreamer latent space, so we inherit its validation rather than re-litigate it.

The scalar σ^2 is not optimized during ensemble training and therefore must be supplied separately at reward-computation time, where it enters the ridge $\sigma^2 d \cdot I_T$ in Eq. (7). We estimate it post-hoc from ensemble-mean residuals.

Estimating σ^2 The scalar σ^2 does not appear in the ensemble training loss and must be estimated post-hoc for use in the ridge term of Eq. (7). We use residuals against the *ensemble mean* $\bar{\mu}(s_n, a_n) = \frac{1}{M}\sum_k \mu_k(s_n, a_n)$:

$$\hat{\sigma}^2 = \frac{1}{Nd}\sum_{n=1}^{N} \left\| s_{n+1} - \bar{\mu}(s_n, a_n) \right\|^2. \tag{9}$$

Residuals against individual members would conflate aleatoric noise with epistemic disagreement; the ensemble mean cancels the symmetric component of that disagreement and leaves a residual whose squared norm is dominated by the aleatoric contribution. For finite M, $\bar{\mu}$ retains some epistemic error, so $\hat{\sigma}^2$ upper-bounds the true aleatoric floor. This direction of bias is the conservative one: overestimating the aleatoric floor attenuates the conditional correction rather than amplifying it. In practice, $\hat{\sigma}^2$ is maintained as a running average over ensemble-mean residuals on training transitions.

A.6 Compute & Wall-clock Time

All experiments were conducted on machines equipped with a single NVIDIA RTX 3090 GPU (24 GB VRAM), 32 GB RAM, and 8 CPU cores. Each individual run completes in under 24 hours, with a typical wall-clock time of approximately 20 hours depending on the level of metric logging. The computational overhead of the proposed intrinsic reward mechanisms is negligible relative to the DreamerV2 base agent, which accounts for the majority of resource consumption. The difference between CIG and P2E, the only other ensemble-based baseline in our experiments, is less than 1% in FLOPS, since the ensemble forward passes, which are shared by both, take the majority of compute. However, in practice the difference is around 10% in wall-clock time depending on the GPU architecture and optimizations made by the PyTorch compiler. We expect this gap to narrow with straightforward engineering efforts.

Our main evaluation comprises 12 tasks across 7 method configurations, each repeated over 5 seeds, yielding a total of 420 runs. Reproducing the full set of main results therefore requires approximately 8,400 single-GPU hours.

A.7 Code

The code used to produce the experiments is original code by the authors and based on PyTorch (https://pytorch.org/).

The environments are derived from the following projects:

1. OGBench (https://github.com/seohongpark/ogbench) — MIT License
2. MiniGrid (https://github.com/Farama-Foundation/Minigrid) — Apache License 2.0

B Gaussian Approximations

Tightness of the Gaussian Entropy Bound (A3) Approximation A3 replaces the mixture entropy $H(s_{1:T})$ with the entropy of the moment-matched Gaussian $q = \mathcal{N}(\bar{\boldsymbol{\mu}},\, \Sigma)$, where $\Sigma = \sigma^2 I_{Td} + C$. Because q shares the mean and covariance of the mixture $p = \frac{1}{M}\sum_k \mathcal{N}(\boldsymbol{\mu}_k,\, \sigma^2 I_{Td})$, the cross-entropy $-\int p \log q$ evaluates to $H(q)$, and the gap introduced by A3 is

$$H(q) - H(p) \;=\; D_{\mathrm{KL}}(p \,\|\, q) \;\geq\; 0. \tag{10}$$

The surrogate therefore overestimates entropy for every trajectory. We now argue that this overestimation is benign in both regimes encountered during training.

Well-explored regions. As the agent collects data in a region of the state space, the ensemble members converge toward agreement: the means $\boldsymbol{\mu}_k$ approach a common function and the epistemic covariance $C \to 0$. In this limit the mixture p collapses to a single Gaussian identical to q, so $D_{\mathrm{KL}}(p \,\|\, q) \to 0$ and the bound becomes tight. The surrogate is therefore most faithful in precisely the regions where the agent must discriminate between trajectories of similar information content.

Unexplored regions. When the ensemble members disagree strongly, the mixture is multi-modal and the Gaussian approximation can be loose. Two observations limit the practical impact of this looseness. First, the true information gain is itself large in unexplored regions, so even a coarse ranking of trajectories directs the agent toward informative transitions. Second, the policy has little basis for fine-grained trajectory selection in parts of the state space it has rarely visited; the surrogate need only distinguish broadly informative directions from uninformative ones, a task for which the overestimate suffices. The failure mode of A3 is therefore over-exploration rather than premature convergence: the agent may spend additional time in high-uncertainty regions, but it will not be steered away from them.

Empirical precedent. Plan2Explore [5] relies on the same moment-matching approximation applied independently to each step: its ensemble-disagreement reward is the entropy of the Gaussian matched to the per-step predictive $p(s_t)$, discarding the mixture structure in exactly the same way as A3. The empirical success of this per-step surrogate across a range of continuous-control tasks provides evidence that the Gaussian approximation is adequate for exploration in practice. Our usage

extends the approximation from the d-dimensional per-step marginal to the Td-dimensional joint trajectory distribution, where cross-step correlations can introduce additional non-Gaussianity absent from the marginals. The per-step evidence is therefore encouraging but does not directly certify the trajectory-level bound; the experiments in §4 provide this validation.

C Why the Exact Gaussian Bound Fails

Under A1–A3, the Gaussian bound of Eq. (4) is the natural trajectory objective, and maximising it directly would make the design choice D1 unnecessary. Up to a factor of $\frac{1}{2}$ and a policy-independent constant, the bound equals

$$\log\det\big(I_{Td} + C/\sigma^2\big), \tag{11}$$

which is the *No Trace Reduction* variant from our ablations, but we call the *exact objective* for brevity here. In our ablations, however, this variant underperforms CIG (§4.5). This failure is an empirical finding, and we do not attempt to prove that it must occur. Instead, this appendix offers mathematical intuition for why the exact objective fails and under which conditions the failure is most pronounced.

The underlying reason is that the exact objective asks for far more than the ensemble can provide. It depends on the full covariance of the predicted rollout, that is, on how every latent coordinate at every step varies together with every other, yet it observes the posterior over the dynamics only through the M ensemble members. Written in terms of the members' deviations stacked over the rollout, Eq. (5) is the sample covariance of these M members:

$$C = \frac{1}{M}\sum_{k=1}^{M} \mathbf{d}_k \mathbf{d}_k^\top, \qquad \mathbf{d}_k = \big[\boldsymbol{\delta}_k^{(1)}; \ldots; \boldsymbol{\delta}_k^{(T)}\big] \in \mathbb{R}^{Td}. \tag{12}$$

Since the deviations are measured from the ensemble mean, they sum to zero and amount to only $M-1$ independent samples. From these, C estimates a covariance with Td rows and columns; in our experiments, $M = 5$ samples face $Td \geq 3000$ dimensions ($d \geq 200$, $T = 15$). Nothing in this argument is specific to ensembles: it applies equally to M samples from any other approximate posterior over the dynamics. Estimating a covariance from far fewer samples than dimensions is a familiar difficulty in statistics. What goes wrong is how the exact objective combines these entries into a reward for the policy, and this is what the remainder of the appendix examines. We first introduce a view of the objective in which its per-step reward becomes transparent (§C.1), and then use this view to derive two consequences of the small sample size: rank saturation (§C.2) and a signal-to-noise problem (§C.3).

C.1 Approach: Adding the Rollout's Coordinates One at a Time

The policy is trained on per-step rewards, so we examine the reward that the exact objective assigns to a single step. Following the chain rule of Eq. (1), this reward is the increase of the log-determinant when the step is appended to the rollout. A determinant measures the volume of the covariance ellipsoid, so the reward of a step is the increase in log-volume that the step causes. Appending a step extends the deviation vector of every member by the d coordinates of the new step. The question is therefore how the volume of the covariance spanned by these vectors grows as coordinates are added to them. We answer it one coordinate at a time, which is more general than adding a whole step and makes the contribution of each coordinate visible; the reward of a step is then the growth over its d coordinates. Formally, we order the Td coordinates step by step and write $C_{1:n}$ for the covariance of the first n of them, i.e. the leading $n \times n$ block of C. We call $\log\det(I + C_{1:n}/\sigma^2)$ the *log-volume* of these coordinates, and the reward of step t is

$$\Delta_t = \log\det\big(I + C_{1:td}/\sigma^2\big) - \log\det\big(I + C_{1:(t-1)d}/\sigma^2\big), \tag{13}$$

and these rewards sum to Eq. (11).

