Can In-Context Learning Support Intrinsic Curiosity?

Eric Elmoznino[*,1,2], **Sangnie Bhardwaj**[*,2,3], **Johannes von Oswald**[1], **Rajai Nasser**[1], **Blaise Agüera y Arcas**[1], **João Sacramento**[1], **Rif A. Saurous**[1], **Guillaume Lajoie**[1,2]

[1]Google – Paradigms of Intelligence Team, [2] Mila & Université de Montréal [3]Google DeepMind

Abstract

Effective machine learning depends not only on how we model data, but also on what data we choose to collect. While large sequence models have revolutionized data modeling, the problem of automated data selection, or "intrinsic curiosity", remains a significant challenge. Classic approaches incentivize exploration by rewarding an agent based on its "learning progress", which measures how much a newly acquired observation improves a world model's predictive ability. However, evaluating these rewards traditionally requires expensive inner loops of gradient descent updates within each trajectory, rendering them computationally impractical at scale. In this work, we investigate whether the emergent in-context learning (ICL) capabilities of sequence models can eliminate this bottleneck by serving as immediate, update-free world models. Specifically, we evaluate whether an exploration policy can be trained to maximize learning progress, using solely the prediction errors and counterfactual context manipulations of an in-context learner. We first prove that in general Markov decision processes, this is in fact impossible in an unbiased way: the resulting intrinsic rewards either suffer from nuisance terms that bias their estimation of true learning progress, or they cannot be implemented using an in-context learner's prediction errors. Conversely, we prove a positive result for a broad subclass of non-temporal settings, encompassing active learning and Bayesian Experimental Design: here, ICL-derived rewards successfully bound and asymptotically converge to the true learning progress. We corroborate our theory with controlled experiments across continuous and symbolic environments, demonstrating that our ICL-driven framework successfully trains curious data-collection policies that explore optimally.

1 Introduction

How should an agent collect data in an unknown environment to ensure optimal exploration? One foundational approach dictates that the data should maximally improve the agent's model of its environment (Schmidhuber, 1991a; Lindley, 1956). Such a model can subsequently support arbitrary downstream tasks, either through explicit planning or by having acquired robust representations. To this end, prior work has proposed utilizing "intrinsic rewards" to drive the collection of data, which are defined strictly as a function of the trajectory of actions and observations collected by a policy, and assume no extrinsic "task".

Given a Bayesian prior on the world, an optimal intrinsic reward is Bayesian information gain (BIG), which measures the expected bits gained about the true environment's dynamics with every action (Itti and Baldi, 2009; Lindley, 1956). However, BIG is difficult to compute in practice because it requires explicitly parameterizing the environment's dynamics and performing intractable Bayesian inference. While tractable prediction-based objectives have been proposed as alternatives (e.g., Schmidhuber, 1991a), they face two major hurdles: (1) their theoretical relationship to BIG remains poorly understood and (2) computing them requires expensive gradient descent updates to a world model that have slow credit assignment.

Overcoming this second bottleneck requires fast and data-efficient learning mechanisms. In-context learning (ICL) has emerged as a highly effective paradigm for addressing this problem: sequence models can act as amortized predictors that *implicitly* approximate the Bayesian posterior predictive in a single forward pass,

*Equal contribution. Correspondence to: {eric.elmoznino,guillaume.lajoie}@gmail.com, sangnie@google.com

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

bypassing the need for explicit inference. This capability is hypothesized to partly drive the success of large language models (Radford et al., 2019; Brown et al., 2020), and has seen great success with Prior-Fitted Networks (PFNs) (Nagler, 2023; Müller et al., 2022) such as TabPFN (Hollmann et al., 2022) which are pretrained on a large prior over datasets to perform amortized learning on a new dataset at inference time.

In this work, we use ICL for evaluating prediction-based intrinsic curiosity rewards. We examine the degree to which data collected by a policy improves a world model learned purely in-context, and reward the policy based on this improvement. This raises several fundamental questions: First, is this possible with ICL, and for which intrinsic rewards? Second, does the resulting reward approximate BIG, and if so, under what circumstances? To answer these questions, we make the following contributions:

1. By formalizing in-context learners as implicit Bayesian predictors, we prove novel mathematical relationships between prediction-based intrinsic rewards and BIG, which were previously unknown.
2. We provide a negative result for general Markov Decision Processes (MDPs), proving that prediction-based intrinsic rewards are biased estimators of BIG. Further, while some of these rewards can be implemented using ICL, we show that others, such as classic learning progress (Schmidhuber, 1991a), cannot.
3. In contrast, we establish a positive result for Bayesian Experimental Design (BED) settings, demonstrating that several intrinsic rewards can be approximated using ICL, and that it is possible to asymptotically approximate BIG for long trajectories with a particular reward structure.
4. We conduct experiments that corroborate our theoretical findings, demonstrating the practical viability of this ICL approach to curiosity.

2 Problem Setting and Notation

Environment. We consider a Bayes-Adaptive MDP (BAMDP) with states s_t, actions a_t, horizon T, and a latent environment parameter $\theta \sim p(\theta)$ fixed throughout an episode. Conditional on θ, the environment's dynamics are Markov and stationary: $s_{t+1} \sim p(\cdot \mid s_t, a_t, \theta)$. Throughout, we write $h_t \coloneqq (s_1, a_1, \ldots, s_{t-1}, a_{t-1})$ for the trajectory history just before s_t.

Intrinsic curiosity objective. Our goal is to train a policy $\pi_\phi(a_t \mid h_t, s_t)$ under a "meta-RL" setting (Duan et al., 2016) in which episodes are rolled out from environments sampled from $p(\theta)$. Crucially, there is no notion of "task" or "extrinsic" reward. We instead consider "intrinsic" rewards that are functions of only the trajectory $(s_1, a_1, \ldots, s_T)$, and which are ideally maximized when the trajectory is highly informative about the latent parameters θ governing the environment's dynamics. By training π_ϕ across a broad distribution of environments sampled from $p(\theta)$, we aim for it to optimally explore a *new* environment sampled from this distribution at inference time. In information theoretic terms, we consider an "optimal" intrinsic reward as one that maximizes the per-step mutual information between an observed state and the latent environment parameters, conditioned on the history. In the literature, this is called Bayesian information gain (BIG) or Bayesian surprise (Itti and Baldi, 2009; Lindley, 1956):

$$\text{BIG} \coloneqq I(s_t; \theta \mid h_t). \tag{1}$$

In-context learning prediction errors. We consider whether a sequence model ρ can be used to approximate BIG, using only its prediction errors and manipulations of its context. We assume that ρ has been pretrained offline to perform next-state predictions on trajectories of BAMDPs sampled from prior $p(\theta)$ with uniform action selection — ρ is then frozen when training the policy. We further suppose that ρ has been trained with sufficient capacity and data, such that it has learned to emit the Bayesian posterior-predictive:

$$\rho(s_t \mid h_t) = \int p(s_t \mid s_{t-1}, a_{t-1}, \theta) p(\theta \mid h_t) d\theta. \tag{2}$$

This perspective of in-context learners as amortized Bayes-optimal predictors has been formalized in substantial prior work (Grau-Moya et al., 2024; Mikulik et al., 2020; Xie et al., 2022; Binz et al., 2024; Nagler, 2023; Müller et al., 2022), and it serves as a starting point for our investigations. In our setting, ρ can be seen as a meta-learner that *implicitly* fits environment parameters θ in-context and exposes their predictions for novel transitions, all within a single forward pass. The speed of in-context learners makes them particularly promising for the purposes of optimizing certain intrinsic curiosity objectives, which, as we will see next, require us to repeatedly assess the learning progress of a world model over the course of a trajectory.

3 Prior Work on Intrinsic Curiosity

We briefly describe approaches to intrinsic curiosity relevant to our work — Aubret et al. (2023) provides a recent review. We discuss intrinsic rewards that are less directly related to our work in Appendix A. We assume $p(\cdot \mid \cdot, \hat{\theta}_t)$ is a predictive world model that has been trained on subtrajectory $(a_{1:t-1}, s_{1:t})$.

Surprisal. A common heuristic for encouraging exploration is to maximize a world model's surprisal, seeking observations for which it has high prediction error. The surprisal reward is $r_t \coloneqq -\log\, p(s_t \mid s_{t-1}, a_{t-1}, \hat{\theta}_{t-1})$. This approach has been used at least as far back as Schmidhuber (1991b) and is simple but surprisingly effective (Burda et al., 2019; Levy et al., 2025; Hester and Stone, 2012; Chentanez et al., 2004). However, surprisal fails to distinguish between two sources of information: epistemic (i.e., learnable) and aleatoric (i.e., noise). The latter can be detrimental: a policy seeking surprisal will learn to sit in front of a "noisy TV", even if these observations contain no new useful information about the environment (Schmidhuber, 1991a). Thus, while surprisal excels in deterministic environments, it fails in stochastic settings (Burda et al., 2019).

Learning progress. Another principled approach to intrinsic curiosity is to maximize "learning progress", introduced in Schmidhuber (1991a) and further explored in (Schmidhuber, 2009, 2010; Storck et al., 1995; Oudeyer et al., 2007; Oudeyer and Kaplan, 2008, 2007; Lopes et al., 2012; Azar et al., 2019). It aims to reward a policy based on the degree to which the data that it acquires improves the predictive ability of a world model. The learning progress intrinsic reward is $r_t \coloneqq \log\, p(s_t \mid s_{t-1}, a_{t-1}, \hat{\theta}_t) - \log\, p(s_t \mid s_{t-1}, a_{t-1}, \hat{\theta}_{t-1})$, i.e. the improvement in the model's ability to predict s_t *after* the transition to it has been observed. $p_{\hat{\theta}_t}$ is often parameterized using a recurrent neural network updated online with gradient descent. Unlike surprisal, it assigns zero reward to observations that only contain aleatoric noise, since they do not improve the model's predictive ability.

In Section 4, we will show that while learning progress has desirable exploration properties, the need to evaluate a model's prediction error on transitions *that it has already seen* makes it non-implementable using an in-context learner's prediction errors, for general environments. Azar et al. (2019) introduced an alternative that instead evaluates improvements in prediction error on *future* transitions $(s_{t+K-1}, a_{t+K-1}, s_{t+K})$, rather than on the current one. Section 4 will show that this alternative — along with a variant that considers improvements on *all* future transitions — *can* be computed using an in-context learner's prediction errors.

Bayesian information gain. Alternatively, we can seek data that maximally changes a model's *belief* over possible environments — an approach termed Bayesian information gain (BIG) or Bayesian surprise (Sun et al., 2011; Little and Sommer, 2013; Itti and Baldi, 2009; Houthooft et al., 2016; Stadie et al., 2015; MacKay, 1992; Fedorov, 1972; Houlsby et al., 2011; Lindley, 1956). The predictive model now considers a distribution over possible environment dynamics, with initial prior $p(\hat{\theta})$. Sun et al. (2011) show that the optimal exploration strategy is to maximize the Kullback-Leibler divergence between the prior and posterior over environment parameters after having observed a novel transition: $r_t \coloneqq KL(p(\hat{\theta} \mid h_t, s_t) \,||\, p(\hat{\theta} \mid h_t))$, where $p(\hat{\theta} \mid h_t, s_t) \propto p(\hat{\theta} | h_t) p(s_t | h_t, \hat{\theta})$. Intuitively, this objective encourages the agent to reduce its uncertainty about the environment dynamics as quickly as possible. Furthermore, if the initial prior $p(\hat{\theta})$ is equal to the environment's true prior distribution over latents $p(\theta)$, this reward is mathematically equivalent in expectation to $I(s_t; \theta \mid h_t)$ from Equation (1).

While optimal in terms of yielding the fastest convergence of $p(\hat{\theta} \mid h_t, s_t)$ towards the true environment dynamics θ, BIG poses significant implementation challenges. It requires a model to do Bayesian inference over a space of possible environments — an operation that is generally intractable and difficult to approximate. Moreover, this approach assumes the model can *explicitly* parameterize the hypothesis space over θ, which is challenging in rich environments with unknown latent structure. In Section 4, we will take BIG as the theoretically-optimal objective for intrinsic curiosity, and we will evaluate the degree to which more tractable prediction-based rewards computed with in-context learners can approximate it.

4 Computing Intrinsic Rewards Using In-Context Learners

We now ask whether an in-context learner ρ can support intrinsic curiosity rewards that approximate BIG. Section 4.1 writes candidate rewards in ρ's interface; Section 4.2 shows none can identify BIG in general BAMDPs at any finite horizon, and that the asymptotic limits required to escape this barrier

are practically out of reach; Section 4.3 restores tractability under a structural restriction, namely that of Bayesian Experimental Design (BED).

4.1 Predictive Rewards

We retain the BAMDP setup of Section 2 and additionally write $h_{t'\setminus t}$ for the history $h_{t'}$ with s_t replaced by a "mask" token — a counterfactual manipulation of ρ's context used by learning-progress-style rewards.

We assume: (i) Markovian, stationary dynamics given θ, (ii) ρ is exactly Bayesian over θ, (iii) the masked trajectory $h_{t'\setminus t}$ removes only s_t's contribution and ρ remains exactly Bayesian on it, (iv) posterior consistency: $p(\theta \mid h_t) \to \delta_{\theta^*}$ as $t \to \infty$, (v) actions are policy-generated; future actions do not update the posterior over θ, (vi) mixing given θ: $I(s_t; s_{t+k} \mid \theta, h_t, a_{t:t+k-1}) \to 0$ as $k \to \infty$. Detailed treatments, including the consequences of approximation, masked-input training, and non-mixing dynamics, are in Appendix B.

We want π_ϕ to collect observations that are maximally informative about the world, and we investigate how to maximize the stepwise Bayesian information gain (BIG) contributed by a new observation s_t: $I(s_t; \theta \mid h_t)$. Importantly, we do not allow explicit access or manipulation of θ, and consider intrinsic rewards reviewed in Section 3 that are built off of state prediction errors, expressed in terms of ρ when possible. Our theoretical results apply to a general class of rewards, including the following common ones:

$$r_t^{\text{sur}} := -\log \rho(s_t \mid h_t), \tag{3}$$

$$r_t^{\text{dl}} := \log \int p(s_t \mid s_{t-1}, a_{t-1}, \theta) p(\theta \mid h_t, s_t) d\theta - \log \rho(s_t \mid h_t). \tag{4}$$

Here, r^{sur} is classical predictive surprisal. r^{dl} is a learning progress reward based on description length (dl) reduction (Schmidhuber, 1991a, 2009, 2010), which asks how much a just-observed transition would improve a model's predictions if included in its posterior. Adding to this list, we propose a novel reward:

$$r_t^{\text{sum}} := \sum_{t'=t+1}^{T} \left[\log \rho(s_{t'} \mid h_{t'}) - \log \rho(s_{t'} \mid h_{t'\setminus t}) \right], \tag{5}$$

where T is the trajectory length. r^{sum} is one of our core contributions, and experiments show it performs well (Section 5). It telescopes over the remaining trajectory and is an extension of the NDIGO reward of Azar et al. (2019), which measured the difference in predictive log-likelihoods for a single future observation K steps in the future. We refer the reader to Appendix C.6 for a treatment of the NDIGO reward. We illustrate our setting in Figure 1, using r^{sum} as an example.

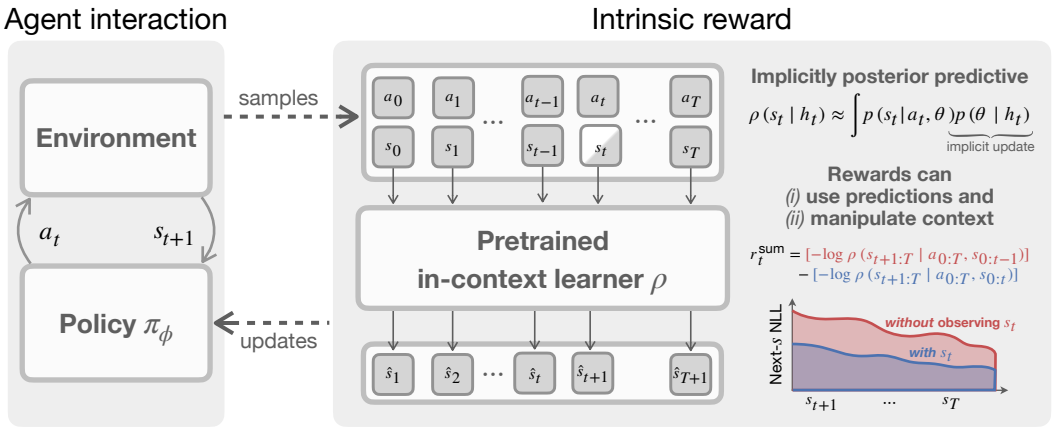

Figure 1: Our method involves using a pretrained in-context learner ρ to construct intrinsic curiosity rewards for a policy π_ϕ. Trajectories unrolled by the policy are passed to the in-context learner, and the reward can be any function of the resulting observation prediction errors on manipulated sequence contexts. We give an example for the reward r^{sum}, which measures the improvement in future prediction errors when a particular state is observed compared to when it is masked.

