Curiosity-Driven Exploration with Information Bottleneck Representations and Matrix-Based Mutual Information

Zhaoxu Meng and Yong Cui *

Posted Date: 10 April 2026

doi: 10.20944/preprints202604.0739.v1

Keywords: curiosity; intrinsic motivation; information bottleneck; mutual information; Renyi entropy; reinforcement learning; kernel density estimation

 doi:10.20944/preprints202604.0739.v1

Article

Curiosity-Driven Exploration with Information Bottleneck Representations and Matrix-Based Mutual Information

Zhaoxu Meng [1,†] and Yong Cui [2,*]

[1] Department of Electrical and Computer Engineering, The University of Hong Kong, Pokfulam, Hong Kong, China
[2] School of Automation Science and Electrical Engineering, Beihang University, No. 37 Xueyuan Road, Haidian District, Beijing 100191, China
* Correspondence: cuiyong@buaa.edu.cn
† This work was done when the first author was at the Beihang University.

Abstract

Exploration remains a central challenge in reinforcement learning, especially in sparse-reward settings where extrinsic feedback alone is often insufficient to guide effective behavior. In this work, we develop a curiosity-driven framework that combines a hybrid intrinsic reward with compact predictive representation learning. Specifically, curiosity is quantified by integrating prediction error with the rarity of state-action pairs in a learned latent space. To make novelty estimation more meaningful for high-dimensional observations such as raw pixels, we employ the Information Bottleneck principle to learn low-dimensional representations that suppress irrelevant variability while preserving predictive structure of the environment dynamics. We further investigate two practical ways to optimize predictive information: one based on entropy decomposition and the other based on matrix-based Rényi entropy. Experiments on Acrobot show that the proposed method substantially improves exploration efficiency over ICM, RND, and a k-NN novelty baseline. On MountainCar, however, the improvement is less evident, suggesting that the proposed framework is particularly beneficial in environments with high-dimensional observations or more structured dynamics.

Keywords: curiosity; intrinsic motivation; information bottleneck; mutual information; Renyi entropy; reinforcement learning; kernel density estimation

1. Introduction

Reinforcement learning (RL) is typically driven by extrinsic rewards supplied by the environment, such as game scores, goal distances, or other task-dependent feedback signals [1]. Although such rewards have supported many impressive achievements—for instance, Deep Q-Networks reaching human-level performance on a wide range of Atari 2600 games [2]—they are often insufficient in sparse-reward settings, where informative feedback arrives only rarely or after long action sequences [3–5]. Under these conditions, exploration becomes a major challenge: the agent may spend many interactions receiving almost no useful learning signal and fail to discover behaviors that lead to progress [3,6]. In addition, policies optimized purely for extrinsic return can become overly specialized to the training environment and transfer poorly to new situations or goals [7–9].

Humans and animals, in contrast, often explore even when no explicit external reward is provided [10,11]. This intrinsic drive to seek novelty, reduce uncertainty, and acquire new knowledge is commonly described as curiosity. Humans do not need rewards such as money or food to investigate unfamiliar things; in many situations, the act of trying, observing, and learning is rewarding in itself.

Developmental studies provide a clear illustration of this phenomenon. For example, infants tend to engage more strongly with novel or uncertain stimuli, and such curiosity-driven behavior is closely related to improved learning and exploratory information sampling [12,13]. These observations

 doi:10.20944/preprints202604.0739.v1

have inspired a substantial body of work in reinforcement learning, where agents are endowed with intrinsic reward signals to promote exploration beyond what is directly encouraged by the task reward alone [4,14,15].

In reinforcement learning, one way to enhance exploration is to endow the agent with an internal drive to interact with its environment even when external rewards provide little guidance. To do so, we need an intrinsic signal that can assess, after each transition, whether the agent is still encountering something worth exploring. In this work, we argue that curiosity should be high in two situations: first, when the agent reaches regions of the environment that have rarely been visited; and second, when it is in an otherwise familiar situation but takes an action that has only seldom been attempted there. Motivated by this view, we define intrinsic reward through the density of state–action pairs rather than states alone.