From coordinates to members. Working with $C_{1:n}$ directly is cumbersome. Its size grows with every coordinate and, over a full rollout, depends on both d and T, which obscures what an individual coordinate contributes. We therefore switch to the matrix of inner products between the members. By Sylvester's determinant lemma, this matrix has the same log-determinant as $C_{1:n}$, but its size is fixed at $M \times M$, and it is built by adding one rank-one term per coordinate. Both matrices are

formed from the same numbers, the deviations of the M members in each coordinate. We collect the deviations in coordinate i, scaled by $1/\sqrt{M}$, in a vector $\mathbf{x}_i \in \mathbb{R}^M$. The covariance then contains the inner products between coordinates, whereas the new matrix $G_{1:n}$ contains the inner products between members:

$$\big(C_{1:n}\big)_{ij} = \mathbf{x}_i^\top \mathbf{x}_j, \qquad G_{1:n} = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top \in \mathbb{R}^{M\times M}. \tag{14}$$

Sylvester's determinant lemma, $\det(I + A^\top A) = \det(I + AA^\top)$, applied to the matrix with columns $\mathbf{x}_i/\sigma$, equates the two log-determinants:

$$\log\det\big(I_n + C_{1:n}/\sigma^2\big) = \log\det\big(I_M + G_{1:n}/\sigma^2\big). \tag{15}$$

Entry (k, l) of $G_{1:n}$ is the inner product of the deviations of members k and l over the first n coordinates, and appending coordinate n simply adds the term $\mathbf{x}_n\mathbf{x}_n^\top$.

Eigenvalues as patterns of disagreement. The determinant of a covariance matrix is the product of its eigenvalues, so the log-volume is a sum with one logarithmic term per eigenvalue of $G_{1:n}$. How the log-volume, and with it the reward, grows as coordinates are added is therefore determined by how these eigenvalues develop as the dimension of the deviation vectors increases. This development is the main concern of the remainder of this appendix. Not all M eigenvalues take part, however. Since the deviations sum to zero across the members, every $\mathbf{x}_i$ is orthogonal to the all-ones vector $\mathbf{1}$: shifting all members by the same amount is not disagreement. Hence $\mathbf{1}$ is an eigenvector of $G_{1:n}$ with eigenvalue zero, which contributes $\log 1 = 0$, and only the remaining $M - 1$ eigenvalues $g_1 \geq \cdots \geq g_{M-1} \geq 0$ enter the log-volume:

$$\log\det\big(I + G_{1:n}/\sigma^2\big) = \sum_{m=1}^{M-1} \log\big(1 + g_m/\sigma^2\big), \qquad g_m = \sum_{i=1}^{n} \big(\mathbf{u}_m^\top \mathbf{x}_i\big)^2. \tag{16}$$

The corresponding orthonormal eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_{M-1}$ are orthogonal to $\mathbf{1}$, so their entries sum to zero, and each describes a pattern of disagreement among the members. The pattern $(1, -1, 0, \ldots, 0)/\sqrt{2}$, for instance, contrasts members 1 and 2. Its eigenvalue measures how strongly the members disagree in this pattern, accumulated over the first n coordinates, to which every coordinate contributes its squared projection onto the pattern. Eq. (16) turns the question of how a step enlarges the volume into a question about the $M - 1$ eigenvalues of G: how many of them the coordinates can raise (§C.2), and how large they already are when a new coordinate arrives (§C.3).

C.2 Rank Saturation

We first ask how many independent directions of disagreement the exact objective can represent, since every such direction contributes a term of its own to the log-volume. The answer is at most $M - 1$, and this limit is reached almost immediately. Intuitively, all disagreement captured by C lies in the subspace spanned by the M deviation vectors $\mathbf{d}_k$ of Eq. (12). Because these vectors sum to zero, this subspace has only $M - 1$ dimensions, however large Td is. The matrix G makes this explicit: it has only M dimensions to begin with, and, as shown above, one of them, the all-ones direction, carries no disagreement. Each of the remaining directions corresponds to one of the patterns of Eq. (16).

Proposition 1 (Rank saturation)**.** *For any latent dimension d, horizon T and predictions of the M members, and for every $n \leq Td$, $\operatorname{rank}(C_{1:n}) = \operatorname{rank}(G_{1:n}) \leq M - 1$. In particular, C and $G = G_{1:Td}$ have at most $M - 1$ nonzero eigenvalues.*

Proof. Let $X \in \mathbb{R}^{M\times n}$ have the columns $\mathbf{x}_1, \ldots, \mathbf{x}_n$. By Eq. (14), $C_{1:n} = X^\top X$ and $G_{1:n} = XX^\top$, so both have the rank of X. Since every $\mathbf{x}_i$ is orthogonal to $\mathbf{1}$, we have $\mathbf{1}^\top X = \mathbf{0}^\top$: the M rows of X are linearly dependent, so $\operatorname{rank}(X) \leq M - 1$. □

The limit is reached quickly because G is built from one rank-one term per coordinate. A direction that no preceding coordinate has reached yet has eigenvalue zero, because all their projections onto it vanish in Eq. (16). Unlike the all-ones direction, whose eigenvalue is zero by construction for every n and which is therefore excluded from Eq. (16), such a direction is one of the $M - 1$ terms of the sum and contributes $\log 1 = 0$ only until the first coordinate disagrees along it. Each new term $\mathbf{x}_n\mathbf{x}_n^\top$

opens such a direction unless $\mathbf{x}_n$ lies in the span of the vectors of the preceding coordinates, i.e. in the column space of $G_{1:n-1}$. For independently drawn members, such as independently initialised ensemble members, $\mathbf{x}_n$ almost surely lies outside this span while fewer than $M-1$ directions are open: the span has dimension below $M-1$ and thus zero volume within the $(M-1)$-dimensional space of vectors whose entries sum to zero, and independent draws land in such a set with probability zero. The exception is a coordinate that is an exact linear function of the preceding coordinates under the posterior, as when the same unknown drives several coordinates (a case we return to in §C.3); its vector is then confined to the span by construction. After $M-1$ coordinates, all directions of disagreement are therefore open almost surely, and since a single step contributes $d \gg M$ coordinates, this typically happens within the first step of the rollout. From then on, every further coordinate is, in the sample, an exact linear combination of its predecessors.

Running out of directions matters because opening a direction and lengthening an existing one are rewarded very differently. Suppose that the vector of coordinate n points along a single direction with eigenvalue g_m, which is zero if the direction is new, and raises this eigenvalue by the coordinate's disagreement $a = \|\mathbf{x}_n\|^2$. By Eq. (16), the log-volume grows by

$$\log\Big(1 + \frac{g_m + a}{\sigma^2}\Big) - \log\Big(1 + \frac{g_m}{\sigma^2}\Big) = \log\Big(1 + \frac{a}{\sigma^2 + g_m}\Big). \tag{17}$$

A new direction ($g_m = 0$) adds a full term $\log(1+a/\sigma^2)$ of its own. Lengthening an existing direction only enlarges a term that is already present: the logarithm compresses the gain, and the disagreement already accumulated in the direction acts as additional noise. Across several coordinates, this is the difference between linear and logarithmic growth. For $n \le M-1$ coordinates of equal disagreement a,

$$\underbrace{n \log\big(1 + a/\sigma^2\big)}_{n \text{ new directions}} \qquad \text{versus} \qquad \underbrace{\log\big(1 + na/\sigma^2\big)}_{\text{one direction lengthened } n \text{ times}} \quad . \tag{18}$$

Rank saturation marks the point after which no direction is new, so every further coordinate can only lengthen existing ones. The information gain that the exact objective approximates has no such limit: with the posterior covariance, every coordinate whose unknown is independent of the preceding ones would open a direction of its own, up to Td of them. In the sample, all coordinates of all steps compete for the same $M-1$ directions.

How strongly the compression bites, however, depends on the size of the existing eigenvalues. While g_m is small compared with σ^2, the denominator in Eq. (17) stays close to σ^2, and lengthening a direction earns almost as much as opening a new one. Whether the lack of fresh directions harms the reward is therefore decided by the size of the eigenvalues, to which we turn next.

C.3 Signal to Noise

The rank bound limits how many eigenvalues are nonzero, not how large they are. Their size matters because it determines how strongly the logarithm compresses further growth. By Eq. (17), raising an eigenvalue by a earns $\log(1 + a/(\sigma^2 + g_m))$. The logarithm compresses every such increase, but while g_m is small compared with σ^2 the compression is negligible and the gain stays close to that of a new direction, whereas once g_m is large the accumulated eigenvalue dominates the noise and the gain shrinks accordingly. To see how this shapes the reward, we ask how much a single coordinate adds to the log-volume, given the coordinates that precede it.