4.2 Limits on Intrinsic Rewards From In-Context Learners

We first characterize the class of rewards easily implementable with in-context learners ρ:

Definition 1 (Class $\mathcal{F}$)**.** $\mathcal{F}$ is the class of arbitrary functions of finitely many predictive likelihoods of ρ at conditioning subsets of the actual trajectory:

$$\mathcal{F} = \Big\{ r_t = f\big((\rho(X_i \,|\, Y_i))_{i\in I}\big) : X_i \text{ observation}, Y_i \subseteq \text{trajectory}, \forall i \in I, \text{ and } f : [0,1]^{|I|} \to \mathbb{R} \Big\}.$$

At any finite horizon T, r^{sur} and r^{sum} belong to $\mathcal{F}$. r^{dl} does not: the integral in Equation (4) requires an explicit Bayesian update of ρ's posterior — outside ρ's predictive interface in general (see Appendix D). We return to r^{dl} in Section 4.3, where it *does* admit a ρ-tractable form for a subclass of environments. Returning to our goal of maximizing BIG, no finite-horizon reward in $\mathcal{F}$ can be an unbiased estimator:

Theorem 1 (Impossibility of BIG identification in $\mathcal{F}$)**.** *Let $\mathcal{M}$ be the class of BAMDPs satisfying (i)–(v) above, whose transition kernels are neither deterministic nor independent across time given θ. Then for every $r_t \in \mathcal{F}$, there exists $M \in \mathcal{M}$ with $\mathbb{E}^M[r_t \,|\, h_t] \neq I(s_t;\theta \,|\, h_t)$.*

The proof (Appendix C.2) constructs two BAMDPs whose priors share enough leading moments to match $\mathbb{E}[r_t]$ for any *finite* $r_t \in \mathcal{F}$ while their expected BIG differs. We now decompose r^{sur} and r^{sum} against $I(s_t;\theta \,|\, h_t)$ to locate the bias.

Theorem 2 (Decomposition of r^{sur})**.** *Under (i)–(iii),*

$$\mathbb{E}[r_t^{\mathrm{sur}} \,|\, h_t] = I(s_t;\theta \,|\, h_t) + \underbrace{\mathrm{H}(s_t \,|\, h_t,\theta)}_{\text{aleatoric entropy}}.$$

$\mathrm{H}(s_t \,|\, h_t,\theta) = \mathrm{H}(s_t \,|\, s_{t-1},a_{t-1},\theta)$ is non-negative and does not vanish as θ becomes identified.

The proof is in Appendix C.3. Theorem 2 highlights the classic "noisy TV problem", where an agent seeks sources of unpredictable noise.

Theorem 3 (Decomposition of r^{sum})**.** *Under (i)–(iii) and (v),*

$$\mathbb{E}[r_t^{\mathrm{sum}} \,|\, h_t,a_{t:T-1}] = I(s_t;\theta \,|\, h_t) + \underbrace{I(s_t;s_{t+1} \,|\, \theta,h_t,a_t)}_{\text{one-step "abductive"}} - \underbrace{I(s_t;\theta \,|\, h_t,a_{t:T-1},s_{t+1:T})}_{\text{"residual"}}.$$

Both terms are non-negative. Under (iv), the residual vanishes as $T \to \infty$; the abductive does not.

The proof is in Appendix C.4. The *abductive* term captures kernel-mediated coupling between s_t and the next observation s_{t+1} that is not explained by their common dependence on θ. Similarly, a general treatment of rewards involving log-ratios lead to signal/abductive/residual decomposition (Appendix C.5). Notably, applied to NDIGO, it produces an analogous decomposition whose abductive and residual both fail to vanish.

Beyond $\mathcal{F}$: infinite-horizon rewards. Taking $T \to \infty$ to drive r^{sum}'s residual to zero already pushes r^{sum} outside the finite-horizon class $\mathcal{F}$ of Theorem 1. Are there infinite-horizon rewards built from ρ's predictives that recover BIG in general BAMDPs? Consider the generalization of r^{sum} with a gap parameter K separating s_t from the predicted block, $r_t^{\mathrm{gap}}(K) := \log \rho(s_{t+K:T} \mid h_t,s_t,a_{t:T-1}) - \log \rho(s_{t+K:T} \mid h_t,a_{t:T-1})$. Under the mixing assumption (vi), r^{gap} recovers BIG in the iterated limit $\lim_{K\to\infty}\lim_{T\to\infty}$ (Corollary 6.1), and this double limit is structurally necessary — any log-ratio reward universally identifying BIG must place all weight at infinite gap with blocks of unbounded size (Theorem 7). We use r^{gap} as an analytical tool only: it is not feasibly tractable using in-context learners (see below; details in Appendix C.7).

Summary. The section identified compounding obstructions to building ρ-based estimators of BIG:

- *Interventional rewards* — those requiring modifications of ρ's posterior — lie outside ρ's standard interface. The canonical example, r^{dl} (Equation (4)), folds the just-observed s_t into the posterior over θ before re-evaluating the kernel at (s_{t-1},a_{t-1}), and is not implementable through ρ in general BAMDPs.
- Among ρ-implementable rewards in $\mathcal{F}$, no finite-horizon estimator identifies BIG (Theorem 1); r^{sur} and r^{sum} exemplify the bias structure, the former acquiring an aleatoric "noisy TV" offset (Theorem 2) and the latter a persistent one-step abductive (Theorem 3).
- Recovery of BIG via a reward related to $\mathcal{F}$ requires both sequence length $T \to \infty$ and marginalizing gap $K \to \infty$, as exemplified by the theoretical reward r^{gap} (Corollary 6.1 and Theorem 7). The latter is a substantial obstacle: significant unobserved intermediate states $s_{t+1:t+K-1}$ that ρ must marginalize over are poorly supported in sequence models (Appendix B.3).

In sum, we argue there is no practical reward easily implementable with an in-context learner ρ that converges to BIG in general BAMDPs. Moreover, r^{sum} stands as the best ρ-implementable approximation of BIG — scaling naturally with trajectory length but nevertheless retaining an irreducible abductive bias. We now consider BED, in which several of these obstructions vanish.

4.3 Bayesian Experimental Design Setting

We now consider *Bayesian Experimental Design* (BED) environments, where the transition kernel satisfies $p(s_t \mid s_{t-1}, a_{t-1}, \theta) = p(s_t \mid a_{t-1}, \theta)$ — the effect of action a_t depends on θ but not on the current state. This covers any sequential experiment with trials conditionally independent given θ and the action sequence, such as active learning. Aleatoric biases remain, and r^{sur}'s decomposition is unchanged (Theorem 2). But, any abductive bias of the form seen in Theorem 3 vanishes structurally, simplifying r^{sum}:

Corollary 3.1 (Decomposition of r^{sum} in BED)**.** *Under (i)–(v),*

$$\mathbb{E}[r_t^{\text{sum}} \mid h_t, a_{t:T-1}] = I(s_t; \theta \mid h_t) - I(s_t; \theta \mid h_t, a_{t:T-1}, s_{t+1:T}),$$

i.e., the abductive of Theorem 3 vanishes structurally. As $T \to \infty$, $\mathbb{E}[r_t^{\text{sum}}] \to I(s_t; \theta \mid h_t)$.

The proof is in Appendix C.9. Thus r^{sum} is an asymptotically unbiased BIG estimator in BED. For r^{dl} in the BED setting, things also improve. The posterior predictive term admits a ρ-predictive form via a counterfactual action commitment that "copies" the current transition (s_{t-1}, a_{t-1}, s_t) into ρ's context:

$$\int p(s_t \mid s_{t-1}, a_{t-1}, \theta) p(\theta \mid h_t, s_t) d\theta = \rho\big(s_{t+1} = s_t \big| h_t, s_t, a_t = a_{t-1}\big). \tag{6}$$

The counterfactual is the action choice $a_t = a_{t-1}$ (re-using the same action), valid because actions are policy-generated and the BED environment's transition kernel has no s_{t-1} dependence. The identity in Equation (6) makes r^{dl} implementable using in-context learners at the cost of a hypothetical extra rollout step.

Theorem 4 (Decomposition of r^{dl} in BED)**.** *Under (i)–(v), with $L(\theta) := p(s_t \mid a_{t-1}, \theta) / \rho(s_t \mid h_t)$,*

$$\mathbb{E}[r_t^{\text{dl}} \mid h_t] = I(s_t; \theta \mid h_t) + \mathbb{E}_{s_t \mid a_{t-1}} \Big[\log \mathbb{E}_{p(\theta \mid h_t, s_t)}[L(\theta)] - \mathbb{E}_{p(\theta \mid h_t, s_t)}[\log L(\theta)] \Big],$$

i.e., BIG plus a non-negative Jensen gap. Both terms in the expectation vanish as $t \to \infty$ under (iv).

The proof is in Appendix C.10. Like r^{sum}, r^{dl} converges to BIG in BED, but with opposite-sign bias: r^{sum} is biased downward by a residual on future data, r^{dl} upward by a Jensen gap on past data. We thus get $\mathbb{E}[r_t^{\text{sum}}] \le I(s_t; \theta \mid h_t) \le \mathbb{E}[r_t^{\text{dl}}]$, with the gap closing asymptotically. Importantly, however, it should be noted that r^{dl}'s bias (the Jensen gap) vanishes together with BIG $I(s_t; \theta \mid h_t)$ itself, as $t \to \infty$ (see Appendix C.9).

Summary. BED removes the obstructions of general BAMDPs: r^{sum} and r^{dl} both asymptotically recover BIG. However, r^{sum} recovers BIG as $T \to \infty$ while r^{dl} does so as $t \to \infty$. This means that given long sequences, r^{sum} can, in principle, recover BIG at finite t while r^{dl} only trivially recovers BIG in the limit where the signal itself vanishes. Nevertheless, both rewards can still be useful at finite T and finite t, as we now investigate in experiments.

5 Experiments

We evaluate our framework on three structured BED environments: Gaussian Process function estimation, Mastermind code-breaking, and Alchemy transition-rule discovery, each requiring the policy to actively gather observations that are informative about unknown latent variables.

We train policies with PPO or REINFORCE using curiosity-driven rewards from Section 4.1: r^{sur}, r^{dl}, and r^{sum}. We also compare to r^{task}, which uses a validation metric of the environment (indicative of having learned its dynamics) as the reward. For each environment, we pretrain an in-context learner ρ by sampling environment latents from the prior $\theta \sim p(\theta)$ and unrolling trajectories with uniformly-random actions; as such, we call these Prior-Fitted Networks (PFNs) in line with prior work (Müller et al., 2022). Some environments additionally admit an exact predictive Bayesian oracle predictive ρ^*. Details and results for each environment are given below.

5.1 Gaussian Process

The Gaussian Process (GP) environment models active function estimation in a continuous domain. An unknown function f is sampled from a GP prior with a rational quadratic kernel. The function f is represented by a finite set of inducing points sampled on a regular grid of resolution R over the domain $[-x_{\max}, x_{\max}]^2$; the posterior mean at any query location is obtained by exact GP conditioning on these inducing points. More details can be found in Appendix E.1.1.



Figure 2: Comparison of validation log-likelihood for Gaussian Processes for the Oracle and PFN predictive models. (*Left*) For r^{sur} validation score decreases over training steps. (*Center and right*) Final validation scores: r^{sum} and r^{dl} achieve equivalent performance to training on r^{task}, but r^{sur} performs significantly worse than a random policy.

A fresh function f is drawn at the start of each episode. At each step the policy selects a query location $x_t \in [-x_{\max}, x_{\max}]^2$ and observes a noisy evaluation $y_t = f(x_t) + \varepsilon_t$, where $\varepsilon_t \sim \mathcal{N}\big(0, \sigma^2_{\text{noise}}(x_t)\big)$ is spatially varying noise. The noise is in the form of a tiled checkerboard pattern (details in Appendix E.1.1), and is the same in every episode. An optimal information-gathering policy should therefore learn to preferentially query the low-noise tiles. Consecutive observations are conditionally independent given f and the noise map. Therefore, the environment follows the BED setting with $\theta = f$, $a_t = x_t$, and $s_{t+1} = y_t$.

Predictive model. The Bayesian oracle ρ^* performs exact GP posterior inference. After observing h_t, the predictive distribution for y_t is $\rho^*(y_t \mid h_t, x_t) = \mathcal{N}\big(y_t; \mu_t(x_t), \sigma_t^2(x_t)\big)$ where μ_t and σ_t^2 are the standard GP posterior mean and predictive variance (including observation noise) conditioned on h_t. Note ρ^* has access to the true kernel and the noise map. The learned PFN ρ is a causal Transformer. Details about the training and architecture are in Appendix E.1.3. We also show in Figure E.1 that ρ approximates ρ^* well.

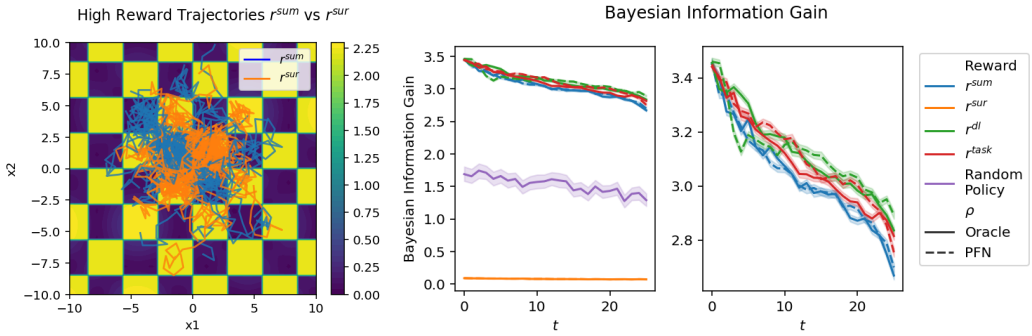

Figure 3: *Left:* The colourbar depicts noise strength of the GP. Top 10% of random paths ranked by the rewards show that r^{sum} is high for the noiseless tiles, conversely r^{sur} favours the noisy regions. *Center:* BIG across trajectories generated by trained policies. r^{sur} has the lowest information gain due to the Noisy TV problem. *Right:* Zoomed comparison. Higher BIG for r^{sum}, r^{dl}, and r^{task} indicates that these policies select queries that are more informative about the underlying function.

Validation. To measure the quality of the collected trajectory h_T, we evaluate on a set of validation points on the grid. Let $(X_{val}, Y_{val}) = \{(\tilde{x}_j, \tilde{y}_j)\}_{j=1}^{M}$ be a random subset of the episode's ground-truth GP inducing points. Regardless of whether ρ is a Bayesian oracle or a trained PFN, we validate by conditioning the oracle ρ^* on the trajectory h_T and report the mean log-likelihood $V = \frac{1}{M}\sum_{j=1}^{M} \log \rho^*(\tilde{y}_j \mid h_T, \tilde{x}_j)$.

Results. Since the environment contains aleatoric noise, we observe a stark divergence in performance between the rewards. As shown in Figure 2, both the random baseline and the policy trained with r^{sur} perform significantly worse than those trained with r^{dl} and r^{sum}. We note that the validation log-likelihood steadily decreases over training, despite r^{sur} increasing (Figure E.2). This provides strong empirical confirmation of Theorem 2, which states that r^{sur} is susceptible to the "noisy TV" problem, whereas r^{sum} and r^{dl} successfully distinguish between reducible epistemic uncertainty and irreducible aleatoric noise.

To further confirm this, in Figure 3, we collect 1000 trajectories with a random policy. We then calculate r^{sur} and r^{sum} for these trajectories, and in green highlight the ones in the 90% percentile for each reward. We can observe that the trajectories with the highest r^{sum} stray away from the noisy checkerboards, whereas those for r^{sur} concentrate in these regions instead. We also plot the BIG across trajectories collected by the policies trained with the different rewards. As predicted, the BIG of r^{sur} is much lower than that of other rewards. Finally, in Appendix E.2, we show empirically that r^{sum} and r^{dl} decompose into BIG and a shrinking finite-time nuisance term, as predicted by Corollary 3.1 and Theorem 4.

5.2 Mastermind

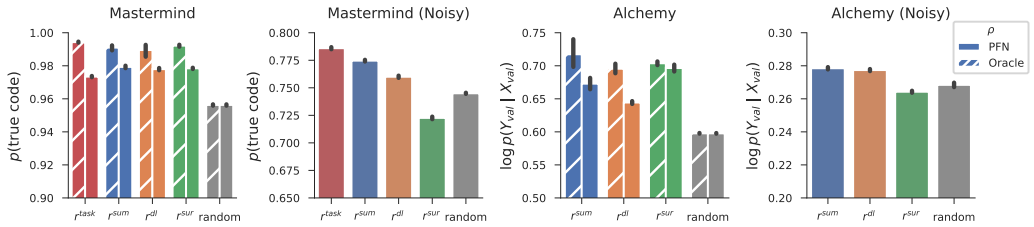

Figure 4: Validation scores for Mastermind and Alchemy. In the standard variant, all rewards are competitive. In both the noisy variants, r^{sur}'s performance falls below the random baseline, whereas r^{sum} and r^{dl} remain robust.