A direct implementation of this idea in the original observation space is, however, problematic. When states are represented by raw pixels, the space of possible observations is extremely large, and most pixel configurations do not correspond to valid or meaningful environment states [16]. As a result, density estimates computed directly from observations can be unreliable and difficult to interpret. To overcome this issue, we first map observations into a compact latent space using the **Information Bottleneck** (IB) principle. The goal is to learn a representation that discards irrelevant variation while preserving the structure needed to describe meaningful environment states and their dynamics.

Following the IB principle [17], we learn a compact latent state $s_t = E_\phi(o_t)$ from the observation o_t. The representation is encouraged to remain concise while retaining enough information to support prediction of future transitions. We adopt the predictive IB objective

$$\mathcal{L}_{\mathrm{IB}} = \mathrm{I}(o_t; s_t) - \beta\, \mathrm{I}(s_{t+1}; s_t, a_t), \tag{1}$$

where $\mathrm{I}(o_t; s_t)$ measures how much information from the observation is preserved in the latent representation, and $\mathrm{I}(s_{t+1}; s_t, a_t)$ encourages the pair (s_t, a_t) to remain informative about the next latent state. For instance, if $o_t \in \mathbb{R}^{256}$ corresponds to a 16×16 image, the encoder can map it to a latent variable $s_t \in \mathbb{R}^d$ with $d \ll 256$, yielding a lower-dimensional representation that is easier to model and more suitable for novelty estimation.

To optimize the predictive-information term, we first consider the standard entropy decomposition

$$\mathrm{I}(s_{t+1}; s_t, a_t) = H(s_{t+1}) - H(s_{t+1} \mid s_t, a_t). \tag{2}$$

Under this view, increasing the marginal entropy $H(s_{t+1})$ encourages the agent to spread its visitation over a broader range of latent states, while reducing the conditional entropy $H(s_{t+1} \mid s_t, a_t)$ promotes a more predictable latent dynamics model.

In addition to this decomposition-based formulation, we also study matrix-based entropy functionals [18,19] derived from Rényi entropy. These functionals provide an alternative way to quantify entropy and mutual information directly from kernel matrices, avoiding explicit density modeling over latent variables.

Our framework builds on the representational-space novelty idea of [20], while introducing several extensions. Specifically, we use the Information Bottleneck principle to shape a compact latent space for curiosity estimation, we examine matrix-based entropy functionals as an alternative tool for measuring predictive information [18,21], and we replace KNN-based novelty estimation with KDE in order to obtain a smoother density-based intrinsic reward in the learned representation space.

The main contributions of this work are summarized as follows:

- We propose a curiosity-driven reinforcement learning framework in which intrinsic reward is defined on state–action representations, capturing both rarely visited regions and infrequently attempted actions in familiar situations.
- We employ the Information Bottleneck principle to learn compact latent states that are suitable for both transition prediction and density-based novelty estimation.

 doi:10.20944/preprints202604.0739.v1

- We investigate two practical formulations for encouraging predictive information in the latent space: an entropy-decomposition-based objective and a matrix-based Rényi-entropy objective.
- We replace KNN-based novelty estimation with kernel density estimation in the learned representation space, leading to a smoother intrinsic-reward signal.
- We empirically evaluate the proposed framework on benchmark environments and show that its benefit is most evident when representation quality plays an important role in exploration.

2. Relevant Work

To develop an effective curiosity-driven reinforcement learning system, two ingredients are especially important. One is the intrinsic-reward mechanism, which determines what kinds of experiences are treated as novel, informative, or worth revisiting. The other is the representation space in which such judgments are made, since poor state representations can make even well-designed curiosity signals noisy or misleading. Because prior work has advanced these two directions in different ways, we review them separately.

2.1. Design of Intrinsic Rewards

A large portion of the curiosity-driven RL literature defines intrinsic reward through prediction-based signals. In this line of work, transitions that are difficult to predict are regarded as more informative and therefore receive higher intrinsic reward. Representative examples include ICM [4] and RND [14], both of which use prediction error as the exploration signal. ICM measures the discrepancy between a predicted next state and the true next state, whereas RND compares a learned predictor against the output of a fixed random target network.