The gain of a coordinate. Eq. (17) assumed that a coordinate points along a single direction. In general, a coordinate spreads its disagreement over several directions, and its gain combines the compression in each of them. Appending coordinate n adds the rank-one term $\mathbf{x}_n\mathbf{x}_n^\top$ to $G_{1:n-1}$, and by the matrix determinant lemma, $\det(A + \mathbf{x}\mathbf{x}^\top) = \det(A)(1 + \mathbf{x}^\top A^{-1}\mathbf{x})$, the log-volume grows by

$$\log\Big(1 + \mathbf{x}_n^\top \big(\sigma^2 I_M + G_{1:n-1}\big)^{-1} \mathbf{x}_n\Big) = \log\Big(1 + \sum_{m=1}^{M-1} \frac{(\mathbf{u}_m^\top \mathbf{x}_n)^2}{\sigma^2 + g_m}\Big), \tag{19}$$

where g_m and $\mathbf{u}_m$ refer to $G_{1:n-1}$. The disagreement $\|\mathbf{x}_n\|^2$ of the coordinate is split among the $M-1$ directions according to its squared projections onto them, and each share is measured against σ^2 plus the disagreement that the preceding coordinates have already accumulated in that direction. If $\mathbf{x}_n$ points along a single eigenvector, the sum has one term and Eq. (17) is recovered. Accumulated

disagreement thus acts as additional noise for every new coordinate, and how much it matters depends on the size of the eigenvalues. We first treat the case in which all eigenvalues are small, and then the case in which they are large, where the outcome depends on how the posterior uncertainty is distributed.

Small eigenvalues: the exact objective reduces to Plan2Explore. Suppose that the coordinates add so little disagreement that all eigenvalues stay small compared with σ^2. The logarithm then barely compresses: by Eq. (17), lengthening an existing direction earns nearly as much as opening a new one, and each term of Eq. (16) is close to g_m/σ^2. The log-volume is then approximately the sum of the eigenvalues divided by σ^2. By the definition of g_m in Eq. (16), this sum is the total disagreement of all coordinates, $\sum_m g_m = \sum_i \|\mathbf{x}_i\|^2$, since the eigenvectors complete $\mathbf{1}$ to an orthonormal basis and every $\mathbf{x}_i$ is orthogonal to $\mathbf{1}$. The reward of step t thus reduces to the disagreement of its own coordinates,

$$\Delta_t \approx \frac{\operatorname{tr} C_{tt}}{\sigma^2}, \tag{20}$$

which is the ensemble disagreement of step t and hence the Plan2Explore reward up to a constant factor. The reason is that the exact objective detects redundancy only through the compression of the logarithm: a coordinate that repeats disagreement accumulated by earlier coordinates lengthens an existing direction and should earn less than one that opens a new direction. Without compression, both earn the same, and the reward no longer depends on the prefix. In this regime, the exact objective therefore retains the lifelong signal but loses the prefix-redundancy correction. However, there are two reasons why this is not a failure: First, the information gain is nearly additive across steps in this setting. Second, the regime is confined to states in which the members almost agree, which a policy that seeks disagreement rarely targets.

Large eigenvalues. Once the eigenvalues are comparable to or larger than σ^2, the compression is substantial: how much a new coordinate earns now depends on the disagreement that the preceding coordinates have accumulated in each direction (Eq. (19)). This is what allows the exact objective, in principle, to discount redundant coordinates and thus to go beyond Plan2Explore. Whether it does so depends on how a coordinate splits its disagreement among the $M-1$ directions, which is determined by the structure of the posterior uncertainty. We consider two settings at opposite ends.

Large eigenvalues, isotropic uncertainty: the reward ignores what a step probes. With isotropic uncertainty, a step that repeats an earlier one and a step that probes new unknowns receive the same reward. Suppose that within each step, the posterior uncertainty is spread over many latent directions that fluctuate independently with comparable variances, so that no single unknown dominates. Each individual coordinate adds a rank-one term and therefore raises the eigenvalues only along one specific direction, which differs randomly from coordinate to coordinate. Over the $d \gg M$ coordinates of a step, these differences average out: because the members are exchangeable, every direction orthogonal to $\mathbf{1}$ receives the same share in expectation, and a sum of many independent terms concentrates around its expectation. Each step therefore raises all $M-1$ eigenvalues by approximately the same amount, $1/(M-1)$ of its disagreement:

$$G_{1:td} - G_{1:(t-1)d} \approx \frac{\operatorname{tr} C_{tt}}{M-1}\Big(I_M - \tfrac{1}{M}\mathbf{1}\mathbf{1}^\top\Big), \tag{21}$$

where the matrix in brackets acts as the identity on the subspace orthogonal to $\mathbf{1}$. This matrix depends on the step only through its disagreement $\operatorname{tr} C_{tt}$, not on which unknowns the step probes. A step that repeats an earlier one and a step that probes new unknowns with the same disagreement therefore add the same matrix to G and, by Eq. (16), receive the same reward, although the information gain would credit the new step more than the repeat. The compression is active, but it discounts both kinds of step equally, and the redundancy signal that the chain rule should supply is lost.

Large eigenvalues, concentrated uncertainty: redundancy is detected for at most $M-1$ steps. At the other extreme, the uncertainty of each step is concentrated on a single unknown. We show that the exact objective then distinguishes new from redundant steps, but only until $M-1$ new steps have been taken, after which it again rewards both alike. This is the rank-saturation argument of §C.2, carried over from coordinates to steps. Suppose that each step probes a single unknown that moves the predictions of all members along a fixed latent direction. All coordinates of the step are then driven by the same unknown, so their vectors $\mathbf{x}_i$ are parallel (the exception noted in §C.2), and the

step as a whole adds a single rank-one term to G: its entire disagreement $\operatorname{tr} C_{tt}$ falls on one direction in member space, the direction along which the members disagree about this unknown. Eq. (19) therefore applies to the whole step, with $\mathbf{x}_n$ replaced by a vector $\mathbf{x}$ along this direction whose squared length is $\operatorname{tr} C_{tt}$. To keep the argument transparent, we assume that independent unknowns disagree along orthogonal directions.

As long as some directions remain unopened, new and redundant steps are rewarded differently. A redundant step probes an unknown that an earlier step has already probed, possibly through a different latent direction, and therefore lengthens the direction of that step; a new step probes an independent unknown and opens a direction of its own. By Eq. (17) with $a = \operatorname{tr} C_{tt}$, the new step earns a full term, whereas the redundant step is compressed by the disagreement already accumulated in its direction. In this phase, the exact objective counts distinct unknowns, which is precisely the redundancy signal the chain rule is meant to provide.

As in §C.2, however, this phase ends once all $M-1$ directions are open. A further new unknown then has no direction of its own left and, like a redundant step, can only lengthen existing directions. If the $M-1$ new steps had equal disagreement, all eigenvalues share a common value g, and Eq. (19) gives

$$\Delta_t = \log\Big(1 + \sum_{m=1}^{M-1} \frac{(\mathbf{u}_m^\top \mathbf{x})^2}{\sigma^2 + g}\Big) = \log\Big(1 + \frac{\operatorname{tr} C_{tt}}{\sigma^2 + g}\Big) \tag{22}$$

for new and redundant steps alike, since $\sum_m (\mathbf{u}_m^\top \mathbf{x})^2 = \|\mathbf{x}\|^2 = \operatorname{tr} C_{tt}$ however the disagreement is split among the directions. With equal eigenvalues, G is again a multiple of the identity on the subspace orthogonal to $\mathbf{1}$, as in the isotropic case, and the reward no longer reveals whether a step is new. If the eigenvalues differ, the reward depends on which directions a step lengthens, but it no longer systematically favours new unknowns. With $M = 5$, a rollout of $T = 15$ steps that probes a new unknown at every step thus exhausts the directions after four steps, and its remaining eleven steps are rewarded as if novelty did not matter. In practice, the directions of independent unknowns are not exactly orthogonal but overlap by chance, so the distinction blurs even before all $M-1$ directions are open.

D How the Trace Reduction Fixes the Exact Objective

§C traced the failure of the exact objective to a lack of samples: it estimates a covariance with Td rows and columns from only M members. The trace reduction D1 of Eq. (6) uses the same deviations of the members but counts them differently. Each trace pairs the deviations at two steps coordinate by coordinate and sums over the d latent coordinates. Up to the factor d, K is therefore the sample covariance of Md samples of a T-dimensional vector, one for each pair of member and latent coordinate, which collects the deviations of that member in that coordinate across the T steps. The latent coordinates thus move from the dimensions of the estimated covariance to its samples. Since the deviations in each coordinate sum to zero across the members, these amount to at most $(M-1)d$ independent samples; in our experiments, $(M-1)d \geq 800$ samples face $T = 15$ dimensions. This section shows how this reversal overcomes the two consequences of the small sample size derived in §C. We first recast the CIG surrogate in the view of §C.1, then revisit rank saturation (§D.1) and the signal-to-noise problem (§D.2), and finally state what the trace reduction gives up (§D.3).