Mastermind is a code-breaking game parameterized by a secret code $c \in \{0,\ldots,C-1\}^L$ made of coloured pegs, where L is the code length and C the number of colours. At each step, the policy submits a guess g_t and receives feedback (b_t, w_t), where b_t counts the number of pegs correct in both colour and position, and w_t counts the total number of pegs correct in colour irrespective of position. The feedback depends only on the current guess and the fixed code. This follows the BED setting with $\theta = c$, $a_t = g_t$, and $s_{t+1} = (b_t, w_t)$.

We additionally consider a noisy variant of the environment in which one randomly chosen colour per episode is designated as "evil." Any guess position whose colour matches the evil colour has that position's value replaced with a uniformly random colour before feedback is computed, so the resulting (b_t, w_t) may not reflect the true code. Additionally, the evil colour is guaranteed not to appear in the secret code itself. The evil colour identity is unknown to the policy and the learned world model ρ.

Predictive model. The oracle ρ^* maintains an exact posterior over secret codes by enumerating all C^L possibilities and filtering to those consistent with the observed history. It computes the predictive probability of each new response as the fraction of consistent codes that would produce that response:

$$\rho^*(s_t \mid h_t) = \frac{|\{c' : c' \text{ consistent with } h_t \text{ and } (a_{t-1}, s_t)\}|}{|\{c' : c' \text{ consistent with } h_t\}|}.$$

ρ^* is only tractable when the environment is deterministic. The learned PFN ρ is a causal Transformer pretrained on randomly generated trajectories. More details about consistency, the PFN architecture, and training process are in Appendices E.3.1 and E.3.2.

Validation. After the policy collects a trajectory of T guesses and responses h_T, we measure how much probability mass the world model places on the true secret code. We construct a synthetic action/observation pair $\tilde{a}_v = c$, $\tilde{s}_v = (L, L)$ — the true code paired with a perfect-score response — append it to the trajectory, and report $V = \rho^*(\tilde{s}_v \mid h_T, \tilde{a}_v)$. For the noisy variant, we use ρ instead of ρ^*.

Results. As seen in Figure 4, r^{sur} achieves near-optimal performance in the standard setup given that the environment has no source of aleatoric noise, matching the performance of training directly for the task. r^{sum} and r^{dl} also achieve comparable performance. In this setting, all reward functions perform consistently across both the Bayesian oracle and the trained PFN, further validating that sequence models can serve as effective predictive models for intrinsic reward computation.

However, the introduction of noisy pegs significantly alters these dynamics. In the noisy variant, the performance of the policy trained with r^{sur} falls even below the random baseline. Conversely, policies trained with r^{sum} and r^{dl} remain robust, retaining high performance despite the corruption mechanism. This confirms that r^{sur} becomes a liability in environments with stochastic transitions, while r^{sum} and r^{dl} consistently identify informative actions.

5.3 Alchemy

Alchemy (Wang et al., 2021) is a symbolic meta-learning environment. Each environment defines a fixed set of transition rules mapping (stone, potion) input pairs to transformed output stones, and the agent must discover these rules through sequential experimentation.

The environment is parameterized by a discrete ID θ, which determines the transition dynamics. Stones are four-dimensional discrete vectors whose dimensions have $(3,3,3,4)$ possible values respectively, and potions take one of 6 values. At each step the policy selects both the initial stone and the potion (i_t, p_t) and observes final stone f_t from the environment. Consecutive observations are conditionally independent given θ (i.e. i_{t+1} is chosen freely and need not equal f_t). Therefore, Alchemy is BED with $a_t = (i_t, p_t)$, and $s_{t+1} = f_t$.

Similar to Mastermind, we consider a noisy variant of the environment in which one randomly chosen stone type per episode is designated as "evil." Any transition whose input stone matches the evil stone has its output replaced with a uniformly random stone.

Predictive model. Like Mastermind, the oracle ρ^* performs exact Bayesian inference by enumerating all N candidates, and eliminating those that are not consistent (all observed transitions match θ's rules). ρ^* is only tractable when the environment is deterministic. The learned PFN ρ is a causal Transformer with an autoregressive GRU decoder that factorizes output prediction across the components of f_t (see Appendix E.4.2 for full architectural and training details).

Validation. After the policy collects a trajectory of T transitions h_T, we evaluate how well the world model can predict held-out transitions from the same environment. From the ground-truth transition table of the current environment ID, we randomly sample $M{=}10$ transitions, excluding any whose initial stone matches the evil stone (so that only clean transitions are used). These are appended sequentially to the trajectory, and we report the mean predictive log-probability $V = \frac{1}{M}\sum_{j=1}^{M} \log \rho(\tilde{s}_j \mid h_{1:T}, \tilde{a}_{1:j-1}, \tilde{s}_{1:j-1})$. For the non-noisy environment we evaluate under both the Bayesian oracle ρ^* and the trained PFN ρ; for the noisy (evil stone) variant we evaluate under ρ only.

Results. We observe a similar pattern as Mastermind in the Alchemy environment in Figure 4. In the noiseless task, r^{sur} and r^{sum} both exhibit strong performance, followed by r^{dl}. Under the noisy variant, r^{sur} again demonstrates a failure to generalize, with performance below that of a random policy. In contrast, r^{sum} maintains its superiority with both oracle and PFN, achieving the highest validation log-probability.

The results across all three domains underscore the limitation of surprisal-based curiosity in realistic noisy settings, and demonstrate the robustness of r^{sum} and r^{dl}. They also demonstrate the effectiveness of trained in-context learners as predictive models in these settings, as replacements for the intractable BIG.

6 Discussion and Future Work

We demonstrated that in-context learners can efficiently evaluate prediction-based intrinsic curiosity rewards, under certain conditions. In Bayesian Experimental Design (BED) settings, our proposed reward r^{sum} and description length reward from prior work r^{dl} asymptotically approximate Bayesian information gain (BIG) and drive exploration effectively, though r^{dl} suffers from asymptotic pathologies that r^{sum} avoids. Both avoid the pitfalls of standard surprisal. This framework opens several promising directions for future work.

General BAMDPs. Extending this approach to general Bayes-Adaptive Markov Decision Processes (BAMDPs) requires handling environments with temporal structure. Implementing r^{dl} or BIG requires access to a model's parameters, which is challenging using in-context learners that only expose the model's predictions. A promising approach is to project the in-context learner's hidden states into "task vectors" (Hendel et al., 2023) that represent the environment's latent dynamics. Modulating or manipulating these task vectors counterfactually could allow the direct estimation of r^{dl} or BIG in latent space. Attempting to approximate r^{gap} — which asymptotically converges to BIG — using long context lengths and gaps is also a possibility.

Training the in-context learner. Matching the true Bayesian posterior predictive over the distribution of policy-sampled trajectories during training poses significant challenges. Large foundation models trained on broad data mixtures offer powerful in-context learning, but they are often poorly calibrated to the true Bayesian posterior predictive (Badola et al., 2025). Furthermore, standard causal Transformers suffer from length generalization issues and positional encoding leakage, breaking the martingale properties necessary for Bayesian coherence (Falck et al., 2024). While this is partially mitigated in BED by using permutation-invariant sequence models trained on synthetic priors, covariate shift remains a primary

failure mode. Specifically, a model trained on trajectories from a random or fixed policy will degrade when evaluated on informative trajectories collected by an active, reward-maximizing deployment policy (Yadlowsky et al., 2023). Addressing this distribution shift may require techniques that jointly fine-tune the predictive model and the policy (Ross et al., 2011; Ivanova et al., 2024; Azar et al., 2019), though optimizing against non-stationary intrinsic rewards from a changing ρ might introduce instability.

Online RL setting. This work focuses on an episodic meta-RL setting, but translating these methods to single-lifetime online reinforcement learning introduces several difficulties. First, the fixed context window of sequence models limits the duration over which an agent can accumulate learning progress. Second, in-context learning empirically exhibits saturation; predictive improvements decrease as the context length grows (Elmoznino et al., 2025), perhaps requiring periodic weight updates to consolidate long-term knowledge (Bornschein et al., 2024). Third, the rewards themselves become computationally intractable in continuous online settings. Currently, the intrinsic reward computation is naturally episodic and calculated only at the conclusion of a trajectory (where r^{sum} evaluates over the remainder of the fully unrolled sequence, and r^{dl} requires trajectory reruns with the new parameters). Approximating these episodic metrics for online scaling will require stochastic sampling or truncated evaluation windows. Finally, exploration in domains with low-level action spaces is inherently difficult. Leveraging emergent temporal and action abstractions within autoregressive models could allow intrinsic rewards to operate over higher-level behaviors, improving the efficiency of information gathering (Kobayashi et al., 2025).

Comparing r^{sum} and r^{dl}. Empirically, we found that r^{sum} and r^{dl} both performed well, but that the optimal one between the two depended on the environment in which they were deployed, suggesting that they can be complementary. Theoretical analysis in BED settings established that r^{sum} bounds BIG from below, whereas r^{dl} bounds it from above. The bounds for each are fundamentally different, r^{dl} converging as $t \to \infty$, and r^{sum} as $T \to \infty$. r^{dl} technically recovers BIG but its bias vanishes together with the signal itself, and r^{sum} should be more robust and approach BIG at finite t. However, our experiments show that this distinction may not manifest in simple environments, and at finite T and t. Future work might be able to characterize the properties of environments that influence these factors. In addition, because the two bounds converge asymptotically from opposite sides, future work could explore hybridizing them to yield a tighter, more stable estimator of true learning progress.

Acknowledgments and Disclosure of Funding

The authors acknowledge Seijin Kobayashi, Alexander Meulemans, James Evans, and Hugo Larochelle for helpful feedback and discussions. EE and SB are PhD students at the Université de Montréal & Mila – Quebec AI Institute. GL is an Associate Professor at the Université de Montréal and a Core Academic Member at Mila – Quebec AI Institute. EE acknowledges support from Vanier Canada Graduate Scholarship #492702.

References

Aubret, A., Matignon, L., and Hassas, S. (2023). An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey. *Entropy*, 25(2):327.

Azar, M. G., Piot, B., Pires, B. A., Grill, J.-B., Altché, F., and Munos, R. (2019). World discovery models. *arXiv preprint arXiv:1902.07685*.

Badola, K., Simon, J., Hosseini, A., Carthy, S. M. M., Munkhdalai, T., Goyal, A., Kočiskỳ, T., Upadhyay, S., Fatemi, B., and Kazemi, M. (2025). Multi-turn puzzles: Evaluating interactive reasoning and strategic dialogue in llms. *arXiv preprint arXiv:2508.10142*.

Barto, A., Mirolli, M., and Baldassarre, G. (2013). Novelty or surprise? *Frontiers in Psychology*, 4.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. (2016). Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29.

Binz, M., Dasgupta, I., Jagadish, A. K., Botvinick, M., Wang, J. X., and Schulz, E. (2024). Meta-learned models of cognition. *Behavioral and Brain Sciences*, 47:e147.

Bornschein, J., Li, Y., and Rannen-Triki, A. (2024). Transformers for supervised online continual learning. *arXiv preprint arXiv:2403.01554*.

Bradley, H., Dai, A., Teufel, H. B., Zhang, J., Oostermeijer, K., Bellagente, M., Clune, J., Stanley, K., Schott, G., and Lehman, J. (2024). Quality-diversity through AI feedback. In *The Twelfth International Conference on Learning Representations*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., and Efros, A. A. (2019). Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*.

Chentanez, N., Barto, A., and Singh, S. (2004). Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 17.

De, S., Smith, S. L., Fernando, A., Botev, A., Cristian-Muraru, G., Gu, A., Haroun, R., Berrada, L., Chen, Y., Srinivasan, S., Desjardins, G., Doucet, A., Budden, D., Teh, Y. W., Pascanu, R., Freitas, N. D., and Gulcehre, C. (2024). Griffin: Mixing gated linear recurrences with local attention for efficient language models.

Ding, L., Zhang, J., Clune, J., Spector, L., and Lehman, J. (2023). Quality diversity through human feedback. In *Second Agent Learning in Open-Endedness Workshop*.

Du, Y., Kosoy, E., Dayan, A. L., Rufova, M., Gopnik, A., and Abbeel, P. (2023). What can AI learn from human exploration? intrinsically-motivated humans and agents in open-world exploration. In *Second Agent Learning in Open-Endedness Workshop*.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. (2016). RI^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Elmoznino, E., Marty, T., Kasetty, T., Gagnon, L., Mittal, S., Fathi, M., Sridhar, D., and Lajoie, G. (2025). In-context learning and occam's razor. In *Forty-second International Conference on Machine Learning*.

Falck, F., Wang, Z., and Holmes, C. (2024). Is in-context learning in large language models bayesian? a martingale perspective. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Fedorov, V. (1972). *Theory of Optimal Experiments*. Academic Press, NY.

Grau-Moya, J., Genewein, T., Hutter, M., Orseau, L., Deletang, G., Catt, E., Ruoss, A., Wenliang, L. K., Mattern, C., Aitchison, M., and Veness, J. (2024). Learning universal predictors. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F., editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 16178–16205. PMLR.

Hazan, E., Kakade, S., Singh, K., and Van Soest, A. (2019). Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691. PMLR.

Hendel, R., Geva, M., and Globerson, A. (2023). In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333.

Hester, T. and Stone, P. (2012). Intrinsically motivated model learning for a developing curious agent. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–6.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. (2022). Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. (2016). Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29.

Hughes, E., Dennis, M. D., Parker-Holder, J., Behbahani, F., Mavalankar, A., Shi, Y., Schaul, T., and Rocktäschel, T. (2024). Position: Open-endedness is essential for artificial superhuman intelligence. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F., editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20597–20616. PMLR.

Itti, L. and Baldi, P. (2009). Bayesian surprise attracts human attention. *Vision research*, 49(10):1295–1306.

Ivanova, D. R., Hedman, M., Guan, C., and Rainforth, T. (2024). Step-dad: Semi-amortized policy-based bayesian experimental design. In *ICLR 2024 Workshop on Data-centric Machine Learning Research (DMLR)*, volume 2, page 21.

Kim, H., Kim, J., Jeong, Y., Levine, S., and Song, H. O. (2019). EMI: Exploration with mutual information. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3360–3369. PMLR.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005a). All else being equal be empowered. In *European Conference on Artificial Life*, pages 744–753. Springer.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005b). Empowerment: A universal agent-centric measure of control. In *2005 ieee congress on evolutionary computation*, volume 1, pages 128–135. IEEE.

Kobayashi, S., Schimpf, Y., Schlegel, M., Steger, A., Wolczyk, M., von Oswald, J., Scherrer, N., Maile, K., Lajoie, G., Richards, B. A., et al. (2025). Emergent temporal abstractions in autoregressive models enable hierarchical reinforcement learning. *arXiv preprint arXiv:2512.20605*.

Levy, G., Colas, C., Oudeyer, P.-Y., Carta, T., and Romac, C. (2025). Worldllm: Improving llms' world modeling using curiosity-driven theory-making. *arXiv preprint arXiv:2506.06725*.

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005.

Little, D. Y. and Sommer, F. T. (2013). Learning and exploration in action-perception loops. *Frontiers in neural circuits*, 7:37.

Lopes, M., Lang, T., Toussaint, M., and Oudeyer, P.-Y. (2012). Exploration in model-based reinforcement learning by empirically estimating learning progress. *Advances in neural information processing systems*, 25.

MacKay, D. J. (1992). Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604.

Mikulik, V., Delétang, G., McGrath, T., Genewein, T., Martic, M., Legg, S., and Ortega, P. (2020). Meta-trained agents implement bayes-optimal agents. *Advances in neural information processing systems*, 33:18691–18703.

Mohamed, S. and Jimenez Rezende, D. (2015). Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28.

Müller, S., Hollmann, N., Arango, S. P., Grabocka, J., and Hutter, F. (2022). Transformers can do bayesian inference. In *International Conference on Learning Representations*.

Nagler, T. (2023). Statistical foundations of prior-data fitted networks. In *International Conference on Machine Learning*, pages 25660–25676. PMLR.

Ostrovski, G., Bellemare, M. G., Oord, A., and Munos, R. (2017). Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR.

Oudeyer, P.-Y. and Kaplan, F. (2007). What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:108.

Oudeyer, P.-Y. and Kaplan, F. (2008). How can we define intrinsic motivation? In *the 8th international conference on epigenetic robotics: Modeling cognitive development in robotic systems*. Lund University Cognitive Studies, Lund: LUCS, Brighton.

Oudeyer, P.-Y., Kaplan, F., and Hafner, V. V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ross, S., Gordon, G., and Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.