Beyond a single prediction model, other methods quantify curiosity through predictive uncertainty. Disagreement [22], for instance, rewards transitions that induce large variation across multiple forward-model predictions. Related uncertainty-estimation techniques, such as MC-Dropout [23] and deep ensembles [24], have also been used more broadly to characterize model uncertainty, including in recent studies of large language models. From the perspective of exploration, such methods are appealing because they reward interactions for which the agent's current model remains uncertain.

Another family of approaches focuses not on raw surprise, but on *learning progress*. The underlying idea is that curiosity should be strongest in regions where the agent is still improving, rather than in regions that are either fully understood or inherently unpredictable. This intuition has been instantiated in several ways, including changes in transition dynamics [25], temporal variation in prediction error [10], and Bayesian model updates as in VIME, where intrinsic reward is tied to the KL divergence between posterior and prior dynamics parameters [26]. Closely related ideas also appear in diversity-driven exploration, where agents are rewarded for inducing substantial shifts in their policy behavior over time [27].

Some methods instead emphasize explicit uncertainty reduction. These approaches favor state–action pairs that are expected to yield informative feedback, for example by targeting regions with high epistemic uncertainty [28] or by rewarding actions that improve an auxiliary estimate of confidence or reliability. Such formulations are closely connected to the view of curiosity as targeted information acquisition rather than mere novelty seeking.

More recent work has also moved novelty estimation into learned or auxiliary embedding spaces. RE3 [29] measures rarity under a fixed random representation, which helps reduce the non-stationarity that can arise when the embedding itself changes during training. ProtoRL [30], in contrast, learns prototypes and encourages exploration of states that are far from existing prototype centers, thereby favoring underrepresented regions of the learned space. These methods highlight the growing recognition that the usefulness of an intrinsic reward depends strongly on the representation in which novelty is evaluated.

At a broader level, intrinsic motivation in RL has also been organized from an information-theoretic perspective. The survey in [6], for example, distinguishes among several related concepts, including surprise, novelty, and skill learning, and argues that successful intrinsic-reward mechanisms

 doi:10.20944/preprints202604.0739.v1

typically require a balance between seeking unfamiliar experiences and focusing on interactions that remain relevant for future learning.

2.2. Learning Representations

Representation learning plays a central role in modern reinforcement learning, especially when agents operate on high-dimensional observations such as images, videos, or sensor streams. In these settings, the raw input often contains large amounts of redundant or task-irrelevant variation, making it difficult to directly estimate novelty, predict transitions, or learn robust control policies. A learned latent state can therefore serve as an intermediate representation that is easier to model and better aligned with the structure of the environment.

A common goal is to map observations into a lower-dimensional latent space that preserves information needed for decision-making while suppressing nuisance factors. From a technical perspective, such representations are useful only if they support downstream objectives relevant to reinforcement learning, including dynamics prediction, reward estimation, and exploration. This is particularly important in curiosity-driven settings, where the quality of the intrinsic reward often depends directly on the geometry of the latent space in which novelty or uncertainty is measured.

One principled approach to this problem is provided by the information bottleneck (IB) principle [17]. In our setting, an encoder produces a latent state $s_t = E_\phi(o_t)$ from the observation o_t, and the representation is encouraged to remain compact while preserving predictive information about future transitions. A predictive IB objective can be written as

$$\mathcal{L}_{\mathrm{IB}} = \mathrm{I}(o_t; s_t) - \beta\, \mathrm{I}(s_{t+1}; s_t, a_t), \tag{3}$$

where the first term penalizes excessive dependence of the latent code on the raw observation, and the second term promotes representations for which the pair (s_t, a_t) remains informative about the next latent state s_{t+1}. In practice, this objective encourages a representation that is neither overly detailed nor overly compressed, but sufficiently structured for predictive modeling in the latent dynamics.

IB-style objectives have been explored in several reinforcement-learning-related settings. Early work showed that bottleneck regularization can improve sample efficiency and learning stability by discouraging latent representations from memorizing irrelevant observation details [31]. Subsequent studies extended this idea to behavior cloning for robot manipulation [32], multimodal reinforcement learning in which multiple sensory streams are compressed into a predictive joint representation [33], discrete bottleneck models that encourage structured abstractions [34], and goal-conditioned learning, where compressed goal representations can improve transfer and generalization [35]. Taken together, these works suggest that bottlenecked representations are particularly valuable when the agent must balance abstraction with predictive sufficiency.