Under the trace reduction, the smallest unit of the rollout that enters the objective is a whole step rather than a single coordinate, since every entry of K pairs two whole steps. We therefore follow §C.1 but add the rollout one step at a time. Stacking the vectors $\mathbf{x}_i$ of Eq. (14) over the d coordinates of step t gives one vector per step,

$$\mathbf{y}_t = \big[\mathbf{x}_{(t-1)d+1}; \ldots; \mathbf{x}_{td}\big] \in \mathbb{R}^{Md}. \tag{23}$$

By Eq. (6), the leading $t \times t$ block $K_{1:t}$ of K contains the inner products between steps, whereas the counterpart of $G_{1:n}$ contains the inner products between the Md pairs of member and latent coordinate and gains one rank-one term per step:

$$\big(K_{1:t}\big)_{js} = \mathbf{y}_j^\top \mathbf{y}_s, \qquad H_{1:t} = \sum_{j=1}^{t} \mathbf{y}_j \mathbf{y}_j^\top \in \mathbb{R}^{Md \times Md}. \tag{24}$$

Sylvester's determinant lemma equates the two log-determinants as in Eq. (15), with the ridge $\sigma^2 d$ of Eq. (7) in place of σ^2:

$$\log\det\big(I_t + K_{1:t}/(\sigma^2 d)\big) = \log\det\big(I_{Md} + H_{1:t}/(\sigma^2 d)\big). \tag{25}$$

Since every $\mathbf{x}_i$ is orthogonal to $\mathbf{1}$, every $\mathbf{y}_t$ is orthogonal to the d vectors that shift all members equally in a single latent coordinate, which therefore carry no disagreement. The remaining $(M-1)d$ eigenvectors of $H_{1:t}$ are its directions. A direction of G is a single pattern of disagreement among the members, whereas a direction of H specifies one such pattern for each latent coordinate.

The CIG reward of Eq. (8) is the increase of $\log\det\tilde{K}$ when step t is appended to the rollout (§3.3). Factoring the ridge out of $\tilde{K}$ splits this increase into the floor $\log(\sigma^2 d)$ and the growth of the log-volume in Eq. (25). Appending step t adds the rank-one term $\mathbf{y}_t\mathbf{y}_t^\top$ to $H_{1:t-1}$, so the matrix determinant lemma gives, as in Eq. (19),

$$r_t = \log(\sigma^2 d) + \log\Big(1 + \sum_{m=1}^{(M-1)d} \frac{(\mathbf{u}_m^\top \mathbf{y}_t)^2}{\sigma^2 d + g_m}\Big), \tag{26}$$

where g_m and $\mathbf{u}_m$ now refer to $H_{1:t-1}$. Eq. (26) has the form of Eq. (19), and two differences between them carry the remainder of this section. First, a whole step enters as a single rank-one term, whereas the exact objective adds one term for each of its d coordinates. Second, these terms share $(M-1)d$ directions instead of $M-1$. Together, the two differences decide how many directions a rollout can open, to which we turn first.

D.1 No Rank Saturation

Under the trace reduction, the number of directions that a rollout can open is limited by its length rather than by the size of the ensemble. Rank saturation in §C.2 had two causes: the exact objective has only $M-1$ directions, and the d coordinates of the first step typically open all of them. The trace reduction removes both. It provides $(M-1)d$ directions, and since each step adds a single rank-one term to H, each step opens at most one of them.

Proposition 2 (Rank under the trace reduction)**.** *For any latent dimension d, horizon T and predictions of the M members, and for every $t \le T$,* $\operatorname{rank}(K_{1:t}) = \operatorname{rank}(H_{1:t}) \le \min\big(t, (M-1)d\big)$.

Proof. Let $Y \in \mathbb{R}^{Md\times t}$ have the columns $\mathbf{y}_1, \dots, \mathbf{y}_t$. By Eq. (24), $K_{1:t} = Y^\top Y$ and $H_{1:t} = YY^\top$, so both have the rank of Y, which is at most its number t of columns. Every column is orthogonal to the d vectors that shift all members equally in a single latent coordinate. These vectors are linearly independent, since each is nonzero only in the entries of its own coordinate, so the columns lie in a subspace of dimension $(M-1)d$ and $\operatorname{rank}(Y) \le (M-1)d$. □

Within this bound, every step that the posterior does not confine to its predecessors opens a direction of its own. Step t opens a direction unless $\mathbf{y}_t$ lies in the span of the vectors of the preceding steps, i.e. in the column space of $H_{1:t-1}$. While $t \le (M-1)d$, this span has dimension below $(M-1)d$ and thus zero volume within the $(M-1)d$-dimensional space spanned by the directions. If, as an idealization, the deviations of a step vary freely across its latent coordinates, $\mathbf{y}_t$ therefore almost surely falls outside the span, by the argument of §C.2 with steps in place of coordinates. The exception is a step that the posterior confines to the span of its predecessors, such as an exact repeat of an earlier step. In our setting, a rollout of $T = 15$ steps thus opens one direction per step and uses 15 of at least 800 directions, so it never runs out of them.

Directions that remain available matter because opening a direction and lengthening an existing one are rewarded differently. By Eq. (26), the comparison of Eq. (17) carries over to a whole step that points along a single direction, with $\sigma^2 d$ in place of σ^2: if the direction is fresh, the step earns a full term of its own, whereas if it is already open, the gain is compressed by the disagreement accumulated in it. In the exact objective, this difference disappears once all $M-1$ directions are open, typically within the first step. Under the trace reduction, fresh directions remain available until the end of the rollout, so the difference persists for every step. A fresh direction does not make a step new, however. Its vector $\mathbf{y}_t$ generally also has components along directions that earlier steps have opened, and only the remainder opens a direction of its own. How a step splits between the two, and how strongly the logarithm compresses the former, decides whether the reward separates new from redundant steps, which we examine next.

D.2 Signal to Noise

With a fresh direction available at every step, the reward can separate new from redundant steps, and under idealizations analogous to those of §C.3 it does so in both large-eigenvalue settings in which the exact objective fails. The small-eigenvalue regime, in contrast, is unchanged. The mechanism is visible in Eq. (26). The part of $\mathbf{y}_t$ that lies outside the span of the preceding steps falls on fresh directions, whose eigenvalue is zero, and is measured against $\sigma^2 d$ alone. The part inside the span lengthens directions that earlier steps have opened and is compressed by their eigenvalues. How much of its vector a step places outside this span therefore decides how much of its disagreement is rewarded in full, and we now examine this in the three settings of §C.3.

Small eigenvalues: CIG reduces to Plan2Explore. If all eigenvalues of H stay small compared with $\sigma^2 d$, the logarithm barely compresses, and lengthening a direction earns nearly as much as opening one. As for the exact objective, the reward then loses its dependence on the prefix and reduces to $\log(K_{tt} + \sigma^2 d)$, the reward of Eq. (8) without its prefix-redundancy term, which depends on the step only through its Plan2Explore disagreement K_{tt}. Proposition 4 makes this precise: the prefix-redundancy term vanishes quadratically as the disagreement of the rollout becomes small compared with $\sigma^2 d$. The reasons given in §C.3 for why this is not a failure apply unchanged.

Large eigenvalues, isotropic uncertainty: the reward separates new from repeated steps. Under isotropic uncertainty, the exact objective rewards a step that probes new unknowns and a repeat of an earlier step alike (Eq. (21)). The trace reduction separates the two: it rewards a new step nearly in full and discounts a repeat. This separation weakens with every step of the rollout, but in our setting it remains nearly intact until the end.

The separation rests on how many directions of H are already open when a step arrives. Each preceding step has added a single rank-one term to H, so at most $t-1$ of its $(M-1)d$ directions are open before step t (Proposition 2), and all others still have eigenvalue zero. Under isotropy, a new step is independent of the preceding steps, and its vector points in a random direction that favours none of the directions of H. Its disagreement therefore falls on the open directions in proportion to their number, and in expectation their share is at most

$$\frac{t-1}{(M-1)d}. \tag{27}$$

Wherever the vector points, however, it adds a single rank-one term to H and therefore opens at most one direction, along the part of the vector that lies outside the open ones. This is where the exact objective differs: a step adds d rank-one terms to G, which together raise all of its $M-1$ eigenvalues alike, so every step adds the same matrix to G (Eq. (21)). Under the trace reduction, a new step instead places nearly all of its disagreement on fresh directions, where Eq. (26) measures it against $\sigma^2 d$ alone, and so earns nearly the full reward of a step that is orthogonal to its entire prefix (Proposition 3). A repeat of an earlier step, in contrast, has the same vector as that step, opens no direction, and earns at most $\log 2$ above the floor, however large its disagreement (Proposition 3). The share in Eq. (27) grows with every step, but in our setting it is largest at the last of the 15 steps and, with at least 800 directions, at most $14/800$, below 2%, so the separation holds throughout the rollout.[1]

Large eigenvalues, concentrated uncertainty: redundancy is detected throughout the rollout. With concentrated uncertainty, the exact objective separates new from redundant unknowns only until $M-1$ new unknowns have opened all directions of G (§C.3). The trace reduction keeps separating them throughout the rollout, as long as distinct unknowns act along different latent directions. The reason is that, by Eq. (6), each entry of K pairs the deviations of the members at two steps coordinate by coordinate, so two steps whose deviations are orthogonal have orthogonal vectors, whatever the patterns in which the members disagree. G, in contrast, records only these patterns, so steps along orthogonal latent directions still compete for its $M-1$ directions.

First, a step that probes a new unknown along a latent direction orthogonal to those of all preceding steps has deviations orthogonal to theirs and hence a vector orthogonal to all preceding vectors.

[1] If the uncertainty of a step is spread over fewer latent directions of comparable variance, its disagreement spreads over correspondingly fewer directions of H, and d in Eq. (27) is replaced by their number.