Salge, C., Glackin, C., and Polani, D. (2014). *Empowerment–An Introduction*, pages 67–114. Springer Berlin Heidelberg, Berlin, Heidelberg.

Schiff, J. L. (1993). *Normal Families*. Universitext. Springer-Verlag, New York.

Schmidhuber, J. (1991a). Curious model-building control systems. In *Proc. international joint conference on neural networks*, pages 1458–1463.

Schmidhuber, J. (1991b). A possibility for implementing curiosity and boredom in model-building neural controllers. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*. The MIT Press.

Schmidhuber, J. (2009). Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. In Pezzulo, G., Butz, M. V., Sigaud, O., and Baldassarre, G., editors, *Anticipatory Behavior in Adaptive Learning Systems*, pages 48–76, Berlin, Heidelberg. Springer Berlin Heidelberg.

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247.

Schmüdgen, K. (2017). *The Moment Problem*, volume 277 of *Graduate Texts in Mathematics*. Springer.

Seitzer, M., Tavakoli, A., Antic, D., and Martius, G. (2022). On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations*.

Stadie, B. C., Levine, S., and Abbeel, P. (2015). Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*.

Stanley, K. O. and Lehman, J. (2015). Why greatness cannot be planned: The myth of the objective. *(No Title)*.

Storck, J., Hochreiter, S., Schmidhuber, J., et al. (1995). Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the international conference on artificial neural networks, Paris*, volume 2, pages 159–164.

Strehl, A. L. and Littman, M. L. (2008). An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331.

Sun, Y., Gomez, F., and Schmidhuber, J. (2011). Planning to be surprised: Optimal bayesian exploration in dynamic environments. In *International conference on artificial general intelligence*, pages 41–51. Springer.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. (2024). Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*.

Wang, J., King, M., Porcel, N., Kurth-Nelson, Z., Zhu, T., Deck, C., Choy, P., Cassin, M., Reynolds, M., Song, F., Buttimore, G., Reichert, D., Rabinowitz, N., Matthey, L., Hassabis, D., Lerchner, A., and Botvinick, M. (2021). Alchemy: A structured task distribution for meta-reinforcement learning. *arXiv preprint arXiv:2102.02926*.

Wang, R., Lehman, J., Clune, J., and Stanley, K. O. (2019). Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*.

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. (2022). An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Yadlowsky, S., Doshi, L., and Tripuraneni, N. (2023). Pretraining data mixtures enable narrow model selection capabilities in transformer models. *arXiv preprint arXiv:2311.00871*.

Zhang, J., Lehman, J., Stanley, K., and Clune, J. (2024). OMNI: Open-endedness via models of human notions of interestingness. In *The Twelfth International Conference on Learning Representations*.

Appendix A Additional Prior Work on Intrinsic Curiosity

Surprisal in a Learned Latent Space. One effective approach for mitigating the problem of aleatoric noise is to measure surprisal in a learned latent space that only encodes *predictable* information about the observation (Pathak et al., 2017; Kim et al., 2019; Burda et al., 2019). However, these methods struggle in partially observed environments where the state representation depends on a long history, and also remain sensitive to sources of aleatoric noise that are produced by the policy's actions (Burda et al., 2019). Ultimately, the problem of how to train a latent space for surprisal-based exploration that is robust to aleatoric noise remains open. Since our investigation into the use of in-context learners for intrinsic curiosity leverages only ρ's input/output interface for computing observation-level prediction errors, we do not consider methods that operate in latent space (although if in-context learners are trained in an appropriate latent space, the approach might be viable).

Novelty. An approach with a long history in reinforcement learning is to seek *novel* states of the environment, as measured by visitation counts or density models (Hazan et al., 2019; Strehl and Littman, 2008; Bellemare et al., 2016; Ostrovski et al., 2017). Novelty is deeply related to surprisal; while the two are not quite equivalent (Barto et al., 2013), they both suffer from the same fundamental problem: noisy regions of the environment with high observation entropy are a reliable source of both surprisal and novelty.

Empowerment. Empowerment is an alternative framework for intrinsic motivation that, instead of gathering observations that improve a model of the environment, aims to explore regions of the environment that are *controllable* by the policy (Klyubin et al., 2005a,b; Salge et al., 2014). Specifically, empowerment is defined as the channel capacity of the policy's actuation channel: $I(S';A \mid S)$ where I is mutual information, S is a random variable for the history of all past states, A is a random variable for the agent's current action, and S' is a random variable for the state(s) the policy observes after some future horizon following the action (S' can represent a single future state at a fixed horizon, or potentially a trajectory of such states). The distributions over A, S, and S' depend on both the environment and the policy, and using empowerment as an intrinsic reward involves maximizing $I(S';A \mid S)$ with respect to the latter. Intuitively, this yields policies that learn to exert control over their environment by exploring regions of the state space where actions provide substantial information about future environment dynamics.

It is unclear how empowerment theoretically relates to the above methods that seek to improve a model of the environment. While exploring controllable regions of an environment might improve a model as a side effect (e.g., knowing what is and is not controllable involves understanding the environment's dynamics at some level), the two are not equivalent in general. While empowerment appears to be a viable exploration strategy empirically (Mohamed and Jimenez Rezende, 2015) and can be highly effective in modeling human exploration behaviour (Du et al., 2023), we instead focus here on intrinsic rewards that explicitly aim to improve a model of the environment.

Open-endedness. The aims of intrinsic curiosity closely align with those of research on "open-endedness" (Stanley and Lehman, 2015), typically characterized as an ongoing process that continually generates new data, tasks, or behaviours that are interesting, learnable, novel, without being *too* difficult relative to an agent's current capabilities. Most existing approaches operationalize open-endedness either through diversity/complexity heuristics (Wang et al., 2019, 2024) or through proxies intended to capture subjective human notions of interestingness (Ding et al., 2023; Bradley et al., 2024; Zhang et al., 2024).

A recent position paper attempts to mathematically formalize the concept of open-endedness (Hughes et al., 2024), defining it as a process that continually generates data that is (a) surprising and (b) learnable, where both are defined in terms of the prediction errors of a world model that has observed the data up to a particular timepoint. Notably, their learnability component closely resembles learning progress rewards. We further hypothesize that their explicit novelty term is redundant: if learning progress is sustained and does not stagnate, then novelty and surprisal are already implied, because continued progress requires exposure to information not yet captured by the model.

Appendix B Assumptions for Theoretical Results

The main text states six assumptions in compact form. We unpack each here, flagging the nuances that matter for the derivations.

(i) Markovian, stationary dynamics given θ. For each fixed θ, the transition kernel $p(s_{t+1} \mid s_t, a_t, \theta)$ is Markov in the state-action chain and time-independent. Stationarity rules out time-indexed families $\{P_t\}_t$ in which the conditional law itself depends on chain time. Markov dynamics given θ are what let us factorize ρ's joint over future observations (Lemma 4.1); stationarity ensures that masking-based rewards (r^{sum} and the NDIGO reward of Azar et al. (2019)) can be evaluated by anchoring their kernels at the actual chain times.

(ii) ρ is exactly Bayesian over θ. The sequence model emits the exact posterior-predictive $\rho(s_{t'} \mid h_{t'}) = \int p(s_{t'} \mid s_{t'-1}, a_{t'-1}, \theta) p(\theta \mid h_{t'}) d\theta$, with prior $p(\theta)$ matching the distribution of training tasks. In practice ρ is a finite-capacity neural network and exact Bayesian computation is an idealization; Appendix B.2 discusses how approximation error propagates into reward biases.

(iii) Observation-only masking. The masked history $h_{t'\setminus t}$ replaces s_t with a mask token while leaving every other state and every action intact. We assume ρ's response to $h_{t'\setminus t}$ is the posterior-predictive obtained by Bayesian marginalization over the masked s_t, treating actions as still observed. This is a subtle assumption when ρ is a learned sequence model: masked queries are out-of-distribution unless ρ has been trained to handle them. See Appendix B.3.

(iv) Posterior consistency. The selection rule generates trajectories under which $p(\theta \mid h_t) \to \delta_{\theta^*}$ in probability as $t \to \infty$. This requires (a) θ identifiability from the achievable observation distributions under the policy and (b) sufficient coverage of the action space (each action taken often enough to inform θ). Under standard regularity conditions (smooth parametric model, positive-definite per-arm Fisher information $\mathcal{I}(\theta) \succ 0$ uniformly on the parameter support), Bernstein–von-Mises gives the rate $\mathrm{Var}\big[p(\theta \mid h_t)\big] \sim \mathcal{I}(\theta^*)^{-1}/t$, controlling both the residual in Theorem 3 and the Jensen gap in Theorem 4 at the $1/t$ scale. Strictly, the consistency assumption alone is what the framework requires; the Fisher-information regularity becomes relevant only when one wants asymptotic rates rather than asymptotic limits.

(v) Policy-generated actions. Actions are sampled from π_ϕ conditional on past observations. Because actions are not generated by the environment, conditioning on a future action does not update the posterior over θ: $p(\theta \mid h_t, a_t, a_{t+1}, \ldots, a_{t'-1}) = p(\theta \mid h_t)$. This permits treating future actions as fixed inputs in the derivations, and is equivalent to the standard interventional treatment of actions in causal Markov chains.

(vi) Mixing given θ. For each θ in the support of $p(\theta)$, $I(s_t; s_{t+k} \mid \theta, h_t, a_{t:t+k-1}) \to 0$ as $k \to \infty$. This rules out absorbing states and other non-ergodic dynamics under which the Markov chain never forgets s_t. Mixing is what makes the abductive bias of Theorem 3 (and its r^{gap}-generalized analogues) decay with the gap parameter K; without it, infinite-gap rewards still retain a bias from kernel-mediated coupling between s_t and the deep future.

B.1 The Prior $p(\theta)$ and Meta-Learning

The prior $p(\theta)$ throughout this paper is the training distribution $p_{\text{train}}(\theta)$ over MDPs — ρ is pretrained on trajectories from MDPs sampled from p_{train}, and its posterior-predictive is assumed to be Bayesian *under* p_{train}. This is a meta-learning setup, and the framework's predictions extend to deployment insofar as the deployed environment lies in $\mathrm{supp}(p_{\text{train}})$. We treat deployment mismatch on θ as a separate source of bias and adopt the train-prior framing throughout.

A more pointed meta-learning concern is that ρ's training data — pairs of trajectories and their corresponding posterior-predictives — depends not only on $p_{\text{train}}(\theta)$ but on the behavior of the policy that generates the training trajectories. Pretraining ρ offline, separately from the deployment policy π_ϕ, therefore requires the pretraining policy to cover the trajectory distribution adequately. This is mostly tractable in BED settings, where trajectories factorize across actions given θ. Ensuring enough coverage of action selection during training should suffice. It becomes increasingly difficult as the domain expands, and especially in temporally rich BAMDPs where the trajectory distribution depends nonlinearly on the policy: a deployed reward-maximizing policy can easily probe regions that the pretraining policy under-sampled, leaving ρ uncalibrated where it matters most. In principle, this can be alleviated with large enough training samples under enough random policies. Nevertheless, this is a practical matter to be considered.

This is a potential weakness of any offline-pretraining plus online-policy-training scheme: the predictive model is fixed during policy optimization, but policy iteration may drift outside ρ's effective training

support. On the other hand, the scheme opens the door to the use of massive pretrained but frozen foundation models (maybe even LLMs) to support active ICL in a variety of environments. Practical investigations of policy training in this regime, including the robustness of the method to departure from Bayesian exactness of ρ (see next section), and the question of action selection bias, is exciting future work.

Another avenue for the approach to have practical legs at scale would be that of joint training of π and ρ. This could take the form of a meta-learning regime in which the predictor and the data-collection policy improve together so that ρ's posterior-predictive remains accurate over the trajectories the policy actually generates. We visit this question in Section 6.

B.2 Inexactitude of ρ's Bayesian Posterior Predictive

Assumption (ii) demands that ρ be exactly Bayesian. In practice ρ is a finite-capacity neural network trained by maximum likelihood on a finite dataset, so it differs from the exact Bayesian posterior-predictive ρ^* in a controllable but non-zero way. This subsection discusses the consequences of inexactitude as a property of ρ as a learner — independent of any choice about how queries are conditioned on the trajectory. Issues specifically arising from masked queries during training or inference are deferred to Appendix B.3.

Writing $\Delta(Y) \coloneqq D_{\mathrm{KL}}(\rho^*(\cdot \,|\, Y) \| \rho(\cdot \,|\, Y))$ for the per-context KL between exact and approximate predictives at a conditioning Y, each reward inherits a bias that is a signed combination of such KLs. For example,

$$\mathbb{E}_{\rho^*}[r_t^{\mathrm{sur}} \,|\, h_t] = I(s_t;\theta \,|\, h_t) + \mathrm{H}(s_t \,|\, h_t,\theta) + \Delta(h_t),$$

showing that surprisal acquires an additional non-negative $\Delta(h_t)$ on top of the noisy-TV bias. More generally, rewards in $\mathcal{F}$ formed from log-ratios of predictives at two conditionings Y_+, Y_- pick up a differential $\bar{\Delta}(Y_+) - \bar{\Delta}(Y_-)$ rather than an absolute KL, since the two predictive errors partially cancel under log-subtraction.

Two structural features:

- r^{sur}**'s bias is absolute.** A single non-negative KL $\Delta(h_t)$. Contexts where ρ predicts poorly are systematically over-rewarded, on top of the aleatoric noisy-TV — a model-error-seeking pathology specific to surprisal.
- **Log-ratio rewards have differential bias.** Rewards of the form $\log \rho(\cdot \mid Y_+) - \log \rho(\cdot \mid Y_-)$ inherit only the difference of ρ's errors at Y_+ and Y_-. Uniform approximation error cancels; only the asymmetry between the two conditionings survives.

The differential structure of log-ratio rewards is a robustness property: an across-the-board imperfection in ρ cancels in the log-ratio, leaving only the part of the error that distinguishes the two conditionings. Whether and how this differential is small in practice is a separate question — it depends on how ρ is trained and queried. Figure E.1 in Appendix E.1.3 shows the difference of log-predictive terms between an oracle observer and a trained PFN ρ for the Gaussian process experiment. Figure 2 in the main text shows very little difference in reward training outcomes for the same experiment.

B.3 Training ρ With vs. Without Masked Inputs

Computing r^{sum} (and analogously the NDIGO reward of Azar et al. (2019)) requires evaluating ρ on histories with one or more past states masked out. A sequence model trained with standard autoregressive next-token prediction will never have seen a mask token in its training data, so its behavior on masked queries is out-of-distribution. Two training strategies are natural:

Standard autoregressive training. ρ is trained on natural (unmasked) trajectories. At inference, predictives at masked conditioning sets reflect whatever extrapolation the model performs, with no guarantee of agreement with the Bayesian marginalization over the masked observation. This is the source of the differential bias described in Appendix B.2.

Masked training. ρ is trained with random masking of past observations during training, in the spirit of masked language modeling. ρ then learns predictives that are well-calibrated with respect to the marginalized joint, making mask-baseline rewards potentially more reliable.

We remark that these considerations impact reward evaluation in the full BAMDP setting where parts of transition tuples (s_{t-1}, a_{t-1}, s_t) need to be masked. However in the BED setting, masking simplifies

to simply removing observations from pairs (a_{t-1},s_t) and is not subject to the subtleties described above. Our experiments take place in the BED setting, but a more general implementation of ICL intrinsic rewards should consider the points above.

Appendix C Proofs for Theoretical Results

C.1 Interventional Factorization

Several of our derivations rely on factorizing ρ's joint over future observations into a product of one-step posterior-predictives. The factorization hinges on two structural features: (a) Markov dynamics given θ (assumption (i)), and (b) policy-generated actions that do not update the θ-posterior (assumption (v)). The second is what makes actions *interventional* in the causal sense: they enter the joint as fixed inputs, not as random variables to be modeled.

Lemma 4.1 (Interventional factorization)**.** *For $t_1 < t_2$ and any action sequence $a_{t_1:t_2-1}$,*

$$\rho(s_{t_1:t_2} \,|\, h_{t_1}, a_{t_1:t_2-1}) = \prod_{\tau=t_1}^{t_2} \rho(s_\tau \,|\, h_{t_1}, s_{t_1:\tau-1}, a_{t_1:\tau-1}),$$

and the same identity holds with masked histories provided masked states lie outside $[t_1,t_2]$.

Proof. Under (i), the joint factors as $\prod_\tau p(s_\tau \mid s_{\tau-1}, a_{\tau-1}, \theta)$. By assumption (v), conditioning on future actions does not update the posterior over θ. Marginalizing θ on both sides gives the product of posterior-predictives. Assumption (iii) extends to masked histories. □

This factorization is what allows r^{sum}'s telescoping representation $r_t^{\text{sum}} = \text{pmi}(s_t; s_{t+1:T} \,|\, h_t, a_{t:T-1})$ to be used in the proof of Theorem 3.