Beyond IB, a related line of work uses contrastive or predictive objectives to shape latent states. Contrastive learning methods train encoders to bring temporally or semantically related observations closer together while pushing unrelated samples apart, thereby improving representation quality without relying solely on reconstruction losses. In reinforcement learning, such objectives have been used to enhance data efficiency and improve the structure of pixel-based latent spaces. For example, Curled–Dreamer augments DreamerV3 with a contrastive objective and reports improved performance on DeepMind Control Suite benchmarks [36]. More broadly, world-model and predictive-state approaches also learn latent spaces by requiring them to support rollout prediction or planning, even when not explicitly formulated through an information bottleneck.

Overall, recent work increasingly treats representation learning and intrinsic motivation as tightly coupled rather than separate design choices. Intrinsic rewards are often computed in latent space, and, conversely, exploration objectives can shape the latent space toward representations that emphasize novelty, uncertainty, or controllable change. This interaction is especially important in settings where curiosity depends not only on what the agent observes, but also on how those observations are encoded before novelty or predictive information is measured.

 doi:10.20944/preprints202604.0739.v1

2.3. Psychological Perspectives on Curiosity

Research in psychology and cognitive science provides useful intuition for the design of intrinsic motivation in reinforcement learning. Human curiosity is often described as a drive to acquire information even in the absence of immediate external reward. A particularly influential view is Loewenstein's information-gap theory, which relates curiosity to the discrepancy between what one currently knows and what one wants to know [37]. Under this perspective, curiosity is not triggered by novelty alone, but by novelty that reveals a learnable gap in understanding. Partial knowledge can therefore increase curiosity by exposing missing pieces, whereas excessive uncertainty may fail to produce focused exploration.

This perspective is closely related to a practical challenge in curiosity-driven RL. Many intrinsic-reward methods assign larger reward to larger prediction error or greater uncertainty, implicitly assuming that unpredictability is always useful. Human curiosity, however, appears to be more selective: it tends to be strongest when new information is neither trivial nor completely unstructured. This difference helps explain why purely prediction-error-driven exploration can become unstable in highly stochastic settings, including the well-known "noisy TV" failure mode [38]. It also motivates methods that balance surprise with structure, so that exploration is directed toward transitions that are not only unusual but also informative for future learning.

Psychological studies further suggest that curiosity may take different forms, ranging from broad novelty-seeking to more targeted information-seeking behavior [39,40]. In reinforcement learning, most existing curiosity mechanisms are still closer to the former: they reward novelty, uncertainty, or diversity to improve coverage of the environment [27,41]. By contrast, more directed forms of curiosity remain relatively underexplored. This observation motivates our focus on *state–action* novelty in a learned latent space. Rather than rewarding unpredictability in the raw observation space, we seek an intrinsic signal that favors underexplored but meaningful transitions, where both the representation and the action choice matter for determining what the agent can still learn about the environment.

3. Methods

3.1. Problem Setup and Notation

We consider an agent interacting with an environment modeled as a partially observed Markov decision process. At time step t, the agent receives an observation $o_t \in \mathcal{O}$ and selects an action $a_t \in \mathcal{A}$. The environment transitions to the next observation o_{t+1} and emits an extrinsic reward r_t^{ext}. We denote a terminal indicator by $d_t \in \{0, 1\}$.

Because novelty estimation is highly sensitive to representation quality, we learn a compact latent state $s_t \in \mathbb{R}^d$ through an encoder E_ϕ:

$$s_t = E_\phi(o_t), \qquad d \ll \dim(o_t). \tag{4}$$

For discrete actions we use one-hot encoding $\mathbf{a}_t \in \{0, 1\}^{|\mathcal{A}|}$; for continuous actions we concatenate the raw action vector. We frequently use the state-action embedding

$$y_t = [\, s_t,\, \mathbf{a}_t \,] \in \mathbb{R}^{d+|\mathcal{A}|}, \tag{5}$$

so that taking a novel action in a familiar state is still considered novel.

We store transitions in a replay buffer $\mathcal{B} = \{(o_t, a_t, r_t, d_t, o_{t+1})\}$ and sample mini-batches for optimization (using uniform or prioritized replay, depending on implementation).