It therefore places its entire disagreement on fresh directions of H and earns the full reward of Proposition 3. Since a latent space of dimension d holds d orthogonal latent directions, this holds for up to d new unknowns rather than $M-1$; in our setting, the reward credits a new unknown in full at all 15 steps instead of only the first four.

Second, a step that probes an earlier unknown again along the same latent direction has deviations parallel to those of the earlier step and, since the members disagree in the same pattern, repeats its vector. It therefore opens no direction and earns at most $\log 2$ above the floor, as in the isotropic case.

Both cases rely on the latent directions, since distinct unknowns must act along different latent directions and a repeated unknown must reuse the same one. What happens when either assumption fails is part of what the trace reduction gives up, to which we turn last.

D.3 What the Trace Reduction Gives Up

The trace reduction gains its directions by treating the latent coordinates as samples, as described at the beginning of this section, and it gives up whatever this treatment ignores. Two things are ignored, and they correspond to the two assumptions on which the concentrated setting of §D.2 relied. First, few latent directions may carry all of the uncertainty, so that the coordinates provide fewer samples than their number suggests. Second, the same unknown may move different latent coordinates at different steps, a link between coordinates that the trace discards.

Few latent directions: distinct unknowns compete again. If distinct unknowns act along the same latent direction, the deviations of their steps are parallel, and their vectors differ only in the patterns in which the members disagree. More generally, suppose that the uncertainty of the whole rollout is confined to r latent directions. Every vector $\mathbf{y}_t$ is then determined by one member pattern for each of these latent directions, and since each pattern sums to zero across the members, all vectors lie in a subspace of dimension $(M-1)r$. In the terms of the beginning of this section, the latent coordinates provide only $(M-1)r$ independent samples instead of $(M-1)d$, and the argument of Proposition 2 gives

$$\operatorname{rank}(H_{1:t}) \le \min\bigl(t, (M-1)r\bigr). \tag{28}$$

Rank saturation therefore returns after $(M-1)r$ steps instead of $(M-1)d$. With a single latent direction, H has no more directions than G: the trace reduction then coincides with the exact objective with the ridge $\sigma^2 d$ in place of σ^2, and distinct unknowns saturate after $M-1$ steps, as in §C.3.

Repeats along a different latent direction: redundancy goes unrecognized. The property that lets a new unknown escape saturation in §D.2 also hides some repeats. If an earlier unknown reappears along a latent direction orthogonal to its earlier one, the step has deviations orthogonal to those of the earlier step, and its vector is orthogonal to the earlier vector, although the members disagree in the same pattern. The step therefore lands on fresh directions of H and earns the full reward of Proposition 3, as if its unknown were new. The exact objective, which records only the member patterns, recognizes the repeat, but only while it has directions left. The cause is the trace itself. By Eq. (6), K_{jt} is the trace of the block C_{jt} and keeps only its diagonal entries, which pair each latent coordinate at step j with the same coordinate at step t. An unknown that moves, say, the first latent coordinate at one step and the second at the other enters C_{jt} only through an off-diagonal entry, which the trace discards.

Weighing the trade. Both costs require specific structure in the uncertainty: distinct unknowns that share a few latent directions, or an unknown that changes its latent direction between steps. The failures of the exact objective, in contrast, arise in both large-eigenvalue settings of §C.3. Moreover, the exact objective recognizes the repeats of the second case only while it has directions left, which in general ends within the first step (§C.2) and under concentrated uncertainty after at most $M-1$ steps. With $M=5$ members and $T=15$ steps, the trace reduction retains more samples than dimensions as long as the uncertainty of a rollout spreads over at least four latent directions, and its advantage grows with their number. Which side of the trade matters more is ultimately an empirical question, and in our ablations the trace reduction comes out ahead (§4.5).

E Properties of the CIG reward

E.1 Limiting Cases of the CIG Reward

The interpretation paragraph of §3.3 claims that the CIG reward r_t interpolates between the lifelong disagreement bonus and the aleatoric floor depending on the relationship between step t and the rollout prefix. The following proposition makes both limits precise.

Proposition 3 (Limiting cases)**.** *Let r_t be defined as in Eq.* (8)*, with $\mathbf{k}_{<t} \in \mathbb{R}^{t-1}$ collecting the entries K_{jt} for $j < t$ and $\tilde{K}_{<t} = K_{<t} + \sigma^2 d \cdot I_{t-1}$.*

(i) ***Orthogonal disagreement.*** *If $K_{jt} = 0$ for all $j < t$, then $r_t = \log(K_{tt} + \sigma^2 d)$.*

(ii) ***Redundant disagreement.*** *If there exists $\boldsymbol{\alpha} \in \mathbb{R}^{t-1}$ such that $\mathbf{k}_{<t} = K_{<t}\,\boldsymbol{\alpha}$ and $K_{tt} = \boldsymbol{\alpha}^\top K_{<t}\,\boldsymbol{\alpha}$ (i.e., row t of K lies in the row span of $K_{<t}$), then*

$$\sigma^2 d \;\le\; \exp(r_t) \;\le\; \sigma^2 d\,\big(1 + \|\boldsymbol{\alpha}\|^2\big). \tag{29}$$

Proof. For (i): $\mathbf{k}_{<t} = \mathbf{0}$ implies that the subtraction in Eq. (8) vanishes, giving $\exp(r_t) = K_{tt} + \sigma^2 d$.

For (ii): substituting $\mathbf{k}_{<t} = K_{<t}\,\boldsymbol{\alpha}$ and $K_{tt} = \boldsymbol{\alpha}^\top K_{<t}\,\boldsymbol{\alpha}$ into Eq. (8) and simplifying yields

$$\exp(r_t) = \sigma^2 d\,\big(1 + \|\boldsymbol{\alpha}\|^2 - \sigma^2 d\;\boldsymbol{\alpha}^\top \tilde{K}_{<t}^{-1}\,\boldsymbol{\alpha}\big). \tag{30}$$

The upper bound follows from $\tilde{K}_{<t} \succeq \sigma^2 d \cdot I_{t-1}$, which gives $\boldsymbol{\alpha}^\top \tilde{K}_{<t}^{-1}\,\boldsymbol{\alpha} \ge 0$. The lower bound follows from $\tilde{K}_{<t}^{-1} \preceq (\sigma^2 d)^{-1} I_{t-1}$, which gives $\sigma^2 d\;\boldsymbol{\alpha}^\top \tilde{K}_{<t}^{-1}\,\boldsymbol{\alpha} \le \|\boldsymbol{\alpha}\|^2$. □

Part (i) recovers the lifelong disagreement reward: with no correlation to the prefix, the episodic correction is inactive and r_t equals the total disagreement at step t, the same signal used by Plan2Explore [5]. Part (ii) shows that when step t is fully redundant with the prefix, the epistemic contribution K_{tt} is removed and $\exp(r_t)$ is confined to a band proportional to the aleatoric scale $\sigma^2 d$. In the common case of exact duplication ($\boldsymbol{\alpha} = \mathbf{e}_j$ for some $j < t$), the band tightens to $[\sigma^2 d,\; 2\sigma^2 d]$, confirming that repeated visits to the same model gap yield a reward near the aleatoric floor regardless of the magnitude of the disagreement.

E.2 Characterisation of the Prefix-Redundancy Term

The prefix-redundancy correction $\mathbf{k}_{<t}^\top \tilde{K}_{<t}^{-1} \mathbf{k}_{<t}$ in Eq. (8) distinguishes CIG from per-step ensemble disagreement. Removing this term recovers the *No Prefix Redundancy* ablation of §4, whose reward reduces to $\log(K_{tt} + \sigma^2 d)$. This subsection asks two questions: (i) where in the state space does the correction activate, and (ii) how does that activation pattern evolve over training? Figure 8 summarises the answers on AntMaze across five training windows.

Frontier selectivity at the step level. The top row of Figure 8 plots the prefix-explained fraction $\mathbf{k}_{<t}^\top \tilde{K}_{<t}^{-1} \mathbf{k}_{<t} / K_{tt}$ against the lifelong disagreement $\log K_{tt}$, the diagonal entry of the kernel that constitutes the first term of Eq. (8). The correction concentrates in the rightmost bins, where lifelong disagreement is high, and collapses to near zero for low-K_{tt} steps. The mechanism is straightforward: the Cauchy–Schwarz inequality on the Gram matrix forces $|K_{jt}| \le \sqrt{K_{jj}\,K_{tt}}$, so when both a prefix step j and the current step t lie in well-learned regions, the off-diagonal entries that feed the quadratic form are bounded by the geometric mean of two small quantities. The prefix-redundancy term therefore activates only when step t sits at the exploration frontier, where K_{tt} is large enough for the off-diagonals to carry non-trivial alignment with the prefix.

This selectivity sharpens as training progresses. Over successive windows the conditional mass migrates upward and to the right: the prefix accounts for 10–25% of the lifelong disagreement at frontier steps by 2.4–2.5M env-steps, while low-disagreement steps remain unaffected. The pattern is consistent with the ensemble concentrating its residual uncertainty on a narrowing frontier whose steps become increasingly correlated with one another.