C.2 Proof of Theorem 1

Theorem 1 (Impossibility of BIG identification in $\mathcal{F}$)**.** *Let $\mathcal{M}$ be the class of BAMDPs satisfying (i)–(v) above, whose transition kernels are neither deterministic nor independent across time given θ. Then for every $r_t \in \mathcal{F}$, there exists $M \in \mathcal{M}$ with $\mathbb{E}^M[r_t \,|\, h_t] \neq I(s_t;\theta \,|\, h_t)$.*

Remark. The argument is a marginal-indistinguishability counterexample: we construct two BAMDPs in $\mathcal{M}$ on which any $r_t \in \mathcal{F}$ has the same unconditional expectation but for which the expected BIG differs. Since an unbiased per-history estimator of BIG forces matched unconditional expectations, the contradiction settles the theorem.

Proof.

Step 1: Unconditional reduction. Suppose $r_t \in \mathcal{F}$ satisfies the per-history identity $\mathbb{E}^M[r_t \,|\, h_t] = I^M(s_t;\theta \,|\, h_t)$ for every history h_t and every $M \in \mathcal{M}$. Taking unconditional expectation,

$$\mathbb{E}^M[r_t] = \mathbb{E}^M\big[I^M(s_t;\theta \,|\, h_t)\big].$$

We will exhibit $M_1, M_2 \in \mathcal{M}$ with $\mathbb{E}^{M_1}[r_t] = \mathbb{E}^{M_2}[r_t]$ but $\mathbb{E}^{M_1}\big[I(s_t;\theta \mid h_t)\big] \neq \mathbb{E}^{M_2}\big[I(s_t;\theta \mid h_t)\big]$, contradicting the per-history identity for at least one of them. We refer to such a pair (M_1,M_2) — or, since the kernel is fixed in our construction, the corresponding prior pair (p_1,p_2) — as a *witness* to the theorem.

Step 2: $\mathbb{E}[r_t]$ depends only on the marginal trajectory distribution. Each $r_t = f\big((\rho(X_i \,|\, Y_i))_{i\in I}\big) \in \mathcal{F}$ has finite *touch horizon*

$$Q := \max\big\{\tau : \tau \text{ is a chain index appearing in some } X_i \text{ or } Y_i\big\},$$

i.e., the largest chain index referenced by the reward. Under Assumption (ii), $\rho(X \,|\, Y) = P^M_{\text{marg}}(X \,|\, Y)$ where P^M_{marg} is the marginal trajectory law of M (with θ integrated out). Taking unconditional expectation,

$$\mathbb{E}^M[r_t] = \mathbb{E}_{P^M_{\text{marg}}}\Big[f\big((P^M_{\text{marg}}(X_i \,|\, Y_i))_{i\in I}\big)\Big],$$

which depends on M only through $P^M_{\text{marg}}(s_{1:Q}, a_{1:Q-1})$. In particular, two BAMDPs with identical marginal trajectory distributions on the first Q steps yield identical $\mathbb{E}[r_t]$.

Step 3: A non-degenerate kernel family with prior-dependent BAMDPs. Take the BAMDP family with state space $\mathcal{S}=\{0,1\}$, a single (dummy) action, initial state $s_1 \sim \mathrm{Bernoulli}(0.5)$, and transition kernel

$$p(s_{\tau+1} \neq s_\tau \,|\, s_\tau, \theta) = \theta, \qquad \theta \in (0, 0.5).$$

The kernel is non-deterministic ($\theta > 0$) and not independent across time given θ ($\theta \neq 0.5$), so each fixed $\theta \in (0,0.5)$ yields a kernel meeting the non-degeneracy condition of Theorem 1. Different priors $p(\theta)$ supported on $(0,0.5)$ specify different BAMDPs in $\mathcal{M}$.

The marginal trajectory distribution of a sequence $s_{1:Q}$ depends only on the number of state flips $k = \#\{\tau : s_{\tau+1} \neq s_\tau\}$:

$$P^p_{\mathrm{marg}}(s_{1:Q}) = \tfrac{1}{2} \int_0^{0.5} \theta^k (1-\theta)^{Q-1-k} p(\theta) d\theta.$$

The integrand is a polynomial in θ of degree at most $Q-1$. Expanding,

$$P^p_{\mathrm{marg}}(s_{1:Q}) = \tfrac{1}{2} \sum_{j=0}^{Q-1} \beta_j(k) m_j(p),$$

where $m_j(p) = \int \theta^j p(\theta) d\theta$ is the j-th moment of p and the $\beta_j(k)$ are combinatorial constants. Hence $P^p_{\mathrm{marg}}(s_{1:Q})$ depends on p only through its first $Q-1$ moments.

By the truncated Hausdorff moment problem (Schmüdgen, 2017), the set of probability measures on $(0,0.5)$ matching any prescribed first $Q-1$ moments is convex with multiple distinct extreme points. Choose two distinct priors p_1, p_2 on $(0,0.5)$ that match in their first $Q-1$ moments. Setting $M_1 = (\text{kernel}, p_1)$ and $M_2 = (\text{kernel}, p_2)$, both lie in $\mathcal{M}$, and by Step 2,

$$\mathbb{E}^{M_1}[r_t] = \mathbb{E}^{M_2}[r_t].$$

Step 4: Expected BIG differs. Decompose

$$\mathbb{E}^M\big[I(s_t;\theta \,|\, h_t)\big] = \mathrm{H}^M_{\mathrm{marg}}(s_t \,|\, h_t) - \mathbb{E}_p\big[\mathrm{H}(s_t \,|\, s_{t-1}, \theta)\big].$$

The marginal entropy $\mathrm{H}^M_{\mathrm{marg}}(s_t \,|\, h_t)$ is identical for M_1 and M_2 (Step 3, marginals match). The aleatoric term is

$$\mathbb{E}_p\big[\mathrm{H}(s_t \,|\, s_{t-1}, \theta)\big] = \mathbb{E}_p\big[\mathrm{H}_{\mathrm{bin}}(\theta)\big],$$

where $\mathrm{H}_{\mathrm{bin}}(\theta) = -\theta\log\theta - (1-\theta)\log(1-\theta)$ is the binary entropy function. Since $\mathrm{H}_{\mathrm{bin}}$ is not a polynomial, its expectation under p is not determined by any finite number of moments of p. By the truncated moment problem, we may choose p_1, p_2 matching in the first $Q-1$ moments but with $\mathbb{E}_{p_1}[\mathrm{H}_{\mathrm{bin}}(\theta)] \neq \mathbb{E}_{p_2}[\mathrm{H}_{\mathrm{bin}}(\theta)]$. Therefore $\mathbb{E}^{M_1}\big[I(s_t;\theta \,|\, h_t)\big] \neq \mathbb{E}^{M_2}\big[I(s_t;\theta \,|\, h_t)\big]$.

Conclusion. We have $\mathbb{E}^{M_1}[r_t] = \mathbb{E}^{M_2}[r_t]$ but $\mathbb{E}^{M_1}\big[I(s_t;\theta \,|\, h_t)\big] \neq \mathbb{E}^{M_2}\big[I(s_t;\theta \,|\, h_t)\big]$. The per-history identity $\mathbb{E}^M[r_t \,|\, h_t] = I^M(s_t;\theta \,|\, h_t)$ implies matched unconditional expectations, so it must fail on at least one of M_1, M_2. Since the construction works for any $r_t \in \mathcal{F}$, the theorem follows. □

C.3 Proof of Theorem 2

Theorem 2 (Decomposition of r^{sur})**.** *Under (i)–(iii),*

$$\mathbb{E}[r^{\mathrm{sur}}_t \,|\, h_t] = I(s_t;\theta \,|\, h_t) + \underbrace{\mathrm{H}(s_t \,|\, h_t, \theta)}_{\text{aleatoric entropy}}.$$

$\mathrm{H}(s_t \,|\, h_t, \theta) = \mathrm{H}(s_t \,|\, s_{t-1}, a_{t-1}, \theta)$ *is non-negative and does not vanish as* θ *becomes identified.*

Proof. $\mathbb{E}[-\log \rho(s_t \mid h_t) \mid h_t] = \mathrm{H}(s_t \mid h_t)$. By the conditioning-information identity, $\mathrm{H}(s_t \mid h_t) = \mathrm{H}(s_t \mid h_t, \theta) + I(s_t; \theta \mid h_t)$. Under (ii), ρ's θ-conditional predictive equals the true kernel, so $\mathrm{H}(s_t \,|\, h_t, \theta) = \mathrm{H}(s_t \,|\, s_{t-1}, a_{t-1}, \theta)$. □

C.4 Proof of Theorem 3

Theorem 3 (Decomposition of r^{sum})**.** *Under (i)–(iii) and (v),*

$$\mathbb{E}[r_t^{\text{sum}} \mid h_t, a_{t:T-1}] = I(s_t;\theta \mid h_t) + \underbrace{I(s_t;s_{t+1} \mid \theta, h_t, a_t)}_{\text{one-step "abductive"}} - \underbrace{I(s_t;\theta \mid h_t, a_{t:T-1}, s_{t+1:T})}_{\text{"residual"}}.$$

Both terms are non-negative. Under (iv), the residual vanishes as $T \to \infty$; the abductive does not.

Proof. By Lemma 4.1, $\sum_{t'=t+1}^{T} \log \rho(s_{t'} \mid h_{t'}) = \log \rho(s_{t+1:T} \mid h_t, s_t, a_{t:T-1})$ and $\sum_{t'=t+1}^{T} \log \rho(s_{t'} \mid h_{t'\setminus t}) = \log \rho(s_{t+1:T} \mid h_t, a_{t:T-1})$. Subtracting, $r_t^{\text{sum}} = \text{pmi}(s_t; s_{t+1:T} \mid h_t, a_{t:T-1})$. Take expectation: $\mathbb{E}[r_t^{\text{sum}}] = I(s_t; s_{t+1:T} \mid h_t, a_{t:T-1})$.

Applying the chain rule against θ:

$$\begin{aligned} I(s_t;s_{t+1:T} \mid h_t, a_{t:T-1}) = {} & I(s_t;\theta \mid h_t, a_{t:T-1}) \\ & - I(s_t;\theta \mid h_t, a_{t:T-1}, s_{t+1:T}) \\ & + I(s_t;s_{t+1:T} \mid \theta, h_t, a_{t:T-1}). \end{aligned}$$

Future actions do not update the θ-posterior, so $I(s_t;\theta \mid h_t, a_{t:T-1}) = I(s_t;\theta \mid h_t)$. By Markov given θ, $s_t \perp s_{t+2:T} \mid s_{t+1}, \theta$, and so $I(s_t;s_{t+1:T} \mid \theta, h_t, a_{t:T-1}) = I(s_t;s_{t+1} \mid \theta, h_t, a_t)$ after dropping d-separated future actions. □

Vanishing residual term. Under assumption (iv), $p(\theta \mid h_t, a_{t:T-1}, s_{t+1:T}) \to \delta_{\theta^*}$ in probability as $T \to \infty$; further conditioning on s_t cannot move a delta, so the residual term vanishes. □

C.5 The Log-Ratio Subclass and Its Chain-Rule Decomposition

The proof of Theorem 1 settles the impossibility for the entire class $\mathcal{F}$ via the unconditional reduction. A complementary structural perspective is available for a specific sub-class of $\mathcal{F}$ — log-ratio rewards — whose expected value reduces to an observation-space mutual information rather than to an entropy. This subsection isolates that sub-class and gives the chain-rule decomposition that underlies the named confound terms in r^{sum}'s decomposition (Theorem 3) and in the $r^{\text{one}}(K)$ decomposition (Appendix C.6).

The log-ratio subclass $\mathcal{F}_{\text{LR}}$. For any linear combination of log likelihoods $r_t = \sum_i c_i \log \rho(X_i \mid Y_i) \in \mathcal{F}$, taking expectation under ρ^* gives

$$\mathbb{E}[r_t] = -\sum_i c_i \text{H}(X_i \mid Y_i) = -\sum_i c_i \text{H}(X_i) + \sum_i c_i I(X_i;Y_i).$$

The marginal-entropy contribution $\sum_i c_i \text{H}(X_i)$ vanishes identically iff, for each distinct random variable X appearing among the X_i's, the coefficients on terms with that X sum to zero. Rewards meeting this constraint can be written as a sum of *matched log-ratios*

$$r_t = \sum_k \alpha_k \left[\log \rho(X_k \mid Y_k^+) - \log \rho(X_k \mid Y_k^-)\right].$$

We denote this sub-class $\mathcal{F}_{\text{LR}}$. Membership:

- $r_t^{\text{sum}} \in \mathcal{F}_{\text{LR}}$: each summand is a paired log-ratio with $X_k = s_{t'}$ for the same future state.
- $r_t^{\text{one}}(K) \in \mathcal{F}_{\text{LR}}$: a single paired log-ratio with $X = s_{t+K}$.
- $r_t^{\text{sur}} \notin \mathcal{F}_{\text{LR}}$: its single unpaired term $-\log \rho(s_t \mid h_t)$ has expectation equal to the conditional entropy $\text{H}(s_t \mid h_t)$, with the marginal entropy of s_t surviving.

Chain-rule decomposition. For $r_t \in \mathcal{F}_{\text{LR}}$, the expected reward reduces to a signed sum of observation-space conditional MIs:

$$\mathbb{E}[r_t] = \sum_j \alpha_j I(A_j;B_j \mid C_j),$$

with A_j, B_j, C_j subsets of the trajectory and θ absent from all arguments (since ρ marginalizes over it). Each term decomposes via the chain rule against θ:

$$I(A_j;B_j \mid C_j) = \underbrace{I(A_j;\theta \mid C_j)}_{\text{signal-like}} - \underbrace{I(A_j;\theta \mid C_j, B_j)}_{\text{residual}} + \underbrace{I(A_j;B_j \mid \theta, C_j)}_{\text{abductive}}.$$

Reading the three terms.

- Signal-like terms quantify how the observation subsets A_j inform θ given the conditioning C_j.
- Residual terms measure the θ-information that A_j retains after B_j is also observed.
- Abductive terms capture dependencies between observation subsets that operate through the kernel's dynamics rather than through their shared dependence on θ.

This decomposition is exactly what produces the named offsets in Theorem 3 (one-step abductive and residual for r^{sum}) and in Theorem 5 (K-step abductive and residual for $r^{\mathrm{one}}(K)$). It does *not* extend to rewards outside $\mathcal{F}_{\mathrm{LR}}$: r^{sur} in particular requires its own analysis via the entropy chain rule $\mathrm{H}(s_t \mid h_t) = \mathrm{H}(s_t \mid h_t, \theta) + I(s_t; \theta \mid h_t)$, yielding the BIG-plus-aleatoric form of Theorem 2.

Relation to Theorem 1. For log-ratio rewards, the chain-rule decomposition gives a useful diagnostic of *where* the bias relative to BIG comes from: typically a non-vanishing abductive or a non-vanishing residual. It is not, however, the engine of Theorem 1. The proof above (Appendix C.2) operates at the level of unconditional expectations and works for the entire class $\mathcal{F}$ uniformly, including non-log-ratio rewards, without invoking the chain-rule decomposition.

C.6 NDIGO reward Decomposition

The NDIGO reward of Azar et al. (2019) is the single-summand analogue of r^{sum} at horizon K. We thus refer to it as $r^{\mathrm{one}}(K)$, and define in the ρ setting as:

$$r_t^{\mathrm{one}}(K) \coloneqq \log \rho(s_{t+K} \mid h_t, s_t, a_{t:t+K-1}) - \log \rho(s_{t+K} \mid h_t, a_{t:t+K-1}).$$

Theorem 5 (Decomposition of $r^{\mathrm{one}}(K)$)**.** *Under assumptions (i)–(iii) and (v),*

$$\begin{aligned} \mathbb{E}[r_t^{\mathrm{one}}(K) \mid h_t, a_{t:t+K-1}] = {} & I(s_t; \theta \mid h_t) \\ & + I(s_t; s_{t+K} \mid \theta, h_t, a_{t:t+K-1}) \\ & - I(s_t; \theta \mid h_t, a_{t:t+K-1}, s_{t+K}). \end{aligned}$$

Proof. By assumption (v), $\rho(s_t \mid h_t, a_{t:t+K-1}) = \rho(s_t \mid h_t)$, hence $r_t^{\mathrm{one}}(K) = \mathrm{pmi}(s_t; s_{t+K} \mid h_t, a_{t:t+K-1})$ (where pmi is point-wise mutual information). Take expectation and apply the chain rule against θ as in the proof of Theorem 3. □

At finite K the residual $I(s_t; \theta \mid h_t, a_{t:t+K-1}, s_{t+K})$ does not vanish: a single K-step-ahead observation does not identify θ. $r^{\mathrm{one}}(K)$ is therefore doubly biased relative to $I(s_t; \theta \mid h_t)$ at finite K (abductive plus persistent residual), and under BED structure reduces to $I(s_t; \theta \mid h_t)$ minus a residual. In the limit $K \to T - t$, $r^{\mathrm{one}}(K)$ approaches a single-summand analogue of r^{sum}.