Notation. Uppercase letters (e.g., S_t) denote random variables, while lowercase letters (e.g., s_t) denote their realizations. In the implementation, $s_t = E_\phi(o_t)$ is the learned latent state. For clarity, we use the same (s_t, a_t, s_{t+1}) notation throughout the mutual-information terms.

 doi:10.20944/preprints202604.0739.v1

3.2. Overall Framework

Rather than relying solely on extrinsic rewards, our framework in Figure 1 augments exploration with an intrinsic signal that evaluates how informative a transition is for the agent. We focus on two complementary aspects of curiosity. The first is *model uncertainty*: transitions that are difficult to predict may indicate parts of the environment that are not yet well understood. However, prediction-error-based curiosity alone can be misleading, because large errors may persist even in transitions that are uninformative or inherently stochastic, as illustrated by the well-known "noisy TV" failure mode [38,42].

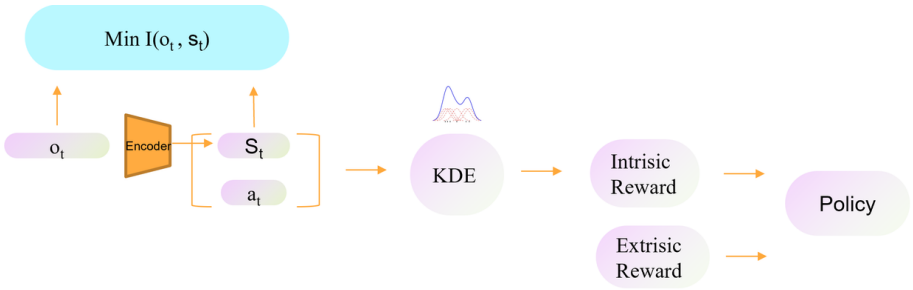

Figure 1. Overview of our curiosity-driven RL pipeline. At each time step t, the raw observation o_t is mapped by an encoder to a compact latent state s_t. The encoder is trained under an information bottleneck (IB) objective that compresses observations by minimizing $\mathrm{I}(o_t; s_t)$ while preserving predictive structure by maximizing $\mathrm{I}(s_{t+1}; s_t, a_t)$. To quantify curiosity, we form the state–action representation (s_t, a_t) and estimate its density ρ_t with KDE over the replay buffer, while a forward dynamics model provides a prediction-error bonus. The intrinsic reward is then computed as a hybrid signal, $r_t^{\mathrm{int}} = (1-\alpha) r_t^{\mathrm{kde}} + \alpha r_t^{\mathrm{pe}}$, where $r_t^{\mathrm{kde}} = -\log(\rho_t/\rho_0)$ and ρ_0 denotes the density under a random-policy baseline.

The second aspect is *rarity*: transitions occurring in rarely visited regions should receive higher intrinsic reward than those that have been repeatedly experienced [43]. This view naturally prevents curiosity from remaining high in familiar parts of the environment. At the same time, density-based novelty on its own may be too myopic, since a region can become familiar before the underlying transition structure has been modeled sufficiently well. A useful intrinsic signal should therefore account for both how rare a transition is and how much the agent still has to learn about it.

Based on this observation, we define curiosity in a learned latent space and combine two ingredients in a unified framework. First, we compute a density-based intrinsic reward on *state-action* representations, so that the agent is encouraged not only to visit unfamiliar states, but also to try underexplored actions in otherwise familiar situations. Second, we learn the latent state with representation objectives motivated by the IB principle, so that the embedding remains compact while preserving predictive structure in the environment dynamics. This design allows the intrinsic reward to be evaluated in a representation space that is more suitable for novelty estimation than the raw observation space.