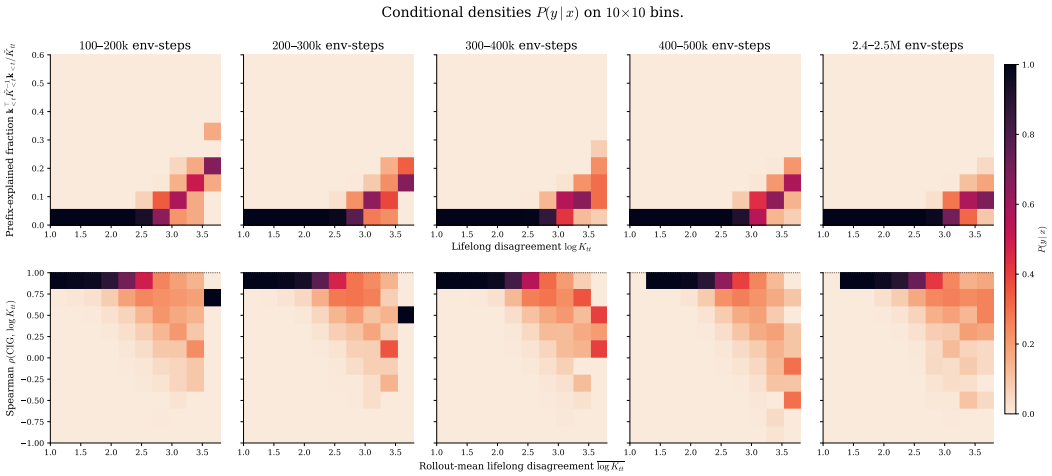


Figure 8: **Prefix-redundancy activation across training (AntMaze-medium).** Each panel shows the column-normalised conditional density $P(y \mid x_{\text{bin}})$ on a 10×10 grid. **Top row:** prefix-explained fraction $\mathbf{k}_{<t}^{\top}\tilde{K}_{<t}^{-1}\mathbf{k}_{<t}/K_{tt}$ (the share of the lifelong disagreement at step t accounted for by the prefix) vs. lifelong disagreement $\log K_{tt}$, the diagonal term of Eq. (8). **Bottom row:** Spearman rank correlation $\rho(\text{CIG}, \log K_{tt})$ between the CIG reward and the lifelong disagreement within each rollout vs. rollout-mean lifelong disagreement $\overline{\log K_{tt}}$. Under the *No Prefix Redundancy* ablation the top-row quantity is identically zero and the bottom-row correlation is identically one; departures from these values reflect off-diagonal structure in K that CIG exploits.

Rank reordering at the rollout level. The bottom row of Figure 8 examines the same effect at rollout granularity, plotting $\rho(\text{CIG}, \log K_{tt})$, the Spearman rank correlation between CIG and the lifelong disagreement $\log K_{tt}$ within each rollout, against rollout-mean lifelong disagreement $\overline{\log K_{tt}}$. Because the ridge $\sigma^2 d$ is constant across steps, $\log K_{tt}$ and $\log(K_{tt} + \sigma^2 d)$ are related by a strictly monotone transform; Spearman ρ is invariant to such transforms, so the correlation is identical whether the reference ranking uses the *Lifelong Only* variant ($\log K_{tt}$, which removes both the prefix correction and the ridge) or the *No Prefix Redundancy* variant ($\log(K_{tt} + \sigma^2 d)$, which retains the ridge but drops the prefix correction).

When rollout-mean lifelong disagreement is low, ρ clusters near one: the imagined trajectory stays in well-learned regions, the prefix-redundancy term is inactive, and CIG ranks steps identically to the lifelong disagreement alone. As $\overline{\log K_{tt}}$ increases, ρ spreads across a wide range, with substantial mass at $\rho < 0.5$ and occasional rank reversals ($\rho < 0$). CIG reorders steps within a rollout only when the rollout visits the frontier, leaving the ranking unchanged in familiar territory. The effect grows over training, consistent with the narrowing frontier producing stronger inter-step correlations that the off-diagonal entries of K capture.

Contrast with E3B×P2E. CIG's frontier selectivity is a structural consequence of deriving both the lifelong and prefix terms from the same kernel K. The Cauchy–Schwarz coupling $|K_{jt}| \le \sqrt{K_{jj}\, K_{tt}}$ ties the magnitude of the correction to the diagonal, forcing it to vanish quadratically when K_{tt} is small. E3B×P2E lacks this coupling. Its E3B factor scores each step against an embedding-space covariance C_t^{-1} built from the encoder ϕ, not from the ensemble's epistemic structure. In well-learned corridors where two states happen to have distant embeddings under ϕ, E3B assigns a large novelty bonus; suppression relies entirely on the P2E factor being small, an indirect cancellation rather than a structural zero.

Formal statement. The quadratic vanishing observed in Figure 8 can be made precise. Define the signal-to-noise ratio at step t as $\epsilon_t = K_{tt}/(\sigma^2 d)$ and let $\epsilon = \max_{1 \le t \le T} \epsilon_t$. Write $r_t^{\text{diag}} = \log(K_{tt} + \sigma^2 d)$ for the *No Prefix Redundancy* reward.

Proposition 4 (Frontier inactivity)**.** *For all $t \le T$,*

$$0 \;\le\; r_t^{\text{diag}} - r_t \;\le\; -\log\bigl(1 - (t-1)\,\epsilon^2\bigr). \tag{31}$$

When $(t-1)\,\epsilon^2 \le 1/2$*, this simplifies to* $r_t^{\mathrm{diag}} - r_t \le 2(t-1)\,\epsilon^2$.

Proof. Lower bound. Because K is a Gram matrix ($K = \frac{1}{M}\Delta^\top\Delta$, where column t of $\Delta \in \mathbb{R}^{Md\times T}$ concatenates the deviation vectors $\boldsymbol{\delta}_k^{(t)}$ across members), K is positive semidefinite. Hence $\tilde{K}_{<t} = K_{<t} + \sigma^2 d \cdot I_{t-1} \succ 0$ and $\mathbf{k}_{<t}^\top \tilde{K}_{<t}^{-1}\mathbf{k}_{<t} \ge 0$, giving $r_t \le r_t^{\mathrm{diag}}$.

Upper bound. Since $\tilde{K}_{<t} \succeq \sigma^2 d \cdot I_{t-1}$, we have $\tilde{K}_{<t}^{-1} \preceq (\sigma^2 d)^{-1}\, I_{t-1}$, so

$$\mathbf{k}_{<t}^\top \tilde{K}_{<t}^{-1}\mathbf{k}_{<t} \;\le\; \frac{\|\mathbf{k}_{<t}\|^2}{\sigma^2 d}. \tag{32}$$

Cauchy–Schwarz on the Gram matrix gives $K_{jt}^2 \le K_{jj}\,K_{tt}$ for all j, t. Summing over the prefix indices and substituting $K_{jj} \le \epsilon\,\sigma^2 d$:

$$\|\mathbf{k}_{<t}\|^2 = \sum_{j=1}^{t-1} K_{jt}^2 \;\le\; K_{tt} \sum_{j=1}^{t-1} K_{jj} \;\le\; (t-1)\,\epsilon\,\sigma^2 d \cdot K_{tt}. \tag{33}$$

Combining Eqs. (32) and (33):

$$\frac{\mathbf{k}_{<t}^\top \tilde{K}_{<t}^{-1}\mathbf{k}_{<t}}{K_{tt} + \sigma^2 d} \;\le\; \frac{(t-1)\,\epsilon\,K_{tt}}{K_{tt} + \sigma^2 d} \;=\; (t-1)\,\epsilon\,\frac{\epsilon_t}{1+\epsilon_t} \;\le\; (t-1)\,\epsilon^2. \tag{34}$$

Writing $r_t^{\mathrm{diag}} - r_t = -\log\!\big(1 - \mathbf{k}_{<t}^\top \tilde{K}_{<t}^{-1}\mathbf{k}_{<t}/(K_{tt}+\sigma^2 d)\big)$ and applying monotonicity of $-\log(1-x)$ yields Eq. (31). The simplified bound uses $-\log(1-x) \le 2x$ for $x \in [0,\, 1/2]$. □

F Extended Experimental Results

F.1 Baselines

This section briefly recaps the intrinsic-reward baselines used as comparisons in our experiments. Each paragraph below summarises only the per-step *reward signal* that defines the method, since this is what differs across baselines and what is directly comparable to CIG; encoder, ensemble, and optimisation details follow the cited papers, and the numerical settings we used are reported in §A.3.

ICM. Intrinsic Curiosity Module learns a state encoder ϕ jointly with an inverse model that predicts the action from $(\phi(s_t), \phi(s_{t+1}))$, so that ϕ retains only the part of the observation that the agent's actions can affect. A forward model $\hat{f}$ is then trained to predict the next-state encoding from $(\phi(s_t), a_t)$, and its squared prediction error is taken as the intrinsic reward:

$$r_t^{\mathrm{ICM}} = \tfrac{1}{2}\,\big\|\hat{f}(\phi(s_t), a_t) - \phi(s_{t+1})\big\|_2^2. \tag{35}$$

The inverse-model objective is what gives ICM its nominal robustness to action-irrelevant pixel noise, but the bonus itself is still a raw forward prediction error and so remains susceptible to genuine stochasticity in the transitions [3].