C.7 Beyond Finite Touch Horizon: The Gap Reward

The decomposition theorems above place r^{sum} and $r^{\mathrm{one}}(K)$ at *opposite* structural extremes within the log-ratio subclass:

- r_t^{sum} queries the entire future block $s_{t+1:T}$ with *no gap*. Its epistemic residual vanishes as $T \to \infty$ (the block contains infinite data, identifying θ), but its one-step abductive $I(s_t; s_{t+1} \mid \theta, h_t, a_t)$ persists — the block starts immediately at $t+1$, fully coupled to s_t through the Markov kernel.
- $r_t^{\mathrm{one}}(K)$ queries a *single state* s_{t+K} at gap K. For $K \to \infty$ in a mixing chain, the abductive $I(s_t; s_{t+K} \mid \theta, h_t, a_{t:t+K-1})$ vanishes; but the epistemic residual $I(s_t; \theta \mid h_t, a_{t:t+K-1}, s_{t+K})$ persists — a single state cannot identify θ.

r^{sum} controls the residual; r^{one} controls the abductive. Each pushes one of two independent structural dials. This invites a unification: query a block that simultaneously starts at a gap K and extends to the trajectory horizon T.

Definition 2 (Gap reward)**.** For $1 \le K \le T - t$,

$$r_t^{\mathrm{gap}}(K) \coloneqq \log \rho\big(s_{t+K:T} \mid h_t, s_t, a_{t:T-1}\big) - \log \rho\big(s_{t+K:T} \mid h_t, a_{t:T-1}\big).$$

The two limits of Definition 2 recover the rewards above. At $K=1$, the conditioning block extends from $t+1$ to T and Lemma 4.1 together with assumption (iii) yields $r_t^{\mathrm{gap}}(1)=r_t^{\mathrm{sum}}$ exactly. Restricting the block of $r_t^{\mathrm{gap}}(K)$ to its first state s_{t+K} recovers $r_t^{\mathrm{one}}(K)$. r^{gap} is the two-parameter family that interpolates between them by independently choosing a gap K and a block size $T-t-K+1$.

Beyond assumptions (i)–(v), the gap-reward analysis crucially uses assumption (vi): mixing of the chain given θ, ruling out absorbing states and other non-ergodic dynamics under which the chain never forgets s_t.

Theorem 6 (Decomposition of r^{gap})**.** *Under (i)–(iii) and (v),*

$$\begin{aligned}\mathbb{E}[r_t^{\mathrm{gap}}(K)\,|\,h_t,a_{t:T-1}] &= I(s_t;\theta\,|\,h_t)\\ &\quad+\underbrace{I(s_t;s_{t+K:T}\,|\,\theta,h_t,a_{t:T-1})}_{\text{abductive at gap } K}\\ &\quad-\underbrace{I(s_t;\theta\,|\,h_t,a_{t:T-1},s_{t+K:T})}_{\text{residual at block end } T}.\end{aligned}$$

Both correction terms are non-negative.

Proof. By Lemma 4.1, $r_t^{\mathrm{gap}}(K)=\mathrm{pmi}(s_t;s_{t+K:T}\,|\,h_t,a_{t:T-1})$; by (v), $\rho(s_t\,|\,h_t,a_{t:T-1})=\rho(s_t\,|\,h_t)$. Take expectation and apply the chain rule against θ as in the proof of Theorem 3:

$$\begin{aligned}I(s_t;s_{t+K:T}\,|\,h_t,a_{t:T-1}) &= I(s_t;\theta\,|\,h_t,a_{t:T-1})\\ &\quad- I(s_t;\theta\,|\,h_t,a_{t:T-1},s_{t+K:T})\\ &\quad+ I(s_t;s_{t+K:T}\,|\,\theta,h_t,a_{t:T-1}).\end{aligned}$$

By (v), $I(s_t;\theta\,|\,h_t,a_{t:T-1})=I(s_t;\theta\,|\,h_t)$. □

Theorem 6 subsumes Theorem 3 ($K=1$) and reveals that the abductive of r^{gap} is a K-shifted analogue of the one-step abductive of r^{sum}. The decomposition immediately yields a positive counterpart to the impossibility theorem in the iterated infinite limit.

Corollary 6.1 (r^{gap} recovers BIG in the double limit)**.** *Under (i)–(vi),*

$$\lim_{K\to\infty}\lim_{T\to\infty}\mathbb{E}[r_t^{\mathrm{gap}}(K)\,|\,h_t,a_{t:T-1}] = I(s_t;\theta\,|\,h_t).$$

Proof. Inner limit ($T\to\infty$, K fixed). As $T\to\infty$ the conditioning block $s_{t+K:T}$ grows without bound. Under (iv), $p(\theta\,|\,h_t,a_{t:T-1},s_{t+K:T})\to\delta_{\theta^*}$ in probability, so the residual $I(s_t;\theta\,|\,h_t,a_{t:T-1},s_{t+K:T})\to 0$.

Outer limit ($K\to\infty$). With the residual at zero, only the abductive remains. By (i), the dependence of $s_{t+K:T}$ on s_t given θ funnels through s_{t+K}: $I(s_t;s_{t+K:T}\,|\,\theta,h_t,a_{t:T-1})=I(s_t;s_{t+K}\,|\,\theta,h_t,a_{t:t+K-1})$ (data-processing along the post-s_{t+K} chain). By (vi), this vanishes as $K\to\infty$. □

Remarks.

- *Outside $\mathcal{F}$.* The limit reward in Corollary 6.1 cannot be expressed as a function of finitely many likelihoods, hence not a member of $\mathcal{F}$. This is consistent with Theorem 1: at every finite T,K, $r_t^{\mathrm{gap}}(K)\in\mathcal{F}_{\mathrm{LR}}\subset\mathcal{F}$ and inherits the bias of Theorem 6; only the asymptotic limit escapes $\mathcal{F}$.
- *Marginalization burden.* Computing $r_t^{\mathrm{gap}}(K)$ requires ρ to evaluate predictives on histories where s_t is masked and the intermediate states $s_{t+1:t+K-1}$ are absent — effectively asking ρ to marginalize over a gap of $K-1$ unobserved states. Assumption (iii) — that ρ's response to such queries is the correctly marginalized posterior-predictive — becomes increasingly strained as the gap grows; see Appendix B.3.
- *Mixing failure.* Assumption (vi) fails for environments with absorbing states or other non-ergodic structure given θ. In such cases the abductive does not decay with K and the double limit need not yield BIG.

C.8 Necessity boundlessly growing gaps for Log-Ratio Rewards

Corollary 6.1 shows that an infinite gap together with an infinite block is *sufficient* to recover BIG within a log-ratio reward. We now prove the converse: the same double-limit geometry is also *necessary*. Concretely, any countable linear combination of matched log-ratios that universally identifies BIG must place all of its weight at infinite gap, with each non-trivial block of infinite size.

Let $\mathcal{F}_{\mathrm{LR},T}$ (cf. Appendix C.5) be the set of estimators which are linear combinations of log ratios involving subsets of observations up to a horizon T, i.e.,

$$\mathcal{F}_{\mathrm{LR},T} = \left\{ \sum_{Z \subseteq \{t+1,\dots,T\}} c_Z \Big[\log\rho(s_Z \,|\, h_t, s_t, a_{t:T}) - \log\rho(s_Z \,|\, h_t, a_{t:T}) \Big] : c_Z \in \mathbb{R}, \forall Z \subseteq \{t+1,\dots,T\} \right\}.$$

The following theorem shows that for any sequence of estimators chosen from $\mathcal{F}_{\mathrm{LR},T}$ which is asymptotically equal to the Bayesian Information Gain $I(s_t;\theta|h_t)$ as $T\to\infty$ in a universal fashion over all BAMDPs, this sequence must involve an unboundedly growing gap akin to the one discussed in Corollary 6.1.

Theorem 7 (Necessity of a growing gap)**.** *Let $\mathcal{M}$ be the class of BAMDPs satisfying (i)–(vi) whose transition kernels are non-deterministic and not independent across time given θ. Suppose that $(R_{t,T})_T$ is a sequence of estimators such that*

- *$R_{t,T} \in \mathcal{F}_{\mathrm{LR},T}$ for all T, i.e.,*

$$R_{t,T} = \sum_{Z \subseteq \{t+1,\dots,T\}} c_{Z,T} \Big[\log\rho(s_Z \,|\, h_t, s_t, a_{t:T}) - \log\rho(s_Z \,|\, h_t, a_{t:T}) \Big],$$

- *$\lim_{T\to\infty} \mathbb{E}^M[R_{t,T} \,|\, h_t, a_{t:T}] = I(s_t;\theta \,|\, h_t)$ for every $M \in \mathcal{M}$ and every $h_t, a_{t:\infty}$, and*
- *there exist constants $C > 0$ and $B \geq 1$ such that $|W_{k,T}| \leq \sum_{k_Z=k} |c_{Z,T}| \leq C B^k$ for all k and T.*[2]

If we define for each block of observations $Z \subset \{t+1,\dots\}$ the gap $k_Z := \min\{k \geq 1 : t+k \in Z\}$ and write

$$W_{k,T} := \sum_{\substack{Z \subseteq \{t+1\dots,T\}: \\ k_Z = k}} c_{Z,T},$$

then

$$\lim_{T\to\infty} W_{k,T} = 0.$$

In other words, as $T\to\infty$, almost all the weight $\sum_Z c_{Z,T}$ "will become concentrated on blocks of infinite gap".

Proof. **Step 1: Chain-Rule Decomposition.**
By definition, each matched log-ratio term in $\mathcal{F}_{\mathrm{LR},T}$ evaluates to a pointwise mutual information. Taking the expectation of $R_{t,T}$ and applying the chain rule against the environment parameter θ, we obtain:

$$\begin{aligned} \mathbb{E}^M[R_{t,T} \,|\, h_t, a_{t:T}] &= \sum_Z c_{Z,T} I(s_t; s_Z \,|\, h_t, a_{t:T}) \\ &= \Big(\sum_Z c_{Z,T} \Big) I(s_t;\theta \,|\, h_t) + \sum_Z c_{Z,T} A_Z - \sum_Z c_{Z,T} E_Z, \end{aligned}$$

where $A_Z := I(s_t; s_Z \,|\, \theta, h_t, a_{t:T})$ is the abductive term and $E_Z := I(s_t;\theta \,|\, h_t, a_{t:T}, s_Z)$ is the epistemic residual.

Step 2: Isolating the Abductive Bias.
The abductive term A_Z captures dependencies generated purely by the known transition kernel, relying on the prior $p(\theta)$ only through the posterior $p(\theta \,|\, h_t)$. We may evaluate the limit on an environment $M \in \mathcal{M}$ whose prior is a degenerate point mass at some θ^*. For such a prior, there is no uncertainty about the

[2] It is worth noting that this is a mild assumption. In practice, numerically stable estimators must satisfy this.

environment, so the true Bayesian Information Gain $I(s_t;\theta \mid h_t)=0$ and all epistemic residuals $E_Z=0$ identically. The universal convergence hypothesis then reduces to:

$$\lim_{T\to\infty}\sum_{Z} c_{Z,T}A_Z(\theta^*)=0 \quad \text{for every } \theta^*. \tag{7}$$

Step 3: The Markov Funnel.
Under assumption (i), conditional on the exact parameter θ^*, the environment is Markovian. By the data-processing inequality, the dependence of any block s_Z on the state s_t must funnel strictly through the chronologically earliest state in that block, s_{t+k_Z}. Dropping the separated future actions, we have:

$$A_Z(\theta^*)=I(s_t;s_{t+k_Z}\mid \theta^*,h_t,a_{t:t+k_Z-1}) \coloneqq I_{k_Z}(\theta^*).$$

Grouping the terms in equation 7 by their gap $k_Z=k$, we rewrite the limit in terms of the gap weights $W_{k,T}$:

$$\lim_{T\to\infty}\sum_{k=1}^{T} W_{k,T}I_k(\theta^*)=0 \quad \text{for every } \theta^*. \tag{8}$$

Step 4: AR(1) Instantiation and Power Series Extraction.
To evaluate equation 8, we instantiate M with the Gaussian AR(1) family parameterized by $\alpha\in(0,1)$, for which the mutual information across a gap k is $I_k(\alpha)=-\frac{1}{2}\log(1-\alpha^{2k})$. Letting $x=\alpha^2\in(0,1)$, we define the sequence of functions:

$$F_T(x)\coloneqq\sum_{k=1}^{T} W_{k,T}\big(-\log(1-x^k)\big).$$

By hypothesis, $\lim_{T\to\infty}F_T(x)=0$ for all $x\in(0,1)$. We extend $F_T(x)$ to the complex plane as $F_T(z)$, which is analytic on the open unit disk $|z|<1$. Expanding the principal branch of the logarithm in its Maclaurin series $-\log(1-z^k)=\sum_{m=1}^{\infty}\frac{z^{km}}{m}$, we express $F_T(z)$ as:

$$F_T(z)=\sum_{k=1}^{T} W_{k,T}\sum_{m=1}^{\infty}\frac{z^{km}}{m}=\sum_{n=1}^{\infty}\frac{V_{n,T}}{n}z^n,$$

where we have grouped terms by $n=km$ and defined $V_{n,T}\coloneqq\sum_{d\mid n,\, d<T} dW_{d,T}$.

Now recall that the theorem statement assumes that there exist constants $C>0$ and $B\geq 1$ such that $|W_{k,T}|\leq\sum_{k_Z=k}|c_{Z,T}|\leq CB^k$ for all k and T. For z strictly inside the unit disk, we may bound the logarithmic term using its Maclaurin series: $|-\log(1-z^k)|\leq\sum_{m=1}^{\infty}|z|^{km}/m\leq|z|^k/(1-|z|)$. Applying our exponential bound to the weights yields:

$$|F_T(z)|\leq\sum_{k=1}^{T}|W_{k,T}||-\log(1-z^k)|\leq\frac{C}{1-|z|}\sum_{k=1}^{\infty}(B|z|)^k.$$

This geometric series converges absolutely for $B|z|<1$. Thus, defining a bounding radius $R\coloneqq\min(1/2,1/B)$, the sequence of functions $\{F_T(z)\}$ is uniformly bounded by a finite constant on any closed ball $B(0,r)$ with $r<R$.

Because $F_T(x)\to 0$ pointwise on the real interval $(0,R)$, and this interval contains an accumulation point strictly within $B(0,R)$, the Vitali-Porter Convergence Theorem (Schiff, 1993, Section 2.4) guarantees that $F_T(z)$ converges uniformly to 0 on any strictly smaller closed contour γ enclosing the origin. By Cauchy's Integral Formula, the n-th Maclaurin coefficient is given by:

$$\frac{V_{n,T}}{n}=\frac{1}{2\pi i}\oint_{\gamma}\frac{F_T(z)}{z^{n+1}}\mathrm{d}z.$$

Since $F_T(z)$ converges uniformly to 0 on γ, we may pass the limit inside the integral, yielding:

$$\lim_{T\to\infty}\frac{V_{n,T}}{n}=0\Longrightarrow\lim_{T\to\infty}V_{n,T}=0 \quad \text{for all } n\geq 1.$$

Step 5: Strong Induction.
We prove $\lim_{T\to\infty}W_{k,T}=0$ by strong induction on k. For the base case $k=1$, we have $V_{1,T}=W_{1,T}$

(since $d < T$ trivially for $T > 1$), and thus $\lim_{T\to\infty} W_{1,T} = 0$. For the inductive step, assume that $\lim_{T\to\infty} W_{d,T} = 0$ for all proper divisors $d < n$. By definition, for $T > n$:

$$V_{n,T} = nW_{n,T} + \sum_{\substack{d|n \\ d<n}} dW_{d,T}.$$

Taking the limit as $T \to \infty$:

$$0 = n \lim_{T\to\infty} W_{n,T} + \sum_{\substack{d|n \\ d<n}} d\underbrace{\Big(\lim_{T\to\infty} W_{d,T}\Big)}_{=0} \implies \lim_{T\to\infty} W_{n,T} = 0.$$

This holds for all integers $n \geq 1$, proving that all finite gap weights strictly vanish as $T \to \infty$. Consequently, any estimator universally approaching BIG must concentrate its weight strictly at an unbounded, infinite gap.

□

Remark (relation to Theorem 1). The earlier impossibility theorem covers $\mathcal{F}$ at every finite touch horizon Q: at finite T, no member of $\mathcal{F}$ identifies BIG, even without log-ratio structure. Theorem 7 extends the negative result *into* the limit: even allowing countable log-ratio combinations and asymptotic T, the only escape is a structural double limit in which both the gap and the block grow without bound. As mentioned in the main text, this imposes severe implementation limitations since predictive models ρ are well suited for long sequence modeling (large T) but less so for marginalization over unobserved sequence blocks (large gaps).