RND. Random Network Distillation defines novelty against a randomly initialised, *fixed* target network f^*. A predictor network $\hat{f}$ is trained to regress f^* on visited states only, and the residual error is the intrinsic reward:

$$r_t^{\mathrm{RND}} = \big\|\hat{f}(s_t) - f^*(s_t)\big\|_2^2. \tag{36}$$

States the predictor has been trained on yield small error, while under-visited states retain large error, so the bonus operates as a state-level novelty signal that is independent of the agent's actions and of any learned dynamics model [4].

Plan2Explore (Disagreement). Plan2Explore equips a Dreamer world model with an ensemble of M one-step latent predictors $\{f_k\}_{k=1}^M$ that all predict the next-state embedding from the current latent and action. The intrinsic reward is the average predictive variance across the ensemble, taken over the embedding's d dimensions:

$$r_t^{\mathrm{P2E}} = \tfrac{1}{d}\sum_{i=1}^{d} \mathrm{Var}_k\big[f_k(s_t, a_t)_i\big]. \tag{37}$$

Because the ensemble members agree on aleatoric noise but disagree where the world model has not yet generalised, this variance approximates epistemic uncertainty over the next embedding [5, 10]. CIG re-uses exactly this ensemble; the difference is that P2E reads off only its *marginal* disagreement at each step, whereas CIG conditions on prior steps within the rollout.

E3B. Elliptical Episodic Bonus (E3B) reuses ICM's inverse-dynamics objective to train an encoder ϕ, but its bonus is not a prediction error. Within a single episode it maintains a running inverse covariance $C_t^{-1} \in \mathbb{R}^{e\times e}$ of the visited embeddings with e being the embedding dimensionality, initialised to $(1/\lambda)\, I$, and scores each step with the elliptical (Mahalanobis-style) bonus

$$r_t^{\text{E3B}} \;=\; \phi(s_t)^\top\, C_t^{-1}\, \phi(s_t), \tag{38}$$

which is large when $\phi(s_t)$ points in a direction the current episode has not yet covered. After each step C_t^{-1} is updated by a Sherman–Morrison rank-1 update and is reset at episode boundaries. Unlike ICM and RND, novelty is therefore computed *within* the current episode rather than against a global predictor [11].

E3B $\times$ P2E. This baseline composes E3B's per-episode elliptical bonus with Plan2Explore's lifelong ensemble disagreement multiplicatively,

$$r_t^{\text{E3B}\times\text{P2E}} \;=\; r_t^{\text{E3B}} \cdot r_t^{\text{P2E}}, \tag{39}$$

so the agent is rewarded only where *both* signals fire: the state must be novel relative to what this episode has already seen (E3B) *and* lie in a region the world-model ensemble disagrees about (P2E). Following Henaff et al. [11], the encoder embedding is ℓ_2-normalised before entering the elliptical update so that the per-step E3B bonus is bounded and the multiplicative product is well-scaled. This combination serves as a simple non-CIG baseline that already mixes episodic novelty with epistemic uncertainty.

APT. Active Pre-Training estimates the entropy of visited states non-parametrically with a particle-based k-nearest-neighbour estimator on the deterministic world-model state. For a particle h_t in the current batch, the reward is the log of the average distance to its k nearest neighbours $\mathcal{N}_k(h_t)$, which approximates state entropy up to an additive constant:

$$r_t^{\text{APT}} = \log\left(1 + \tfrac{1}{k} \sum_{h_j \in \mathcal{N}_k(h_t)} \|h_t - h_j\|_2\right). \tag{40}$$

Unlike the other baselines, the APT bonus directly rewards visiting points that are far from the rest of the batch in feature space, so the agent is pushed towards under-represented regions of the state distribution rather than towards regions where some predictor is currently inaccurate [28].

F.2 Per-Episode Entropy and Downstream Success on Minigrid

The sample-efficiency results in §4.2 leave open whether CIG's advantage arises from visiting more diverse states per episode or from directing visits toward epistemically informative regions. To disentangle these factors, we compute the entropy of the empirical visit-distribution over grid cells for each episode, average over episodes in the final 10% of training, and plot the result against final cumulative successes in Figure 9.

Per-method entropy ordering. Across all six panels, a consistent ordering emerges. E3B and APT achieve the highest per-episode entropy: E3B's episodic novelty bonus pushes the agent to visit cells it has not yet seen within the current episode, while APT's k-nearest-neighbour entropy estimate directly rewards uniform state coverage. CIG produces moderately higher entropy than P2E on every task, consistent with the log-determinant objective penalising within-rollout redundancy and thereby encouraging the policy to spread visits across distinct regions. RND and ICM cluster at lower entropy, reflecting their tendency to concentrate on narrow high-novelty frontiers rather than distributing visits broadly.

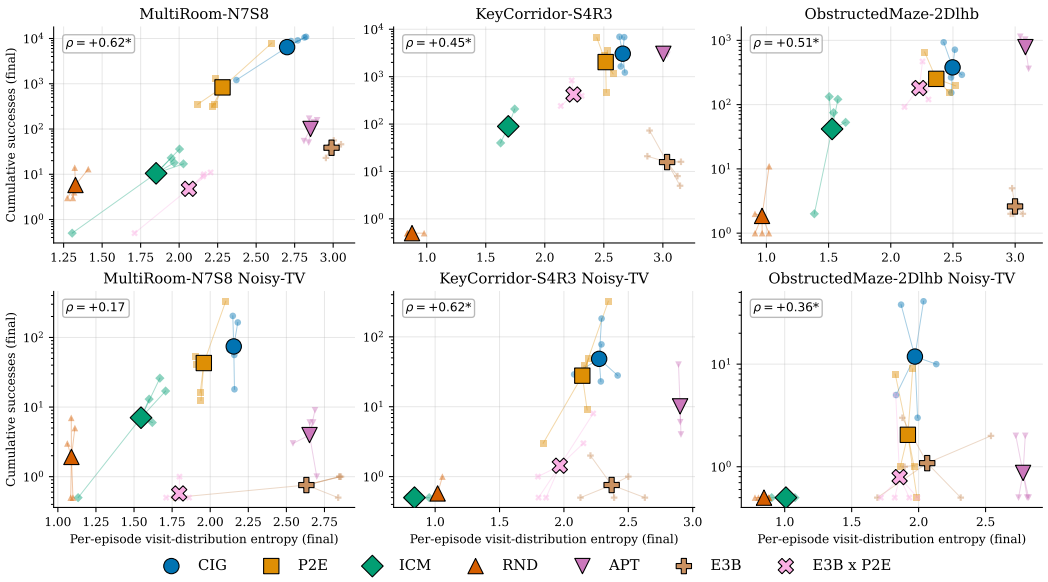


Figure 9: Per-episode visit-distribution entropy (x-axis) versus final cumulative successes (y-axis, log scale) for seven methods across three MiniGrid tasks in clean (top) and Noisy-TV variants (bottom). Large markers denote per-method means; small markers show individual seeds. Each panel also reports the Spearman rank correlation ρ.

Entropy does not predict success. Despite their consistently high entropy, E3B and APT occupy opposite ends of the success axis depending on the task. E3B achieves the highest or near-highest entropy on five of six panels yet accumulates among the lowest cumulative successes everywhere, including the clean tasks. Its episodic novelty signal distributes visits uniformly but does not prioritise states where the dynamics model carries residual error, so broad coverage does not translate into informative data collection. APT follows a different pattern: its high entropy translates into strong performance on clean ObstructedMaze 2DlhB, where the task requires reaching many spatially distributed objects and uniform coverage aligns well with the downstream metric. On the remaining five panels, APT's entropy advantage yields no corresponding success advantage. Under distraction (bottom row), the dissociation is stark: APT retains the highest entropy on ObstructedMaze 2DlhB Noisy-TV (mean ≈ 2.75) but accumulates near-zero successes, because the additional coverage is consumed by distractor states rather than task-relevant regions.

Targeted exploration as the driving factor. CIG's entropy is lower than APT's and E3B's on every task, yet CIG achieves the highest cumulative successes on five of six panels and remains competitive on the sixth (clean ObstructedMaze). The gap between CIG and P2E is instructive: CIG produces only modestly higher entropy than P2E, so the performance difference between them cannot be attributed to broader coverage alone. Instead, the prefix correction in Eq. (8) redirects visits away from uncertainty directions already probed earlier in the rollout, converting a small entropy increase into a larger gain in the diversity of epistemic information collected per episode. The pattern across all six panels supports the same conclusion: in MiniGrid, targeted reduction of dynamics-model uncertainty dominates raw state-coverage breadth as a predictor of downstream task success.

F.3 Exploration selectivity on Puzzle 3×3

Figure 10 reports total button flips for each method on clean Puzzle 3×3. While CIG, APT, and ICM achieve similar final unique-configuration coverage (Figure 4), APT accumulates roughly $10\times$ more flips over the same step budget. APT's k-nearest-neighbour entropy proxy rewards visiting any under-represented state, driving the policy to press buttons indiscriminately; many flips revisit previously discovered configurations. CIG concentrates interactions on transitions whose ensemble disagreement is orthogonal to the rollout prefix, yielding new configurations at a higher rate per flip.

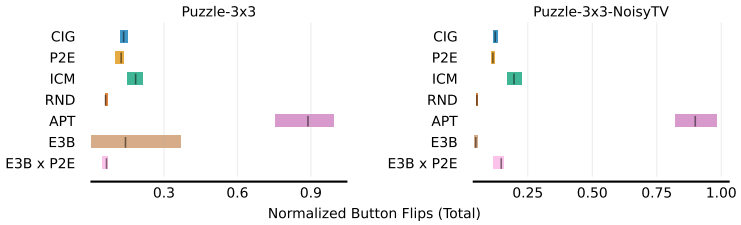


Figure 10: Total button flips over training on Puzzle 3×3. CIG, APT, and ICM reach comparable final unique-configuration coverage (Figure 4, top center), but APT accumulates roughly an order of magnitude more button flips than CIG.

F.4 Extended Ablation Results

Table 5 complements Table 1 by reporting the median and mean alongside the IQM. All three statistics rank the variants identically in both domains. In continuous control, CIG's CIs separate from those of both variants on every statistic, while *No Prefix Redundancy* and *Lifelong Only* separate only on the mean, consistent with the small effect of the ridge in this domain. In MiniGrid, CIG's CIs overlap with those of *No Prefix Redundancy*.

Table 5: Full aggregate statistics for the reward-term ablations of Table 1, with 95 % stratified bootstrap CIs over five seeds per (variant, task) pair. Continuous control: AntMaze, AntMaze Noisy-TV, Cube-Triple, Puzzle 3×3, Puzzle 3×3 Noisy-TV, Scene. MiniGrid: MultiRoom N7S8 (clean, Noisy-TV), KeyCorridor S4R3 (Noisy-TV), ObstructedMaze 2DlhB (clean, Noisy-TV). All three statistics rank the variants identically in both domains.

Variant	Median [95 % CI]	IQM [95 % CI]	Mean [95 % CI]
Continuous control			
CIG	**0.956** [0.943, 0.977]	**0.972** [0.953, 0.987]	**0.957** [0.940, 0.973]
No Prefix Redundancy	0.804 [0.761, 0.848]	0.812 [0.778, 0.844]	0.820 [0.792, 0.847]
Lifelong Only	0.755 [0.731, 0.782]	0.769 [0.740, 0.799]	0.758 [0.730, 0.785]
MiniGrid			
CIG	**0.473** [0.318, 0.668]	**0.472** [0.259, 0.705]	**0.508** [0.378, 0.639]
No Prefix Redundancy	0.359 [0.259, 0.507]	0.385 [0.269, 0.515]	0.408 [0.320, 0.501]
Lifelong Only	0.212 [0.049, 0.317]	0.104 [0.062, 0.203]	0.170 [0.116, 0.231]

G Related Work

Exploration under sparse or absent extrinsic reward relies almost exclusively on intrinsic motivation, where the agent constructs its own reward from model uncertainty, state novelty, or information gain [1, 2]. The resulting design space is organised by what context the reward at step t conditions on: lifelong experience $\mathcal{D}$, the trajectory prefix $s_{<t}$, or both (§2.2). Existing methods treat these contexts as alternatives or combine them heuristically; CIG derives a single reward that conditions on both.

Lifelong intrinsic rewards. Count-based methods generalise visitation counts to continuous spaces via pseudo-counts or density models [17, 31–33]. Prediction-error methods reward novelty through forward-model discrepancies: ICM [3] learns an inverse-dynamics model to obtain representations free of action-independent noise, and AMA [34] extends this to also filter action-dependent stochasticity. RND [4] avoids the noisy-TV problem differently by distilling a fixed random network. Other lifelong approaches maximise state entropy [35, 28], use metric-based bonuses [36], or reward information gain about dynamics parameters [37]. Ensemble-based methods approximate epistemic uncertainty through forward-model disagreement [10, 38, 39]. All these methods assign rewards per state independently, ignoring correlations within a trajectory (§3).

Episodic and hybrid intrinsic rewards. Episodic methods define novelty relative to states visited within the current episode [18, 19, 40]. E3B [11] is closest to CIG among these: it builds a covariance

matrix over inverse-dynamics embeddings using Sherman–Morrison updates, so that each step's reward reflects how much the current state extends the subspace of earlier states. CIG's Cholesky factorisation of $\mathbf{K}$ plays an analogous role, but E3B is purely episodic: its covariance is built from inverse-dynamics features and reset every episode, whereas CIG conditions on lifelong experience through the ensemble parameters, with off-diagonal entries correcting for within-rollout redundancy given that context. Since episodic bonuses alone ignore lifelong progress, several methods multiply or gate an episodic indicator by a lifelong bonus [13, 20, 14, 21, 41–43], and empirical studies confirm that the combination improves over either component alone [12, 44]. The episodic components in these methods typically rely on discretised matching (hashing or binned embeddings) that degrades in continuous control domains where states rarely recur in the same bin. More fundamentally, all such hybrids set their combination weights heuristically, whereas CIG arrives at a hybrid signal as a consequence of a single trajectory-level information-gain objective.

Bayesian optimal experimental design. CIG's log-determinant surrogate (Eq. 7) instantiates D-optimal design [25] for deep ensembles. In the Bayesian Gaussian setting, D-optimality coincides with maximising expected information gain [16, 22, 45], the structural fact underlying our passage from the mutual information of Eq. (1) to the Gaussian bound of Eq. (4). Rainforth et al. [23] survey computational advances in sequential Bayesian experimental design, including the chain-rule decomposition of EIG that parallels our Eq. (1). Within RL, Mehta et al. [46] framed model-based exploration as a BOED problem under GP dynamics. TIP [15] extended this to trajectory-level joint information gain and is the closest antecedent to CIG in motivation.

Trajectory-level information gain in RL. TIP computes joint EIG via log-determinants of GP covariance matrices, naturally penalising redundant state visits, but its reliance on Gaussian-process dynamics restricts it to low-dimensional MDPs. CIG arrives at an analogous objective through ensemble disagreement vectors in a learned latent space, avoiding GP inference entirely; the trace reduction (§3.2) further addresses capacity saturation from finite ensemble size, a challenge absent from TIP's exact GP formulation. Plan2Explore [5] also uses ensemble disagreement in a learned world model but assigns per-step variance rewards, treating each step independently; Caron et al. [47] provide theoretical guarantees for such per-step information-gain bonuses but likewise do not model within-trajectory correlations. CIG and Plan2Explore share an ensemble architecture but differ in objective: per-step variance versus trajectory-level information gain (Eq. 7), with the off-diagonal entries of $\mathbf{K}$ capturing correlations that a per-step reward cannot represent. No existing method derives a per-step intrinsic reward that jointly conditions on lifelong experience and the trajectory prefix from a single information-theoretic objective that scales to deep models; §3 develops the CIG surrogate that achieves this.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim that we present a new intrinsic reward that performs better than the baselines on many environments. Our methods section derives this reward and our experiments show that indeed it performs much better on the selected environments and metrics.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations shortly in the conclusion section. Since CIG shares its architecture with Plan2Explore, but differs in the reward computation, we did only discuss the limitations that are unique to CIG and not general to ensemble-based intrinsic rewards. The main limitation is as stated scalability to longer horizons. Additionally, we clearly state the assumptions that have made in the methods section and the direct consequences of these assumptions are discussed there and in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper is not a theory paper such that the derivations built upon well-known bounds and identities. The appendix also discusses some of the implications of the made assumptions and states the proof for the implications.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The used hyper parameters and used environments are described in the appendix. We also have pseudo-code in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We deliver the code to reproduce the results with detailed instructions how to setup the environment and run the code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We state the we use 5 seeds for all of our experiments. We describe the environments and hyper parameters in the appendix. Additionally, there is code that can be used to trace the method. Our code mostly follows DreamerV2, a well-known model-based RL agent which additionally lessens the burden of understanding how the results are produced.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We follow the recommendations of [29] and report IQM with stratified-bootstrap confidence intervals, the mean and median of the results. This is the community accepted way to get statistical significant results in RL.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: In the appendix we describe the machines that have been used to run the experiments and their typical runtime.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: -

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is foundational research on intrinsic reward mechanisms for reinforcement learning and is not tied to a specific application or deployment. We do not foresee a direct path from improved intrinsic motivation methods to particular positive or negative societal outcomes beyond those already associated with general advances in reinforcement learning.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The research is basic research in exploration in reinforcement learning. There is no immediate way to misuse the models or data.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The original owners are cited in A.7

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The derived environments are clearly documented in the codebase.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: -

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: -

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: -

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.