C.9 Proof of Corollary 3.1

Corollary 3.1 (Decomposition of r^{sum} in BED)**.** *Under (i)–(v),*

$$\mathbb{E}[r_t^{\mathrm{sum}} \,|\, h_t, a_{t:T-1}] = I(s_t;\theta \,|\, h_t) - I(s_t;\theta \,|\, h_t, a_{t:T-1}, s_{t+1:T}),$$

i.e., the abductive of Theorem 3 vanishes structurally. As $T \to \infty$, $\mathbb{E}[r_t^{\mathrm{sum}}] \to I(s_t;\theta \,|\, h_t)$.

Proof. Under BED structure, given θ and the action sequence, $s_t \sim p(\cdot \,|\, a_{t-1}, \theta)$ and $s_{t+1} \sim p(\cdot \,|\, a_t, \theta)$ are independent. Hence $I(s_t; s_{t+1} \,|\, \theta, h_t, a_t) = 0$, and the abductive term in Theorem 3 drops out. □

C.10 Proof of Theorem 4

Theorem 4 (Decomposition of r^{dl} in BED)**.** *Under (i)–(v), with $L(\theta) := p(s_t \,|\, a_{t-1}, \theta)/\rho(s_t \,|\, h_t)$,*

$$\mathbb{E}[r_t^{\mathrm{dl}} \,|\, h_t] = I(s_t;\theta \,|\, h_t) + \mathbb{E}_{s_t|a_{t-1}}\Big[\log \mathbb{E}_{p(\theta|h_t,s_t)}[L(\theta)] - \mathbb{E}_{p(\theta|h_t,s_t)}[\log L(\theta)]\Big],$$

i.e., BIG plus a non-negative Jensen gap. Both terms in the expectation vanish as $t \to \infty$ under (iv).

Proof. Under BED, $p(s_t \,|\, s_{t-1}, a_{t-1}, \theta) = p(s_t \,|\, a_{t-1}, \theta)$, so the integral in Equation (4) simplifies:

$$\int p(s_t \,|\, a_{t-1}, \theta) p(\theta \,|\, h_t, s_t) d\theta = \mathbb{E}_{p(\theta|h_t,s_t)}[p(s_t \,|\, a_{t-1}, \theta)] = \rho(s_t \,|\, h_t) \mathbb{E}_{p(\theta|h_t,s_t)}[L(\theta)],$$

where the last equality uses the definition $L(\theta) = p(s_t \,|\, a_{t-1}, \theta)/\rho(s_t \,|\, h_t)$. Hence

$$r_t^{\mathrm{dl}} = \log \mathbb{E}_{p(\theta|h_t,s_t)}[L(\theta)].$$

By Jensen's inequality,

$$r_t^{\mathrm{dl}} = \mathbb{E}_{p(\theta|h_t,s_t)}[\log L(\theta)] + \mathcal{J}_t,$$

with $\mathcal{J}_t \geq 0$ the Jensen gap. Bayes' rule gives $L(\theta) = p(\theta \mid h_t, s_t)/p(\theta \mid h_t)$, so $\mathbb{E}_{p(\theta|h_t,s_t)}[\log L(\theta)] = D_{\mathrm{KL}}\big(p(\theta \mid h_t, s_t) \,\|\, p(\theta \mid h_t)\big)$. Taking expectation over s_t, the KL term integrates to $I(s_t;\theta \,|\, h_t)$ (Bayesian-surprise identity). □

Remark. Theorem 4 shows an important limitation of r^{dl}. While technically, r^{dl} does recover BIG asymptotically, it does so as $t\to\infty$ which also implies that BIG $I(s_t;\theta\,|\,h_t)$ itself vanishes. This means that r^{dl} biases vanish at the same time as the BIG signal itself does. In contrast, r^{sum} also recovers BIG asymptotically, but it does so as $T\to\infty$, meaning that for long trajectories, one can hope to recover minimally biased BIG signal for finite t. However, the practical effect of finite-time biases may or may not be consequential for a useful policy, depending on the environment. Experiments in the main text illustrate this point.

Appendix D Implementing r^{dl} in General BAMDPs vs. BED

The reward r^{dl} defined in Equation (4) contains an integral term

$$\int p(s_t\,|\,s_{t-1},a_{t-1},\theta)p(\theta\,|\,h_t,s_t)d\theta$$

that pairs the kernel at the trajectory input (s_{t-1},a_{t-1}) with a posterior conditioned on the subsequent observation s_t. We explain here why this combination is generically not implementable through ρ's standard predictive interface, and how the BED restriction restores implementability via a counterfactual action commitment.

Why no ρ-implementation in general BAMDPs. A standard ρ query $\rho(s'\,|\,h)$ evaluates the kernel at the most recent (s,a) in the conditioning history h, averaged under the posterior $p(\theta\,|\,h)$ that this same history induces. The integrand above departs from this template in a structural way: the kernel input is (s_{t-1},a_{t-1}) from time $t-1$, while the posterior is $p(\theta\,|\,h_t,s_t)$ from after s_t has been observed. In a general MDP, where the kernel genuinely depends on the chain-position state, no rearrangement of conditioning subsets in ρ's query interface produces this combination — evaluating ρ at (s_{t-1},a_{t-1}) gives the posterior only up to time $t-1$, and evaluating ρ after s_t shifts the kernel input as well. Implementing r^{dl} in general BAMDPs therefore requires explicit Bayesian-update machinery beyond ρ's forward-pass interface.

BED counterfactual identity. In BED the transition kernel has no s_{t-1} dependence, $p(s_t\,|\,s_{t-1},a_{t-1},\theta)=p(s_t\,|\,a_{t-1},\theta)$, so the integrand becomes

$$\int p(s_t\,|\,a_{t-1},\theta)p(\theta\,|\,h_t,s_t)d\theta=\rho\big(s_{t+1}=s_t\big|h_t,s_t,a_t=a_{t-1}\big),$$

recovering Equation (6). The right-hand side is a standard ρ query: condition on the history (h_t,s_t), commit to the action $a_t=a_{t-1}$ (a counterfactual on the action choice, valid because actions are policy-generated and thus interventional), and read off the predictive probability of the next outcome equaling the observed s_t. This makes r^{dl} implementable in BED at the cost of one hypothetical rollout step in ρ's context.

The key takeaway: BED collapses the chain-time mismatch by removing the s_{t-1} dependence in the kernel — an obstruction that no rearrangement of ρ's queries can resolve in a general MDP. It could be possible to address this issue with a modified sequence model architecture that allows multiple channels for the same sequence location, together with a gating mechanism to implement a form of counterfactual "superposition". This remains a non-trivial design.

Appendix E Experimental Details

E.1 Gaussian Processes

E.1.1 Environment

The Gaussian Process (GP) environment presents the agent with an unknown function $f:\mathbb{R}^2\to\mathbb{R}$ that must be explored through sequential point queries. At the start of each episode, a function f is sampled from a Gaussian process prior $f\sim\mathcal{GP}(0,k)$, where k is the *rational quadratic* kernel:

$$k(\mathbf{x},\mathbf{x}')=\sigma^2\left(1+\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\alpha\ell^2}\right)^{-\alpha}, \tag{9}$$

with $\alpha=1$, lengthscale $\ell=4.0$, and $\sigma^2=1$.

Since a GP sample is an infinite-dimensional function, we represent it in practice via *finite-grid sampling*: we place $R=10$ equally spaced points along each of the d input dimensions in $[-x_{\max},,x_{\max}]$ and take their Cartesian product, yielding a grid $\mathbf{X}_{grid}$ of $N=R^d$ locations. To draw a sample $\mathbf{y}_{\text{grid}}$ from this multivariate Gaussian we use the standard Cholesky method. Function values at arbitrary query points are then obtained via GP posterior interpolation. We use $x_{\max}=10$.

When the agent queries location $\mathbf{x}_t$, it observes:

$$y_t=f(\mathbf{x}_t)+\epsilon_t, \qquad \epsilon_t\sim\mathcal{N}\big(0,\sigma^2_{\text{noise}}(\mathbf{x}_t)\big), \tag{10}$$

where the noise variance $\sigma^2_{\text{noise}}(\mathbf{x})$ is *spatially varying* according to a fixed checkerboard noise map. $\mathbf{X}_{grid}$ is partitioned into an 8×8 grid of cells; Tiles alternate between high-noise ($\sigma^2_{\text{noise}}=5$) and low-noise ($\sigma^2_{\text{noise}}\approx 0$) regions. This creates a heteroscedastic noise landscape that an effective exploration policy must learn to navigate.

E.1.2 Bayesian Oracle World Model

The Bayesian oracle leverages the closed-form GP posterior. Given a history of observations $\{(\mathbf{x}_i,y_i)\}_{i=0}^{t-1}$, the posterior predictive distribution for a new query $\mathbf{x}_t$ is Gaussian:

$$p(y_t\,|\,\mathbf{x}_t,h_t)=\mathcal{N}\big(y_t;\mu_t,\sigma_t^2\big), \tag{11}$$

where the posterior mean and variance are given by the standard GP regression formulae:

$$\mu_t=\mathbf{k}_*^\top(K+\Sigma_{\text{noise}})^{-1}\mathbf{y}_{<t}, \tag{12}$$

$$\sigma_t^2=k(\mathbf{x}_t,\mathbf{x}_t)-\mathbf{k}_*^\top(K+\Sigma_{\text{noise}})^{-1}\mathbf{k}_*, \tag{13}$$

with $\mathbf{k}_*=[k(\mathbf{x}_i,\mathbf{x}_t)]_{i=0}^{t-1}$, $K_{ij}=k(\mathbf{x}_i,\mathbf{x}_j)$, and $\Sigma_{\text{noise}}=\text{diag}(\sigma^2_{\text{noise}}(\mathbf{x}_0),...,\sigma^2_{\text{noise}}(\mathbf{x}_{t-1}))$.

For the *predictive* distribution used by the surprisal-based rewards, the predictive variance additionally includes the noise at the query point:

$$\tilde{\sigma}_t^2=\sigma_t^2+\sigma^2_{\text{noise}}(\mathbf{x}_t). \tag{14}$$

Per-step Surprisal The observation surprisal is the negative log-likelihood under the predictive Gaussian:

$$-\log p(y_t\,|\,\mathbf{x}_t,h_t)=\frac{1}{2}\log(2\pi)+\frac{1}{2}\log\tilde{\sigma}_t^2+\frac{(y_t-\mu_t)^2}{2\tilde{\sigma}_t^2}. \tag{15}$$

Per-step Information Gain (Bayesian Surprise) The information gain reward uses the KL divergence between the GP posterior and the predictive observation distribution. In exact form:

$$\text{G}_t=\frac{1}{2}\log\left(1+\frac{\sigma_t^2}{\sigma^2_{\text{noise}}(\mathbf{x}_t)}\right)-\frac{\sigma_t^2}{2(\sigma_t^2+\sigma^2_{\text{noise}}(\mathbf{x}_t))}+\frac{\sigma_t^2\cdot(y_t-\mu_t)^2}{2(\sigma_t^2+\sigma^2_{\text{noise}}(\mathbf{x}_t))^2}, \tag{16}$$

where σ_t^2 is the latent GP posterior variance (without observation noise).

E.1.3 In-Context PFN World Model

Data-Generating Process The PFN is trained to predict the next observation y_t in-context from the history of observed pairs. For each training example, a function f is sampled from the same GP prior as the environment (rational quadratic kernel with $\alpha=1$, $\ell=4.0$, spatially-varying noise), and $T_{\text{train}}=100$ random query locations are drawn uniformly from $[-10,10]^2$.

Model Architecture The PFN uses a causal Transformer decoder. Each past observation (x_j,y_j) is embedded by summing a linear projection of x_j and a linear projection of y_j into a single "pair" token. This pair token is interleaved with an isolated embedding of the next query x_{j+1}, yielding a sequence of length $2(t-1)$. The causal Transformer decoder has model dimension $d=512$, MLP hidden dimension $F=1024$, $N=12$ layers, and $H=8$ attention heads with head dimension $d/H=64$. The model uses LayerNorm and GELU activations. The Transformer output at each query-token position (odd positions) is projected via a learned linear layer to produce a 2-dimensional output $(\hat{\mu}_t,\hat{\sigma}_t)$, representing a predicted mean and log-standard-deviation.

Training The training objective is the mean negative log-likelihood of the observed y values under the predicted Gaussian, with a variance-weighted loss to encourage better calibration (Seitzer et al., 2022):

$$\mathcal{L}=\frac{1}{(T-1)B}\sum_{b=1}^{B}\sum_{t=1}^{T-1}\left[\mathrm{sg}(\hat{\sigma}_t^2)\right]^{\beta}\cdot\left[\frac{1}{2}\log(2\pi\hat{\sigma}_t^2)+\frac{(y_t-\hat{\mu}_t)^2}{2\hat{\sigma}_t^2}\right], \tag{17}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient, $\hat{\sigma}_t^2=\exp(2\hat{\sigma}_t)$ is the predicted variance, and $\beta=1$. The model is optimized using Adam with a learning rate of 10^{-4}, batch size $B=128$, for 10^6 training steps.

We collect random trajectories from Gaussian processes, and plot the correlation between the predictions of the Bayesian oracle world model and the trained PFN world model in Figure E.1. The trained PFN is effective at approximating the exact likelihood and, as evidenced by other results, is effective even in settings where the oracle cannot be computed.

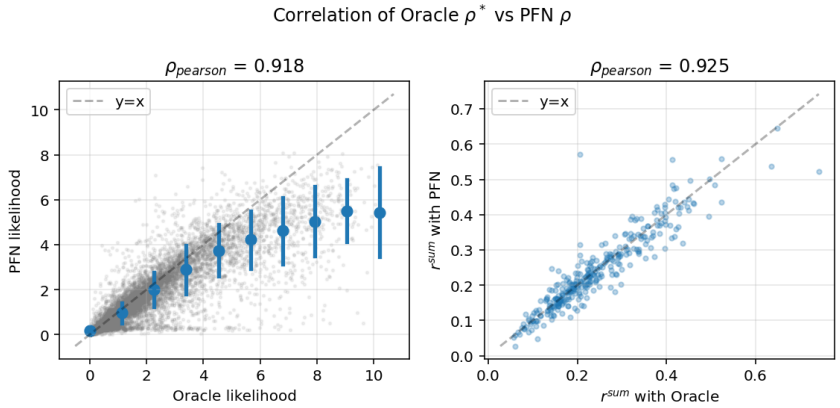

Figure E.1: The correlation between the values computed by an exact bayesian oracle vs our learned PFN for likelihood and reward of randomly generated trajectories from Gaussian processes.

E.1.4 Policy

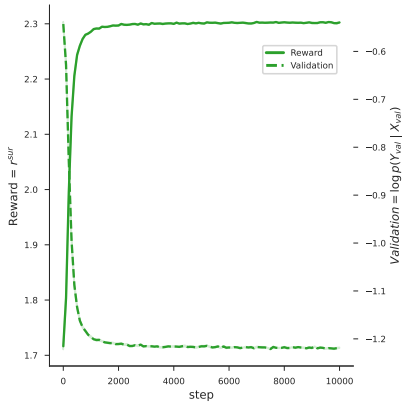

Figure E.2: Training reward vs Validation log-probability over the course of training a policy for Gaussian Process. This highlights that in noisy environments, r^{sur} can behave oppositely to the intended effect of learning the environment's dynamics.

The policy is trained using REINFORCE with a learned value baseline. At each training step, $B=32$ trajectories of length $T=26$ are collected by rolling out the current policy in the environment. A single trajectory-level reward R is computed from the world model, and the policy is updated with a single gradient step per batch. The REINFORCE objective combines a policy gradient loss and a value baseline loss:

$$\mathcal{L}=\mathcal{L}_\pi+\lambda_V\mathcal{L}_V, \tag{18}$$

where $\lambda_V=0.5$.

Model Architecture The policy is a sequence model that processes the history of observations recurrently. The backbone is a Griffin model (De et al., 2024), a recurrent architecture based on linear recurrences (LRU). The Griffin configuration uses model dimension $d=128$, MLP dimension 256, LRU dimension 128, 4 attention heads, and 2 recurrent layers.

At each step t, the observation $o_t=(\mathbf{x}_t,y_t)$ is encoded by concatenating $\mathbf{x}_t\in\mathbb{R}^2$ and $y_t\in\mathbb{R}$, yielding a 3-dimensional vector, which is projected to the model dimension $d=128$ via a learned linear layer.

The sequence model output at each step is mapped to logits over the 64 discrete grid locations via a single linear layer. A separate linear layer maps the sequence model output to a scalar value estimate $V(s_t)\in\mathbb{R}$.

E.2 Decomposition of rewards

In Section 4.3, we analyzed the theoretical behaviour of curiosity rewards in BED environments. Corollary 3.1 showed that r^{sum} decomposes into BIG$-$Residual terms, while Theorem 4 showed that r^{dl} decomposes into BIG$+$JensenGap terms. In our Gaussian process environment using an Bayesian oracle observer, all terms can be computed analytically. After having trained a policy with either r^{sum} or r^{dl}, we can therefore check whether the reward empirically decomposes into the expected terms that our theory predicts.

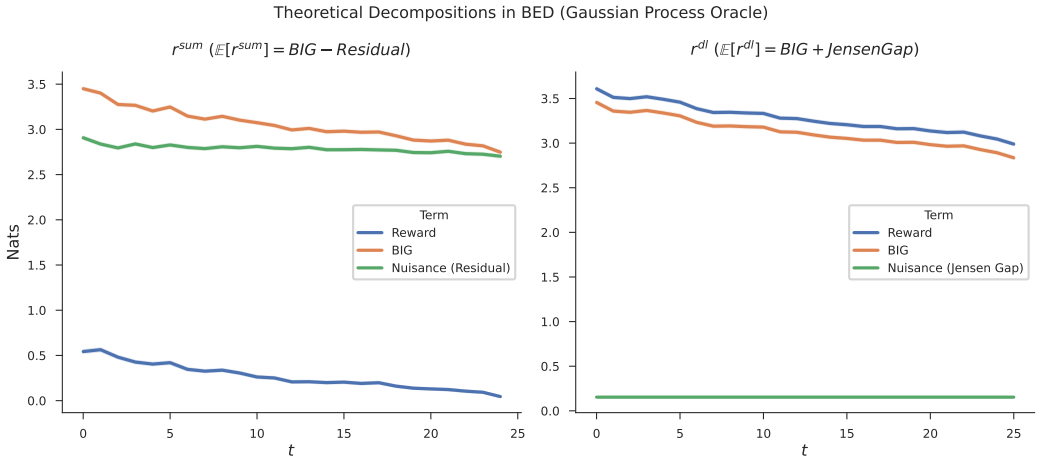

Figure E.3: Empirical decompositions of (i) r^{sum} into BIG and a subtracted residual term and (ii) r^{dl} into an additive Jensen gap, in a Gaussian process environment with oracle in-context learner.

Figure E.3 plots the resulting reward, BIG, and nuisance term as a function of time, averaged across 10 training seeds and 1000 unrolled trajectories. We find that both rewards decompose *exactly* into their theoretical constituents, thus validating our theoretical analyses. Indeed, plotting BIG$-$Residual overlaps exactly with r^{sum}, and plotting BIG$+$JensenGap overlaps exactly with r^{dl}.

E.3 Mastermind

At the start of each episode, the environment samples a secret code $\mathbf{c}^*\in\{0,1,\ldots,C-1\}^L$ uniformly at random, where C is the number of colours and L is the code length. In our experiments we set $C=6$ and $L=7$, yielding $6^7=279{,}936$ possible codes. We hold out 10% of the possible codes for evaluation.

At each timestep t, the agent submits a guess $\mathbf{g}_t\in\{0,1,\ldots,C-1\}^L$ and receives a response (b_t,w_t) where:

- Black peg $b_t\in\{0,\ldots,L\}$ is the number of *exact matches* (correct colour in the correct position), and
- White peg $w_t\in\{0,\ldots,L\}$ is the number of *colour matches* (correct colour regardless of position), computed as $w_t=\sum_{c=0}^{C-1}\min\big(\mathrm{count}(c,\mathbf{c}^*),\mathrm{count}(c,\mathbf{g}_t)\big)$, where $\mathrm{count}(c,\mathbf{v})$ is the number of occurrences of colour c in vector $\mathbf{v}$.

Note that $w_t\geq b_t$ always holds, and colours are not double-counted: if the code contains N repeats of a colour but the guess has $M>N$ entries of that colour, only N contribute to w_t.

The episode proceeds for a fixed trajectory length of $T=8$ guesses. A random initial guess is generated at the start of each episode and forms the first observation.

E.3.1 Bayesian Oracle World Model

The Bayesian oracle world model maintains the exact posterior over secret codes by enumerating over all C^L possible codes. Given a history of guesses and responses $\{(\mathbf{g}_1,b_1,w_1),...,(\mathbf{g}_{t-1},b_{t-1},w_{t-1})\}$, the oracle computes the set of *consistent codes*—those that would have produced the observed responses for all past guesses. Formally, a code $\mathbf{c}$ is consistent at step t if for all $i<t$: $\text{exact}(\mathbf{c},\mathbf{g}_i)=b_i$ and $\text{colour}(\mathbf{c},\mathbf{g}_i)=w_i$.

The uniform prior over codes, combined with consistency filtering, yields the posterior probability of a newly observed response (b_t,w_t) given a new guess $\mathbf{g}_t$ as:

$$p(b_t,w_t \mid h_t=[h_{t-1},s_{t-1},\mathbf{g}_t])=\frac{|\{\mathbf{c}\in\mathcal{C}_t:\text{exact}(\mathbf{c},\mathbf{g}_t)=b_t\wedge\text{colour}(\mathbf{c},\mathbf{g}_t)=w_t\}|}{|\mathcal{C}_t|}, \tag{19}$$

where $\mathcal{C}_t$ denotes the set of codes consistent with all observations prior to step t.

E.3.2 In-Context PFN World Model

Data-Generating Process The PFN is trained on synthetically generated game trajectories. For each training example, a secret code is sampled uniformly at random, and then a sequence of $T_{\text{train}}=50$ random guesses is generated. Each guess is drawn uniformly from $\{0,...,C-1\}^L$, and the corresponding responses (b_t,w_t) are computed by the environment.

Model Architecture The in-context world model is a causal Transformer decoder that processes interleaved guess and response tokens. Specifically, each guess $\mathbf{g}_t$ is one-hot encoded over the C colours for each of the L positions, yielding a vector of dimension $C\times L$, which is then projected to the model dimension d via a linear layer. Each response (b_t,w_t) is one-hot encoded as a concatenation of two vectors of dimension $(L+1)$ each, giving a total dimension of $2(L+1)$, which is projected to d via a separate linear layer. For a trajectory of T steps, the input sequence is constructed by interleaving *pairs* and *isolated guesses*. A pair token is formed by summing the guess and response embeddings for the same timestep: $\mathbf{p}_t=\text{embed}_g(\mathbf{g}_t)+\text{embed}_r(b_t,w_t)$. The isolated guess token for the next step is $\text{embed}_g(\mathbf{g}_{t+1})$. These are interleaved as $[\mathbf{p}_1,\text{embed}_g(\mathbf{g}_2),\mathbf{p}_2,\text{embed}_g(\mathbf{g}_3),...]$, resulting in a sequence of length $2(T-1)$.

The sequence is processed by a causal (left-to-right) Transformer decoder with the model dimension $d=512$, MLP hidden dimension = 1024, 12 layers, and 8 attention heads with head dimension = 64. The Transformer output at every odd position (i.e., the isolated guess positions) is used to predict the corresponding response. A linear readout maps the Transformer output to $(L+1)^2$ logits, representing the joint distribution over all possible (b_t,w_t) pairs via a softmax. The target is the index $b_t\cdot(L+1)+w_t$.

The in-context world model is trained to minimize the cross-entropy loss of predicting the response at each step. In each training step, a fresh batch of $B=128$ random trajectories is generated on-the-fly (no fixed dataset).

E.3.3 Policy

The policy is trained using REINFORCE with a learned value baseline. At each training step, a batch of $B=32$ trajectories of length $T=8$ is collected by rolling out the current policy in the environment. The total loss for REINFORCE is:

$$\mathcal{L}=\mathcal{L}_\pi+\lambda_V\mathcal{L}_V, \tag{20}$$

where $\lambda_V=0.5$. When using the in-context world model (PFN observer), an entropy bonus $-\lambda_H\mathcal{H}[\pi_\phi]$ with $\lambda_H=0.01$ is added to the loss to encourage exploration.

Model Architecture The policy model's backbone is a Griffin model with the same configuration as the one for GP. The guess $\mathbf{g}_t$ is one-hot encoded over the C colours for each position, yielding a vector of dimension $C\times L$. The exact matches b_t and colour matches w_t are each one-hot encoded into vectors of dimension $L+1$. These are concatenated into a single vector of dimension $C\cdot L+2(L+1)$ and projected to the model dimension d via a linear layer.

The policy produces a value estimate and an action distribution. A linear layer maps the sequence model output at each step to a scalar value estimate $V(s_t)$. The action (a guess of L colour values) is generated autoregressively, one position at a time, using a GRU cell. At each position $k\in\{1,...,L\}$:

1. The GRU cell updates its hidden state using the current input (the sequence model output for $k=1$, or the embedding of the previously selected colour for $k>1$).
2. A linear readout produces logits over the C colours for position k.
3. A colour is sampled from Categorical(softmax(logits$_k/\tau$)), where τ is the temperature (set to 1.0 during training).
4. The selected colour is embedded and fed as input to the GRU for the next position.

The log-probability of the full action is the sum of the per-position log-probabilities: $\log\pi(a_t \mid s_t) = \sum_{k=1}^{L} \log\pi_k(a_t^{(k)} \mid s_t, a_t^{(1:k-1)})$.

E.4 Alchemy

The Alchemy environment is inspired by the latent-chemistry paradigm of Wang et al. (2021). At the start of each episode, an environment is sampled with a hidden chemistry: a set of deterministic transition rules governing how potions transform stones. Stones are described by $D=4$ discrete dimensions with value counts $(3,3,3,4)$, yielding $3\times3\times3\times4=108$ distinct stone types. Potions are described by a single discrete dimension with $P=6$ values. Each chemistry defines a fixed mapping from (initial_stone, potion) pairs to resulting stones; if a given (stone, potion) pair has no matching rule, the stone is returned unchanged and the transition is marked as invalid. The full set of chemistries and their transition rules are stored as a precomputed transition table of shape $[N_{\text{envs}}, N_{\text{rules}}, 2D+P]$.

At each timestep t, the agent chooses an action $a_t = (i_t, p_t)$—an initial stone and a potion—and the environment returns an observation (v_t, f_t) where $v_t \in \{0,1\}$ indicates whether the transition was valid (i.e., a matching rule exists in the current chemistry), and f_t is the resulting stone (equal to i_t when $v_t=0$).

The episode proceeds for a fixed trajectory length of $T=5$ steps. A random initial transition is sampled from the environment's transition table at the start of each episode and forms the first observation.

E.4.1 Bayesian Oracle World Model

Similar to Mastermind, the Bayesian oracle world model maintains the exact posterior over hidden chemistries by enumerating all N_{envs} candidate environments. Given a history of observations, the oracle computes a *consistency mask*: a candidate chemistry θ is consistent at step t if, for all prior observations $(i_{t'}, p_{t'}, v_{t'}, f_{t'})$ with $t'<t$:

1. If $v_{t'}=1$: chemistry θ contains a rule $(i_{t'}, p_{t'}) \mapsto f_{t'}$ that produces the observed output stone, and
2. If $v_{t'}=0$: chemistry θ has no rule matching the input pair $(i_{t'}, p_{t'})$.

The posterior probability of a new observation $s_t=(v_t, f_t)$ is computed as:

$$\rho(s_t \mid h_t = [h_{t-1}, s_{t-1}, (i_t, p_T)]) = \frac{\left|\{e \in \mathcal{E}_t : e \text{ is consistent with } (i_t, p_t, v_t, f_t)\}\right|}{|\mathcal{E}_t|}, \tag{21}$$

where $\mathcal{E}_t$ denotes the set of chemistries consistent with all observations prior to step t.

This world model provides exact Bayesian inference but scales linearly in the number of candidate environments, which must be enumerated exhaustively.

E.4.2 In-Context PFN World Model

The world model is a 12-layer causal Transformer with model dimension d=512, 8 attention heads, and feed-forward dimension 1024. Each observation $o_t=(i_t, p_t, f_t)$ is split into an *input token* (encoding i_t and p_t via concatenated one-hot vectors, projected to $\mathbb{R}^d$) and an *output token* (encoding f_t similarly). These are interleaved into a sequence $[\text{in}_1, \text{out}_1, \text{in}_2, \text{out}_2, \ldots]$ of length $2T$ and processed with causal attention, so that the hidden state at each input position in_t attends to all tokens up to and including out_{t-1}.

The output prediction head is an autoregressive GRU decoder that, starting from the Transformer's hidden state at in_t, sequentially predicts each component of the observation output: first a validity flag (2 categories), then each of the four stone dimensions (3, 3, 3, and 4 categories respectively). At each step the GRU receives the embedding of the previous ground-truth component (teacher forcing during training),

updates its hidden state, and produces logits over the current component's categories. Invalid logit positions (where the category index exceeds the component's vocabulary) are masked to $-\infty$ before the softmax.

The model is trained on sequences of $T_{\text{train}}=50$ transitions sampled with uniformly random actions from uniformly sampled environment IDs, with batch size $B=64$ and the Adam optimizer at learning rate 10^{-4}.

Data-Generating Process The PFN is trained on synthetically generated transition trajectories. For each training example, a chemistry (environment) is sampled uniformly at random, and then a sequence of $T_{\text{train}}=20$ random transitions is generated. Each transition draws a random initial stone and potion uniformly from their respective value ranges, and the corresponding observation $(v_t, \mathbf{s}_t^{\text{out}})$ is computed by the environment.

Model Architecture The in-context world model is a causal Transformer decoder that processes interleaved input (action) and output (observation) tokens. Each input token encodes the initial stone and potion via per-dimension one-hot vectors. The stone dimensions $(3,3,3,4)$ and potion dimension (6) are each one-hot encoded and concatenated, yielding a vector of dimension of 19, which is projected to the model dimension d via a linear layer. Each output token encodes the validity indicator v_t and the final stone. The validity is one-hot encoded into a vector of dimension 2, and is concatenated with the final stone's one-hot into a 15-dimensional vector, which is projected to d via a separate linear layer.

For a trajectory of T steps, the input sequence is constructed by interleaving input and output embeddings. For each timestep t, the input embedding and output embedding are placed at positions $2t$ and $2t+1$, resulting in a sequence of length $2T$. The sequence is processed by a causal Transformer decoder with model dimension $d=512$, MLP hidden dimension $F=1024$, $N=12$ layers, $H=8$ attention heads with head dimension $d/H=64$.

Similar to Mastermind's policy, the output prediction uses a GRU-based autoregressive decoder that conditions each output component's prediction on the previously observed ground-truth components (teacher forcing during training). The per-step loss is the sum of the per-component negative log-probabilities: $-\sum_{k=1}^{5}\log p(o_t^{(k)} \mid o_t^{(1:k-1)}, h_t)$. The in-context world model is trained to minimize the cross-entropy loss of predicting the output at each step. In each training step, a fresh batch of $B=64$ random trajectories is generated on-the-fly (no fixed dataset), and the model is optimized using Adam with a learning rate of 10^{-4}.

E.4.3 Policy

The policy is trained using PPO (Proximal Policy Optimization). At each training step, a batch of $B=32$ trajectories of length $T=5$ is collected by rolling out the current policy in the environment. The PPO objective uses $K=4$ update epochs per batch of collected data.

Model Architecture The policy is the same sequence model as used in Mastermind, with a Griffin backbone model (De et al., 2024), and a GRU to predict the final output components autoregressively conditioned on the Griffin's output. The log-probability of the full action is the sum of the per-component log-probabilities: $\log\pi(a_t \mid s_t)=\sum_{k=1}^{5}\log\pi_k(a_t^{(k)} \mid s_t, a_t^{(1:k-1)})$.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 clearly outline our claims in both abstract and introduction, and the theoretical/experimental results support them.

   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: Our discussion in Section 6 clearly outlines the main challenges and limitations of our approach, highlighting where future work can improve on it.

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: Proofs for all theorems and theoretical results are laid out in Appendix C, and assumptions are laid out in Appendix B. Both are referenced in the main text.

   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: Our full experimental settings are described in Appendix E, in enough detail where a practitioner should be able to reproduce our results.

   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: [No]

   Justification: We do not release code at this time, but hope to in the coming months.

   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: All are clearly described in Appendix E.

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: Several of our experiments randomize across seeds, and in these cases we report the number of seeds, the mean of the metric, and the standard error of the mean of the metric.

   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: [No]

   Justification: We do not provide this because our experiments are all fairly lightweight, should run on a single GPU without problem, and jobs should complete within at most a few days (with most completing far more quickly).

   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: We read and have confirmed with the guidelines.

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: [No]

Justification: We study small-scale settings, and there are substantial practical limitations to scaling our methods that must be overcome before any risk of societal danger becomes present. We therefore believe that speculating on the risks of our work would be premature.

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: [No]

Justification: Our models and datasets are both small, our models are trained from scratch, and all experiments are based off of synthetic toy environments.

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: [N/A]

    Justification: We do not use existing assets.

    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: [N/A]

    Justification: We do not release new assets.

    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: Our paper does not involve crowdsourcing nor research with human subjects.

    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: Our paper does not involve crowdsourcing nor research with human subjects.

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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

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.