3.3. Intrinsic Reward from State–Action Novelty

3.3.1. Kernel Density in Representation Space

Given the embedded state–action $y_t = [s_t, \mathbf{a}_t]$, we estimate its replay-buffer density using kernel density estimation (KDE):

$$\rho(y_t) = \frac{1}{N} \sum_{i=1}^{N} \exp\Big(- \frac{\|y_t - y_i\|_2^2}{2h^2} \Big), \tag{6}$$

where $\{y_i\}_{i=1}^{N}$ are state–action embeddings stored in the replay buffer and h is the kernel bandwidth. Compared with hard k-NN neighbor sets, KDE yields a smoother intrinsic-reward landscape, which reduces reward variance when the latent representation evolves during training.

 doi:10.20944/preprints202604.0739.v1

3.3.2. Intrinsic Reward with a Random-Policy Baseline

We adopt the log-density ratio form of intrinsic reward:

$$r_t^{\text{kde}} = -\log\left(\frac{\rho(y_t)}{\rho_0(y_t)}\right), \tag{7}$$

where ρ_0 is a baseline density estimated from samples collected during an early exploration phase. Intuitively, r_t^{kde} is large when y_t lies in low-density regions compared with what a random agent would typically encounter. This reduces sensitivity to trivial novelty and provides a stable reference scale across training.

3.3.3. Prediction-Error Loss and Hybrid Intrinsic Signal

Besides density-based novelty, we also use *prediction error* as an intrinsic signal. Given the learned forward dynamics model T_ψ, we predict the next latent state

$$\hat{s}_{t+1} = T_\psi([\, s_t, \mathbf{a}_t \,]), \tag{8}$$

and define the one-step prediction-error loss as

$$\mathcal{L}_{\text{PE}} = \mathbb{E}\Big[\, \big\|\hat{s}_{t+1} - s_{t+1}\big\|_2^2 \Big], \tag{9}$$

where $\mathcal{L}_{\text{PE}}$ is a special case of the n-step transition loss in Eq. (19) when $n = 1$. Intuitively, large prediction error indicates that the transition $(s_t, a_t) \to s_{t+1}$ is not yet well modeled, and is therefore informative to explore.

To turn prediction error into a numerically stable intrinsic bonus, we use a normalized form

$$r_t^{\text{pe}} = \log\left(1 + \frac{\big\|\hat{s}_{t+1} - s_{t+1}\big\|_2^2}{\bar{e} + \epsilon}\right), \tag{10}$$

where $\bar{e}$ is a running mean of the prediction error and ϵ is a small constant.

Finally, we *hybridize* KDE-based novelty (Eq. (7)) with prediction-error bonus:

$$r_t^{\text{int}} = (1-\alpha)\, r_t^{\text{kde}} + \alpha\, r_t^{\text{pe}}, \qquad \alpha \in [0,1], \tag{11}$$

where $r_t^{\text{kde}} \triangleq -\log\big(\rho(y_t)/\rho_0(y_t)\big)$. This combination leverages complementary strengths: prediction error encourages exploring model-uncertain transitions, while KDE ensures the intrinsic drive naturally decays in frequently visited regions, mitigating pathological rewards in highly stochastic observations.

3.4. Representation Learning via Predictive Information Bottleneck

3.4.1. IB Objective

In our framework, the latent state is not used merely as a compressed encoding of the observation. It must also support two downstream operations that are central to curiosity estimation: predicting future transitions and providing a stable space for measuring state–action novelty. We therefore adopt a predictive Information Bottleneck (IB) objective that encourages the representation to remain compact while retaining the information needed for modeling latent dynamics:

$$\mathcal{L}_{\text{IB}} = \mathrm{I}(o_t; s_t) - \beta\, \mathrm{I}(s_{t+1}; s_t, a_t), \tag{12}$$

where $\mathrm{I}(o_t; s_t)$ penalizes unnecessary dependence of the latent code on the raw observation, and $\mathrm{I}(s_{t+1}; s_t, a_t)$ encourages the pair (s_t, a_t) to remain informative about the next latent state. The coefficient β controls the trade-off between compression and predictive structure.

 doi:10.20944/preprints202604.0739.v1

For the compression term, our goal is not to preserve all observation details, but to retain only the information needed to represent meaningful environment states in a low-dimensional latent space. In practice, this is encouraged by constraining the dimensionality of s_t and optimizing the predictive objective jointly, so that the learned representation remains concise without becoming uninformative.

For the predictive term $\mathrm{I}(s_{t+1}; s_t, a_t)$, we consider two equivalent decompositions (see Figure 2). The first expresses predictive mutual information in terms of the next-state entropy and the conditional uncertainty of the latent dynamics:

$$\mathrm{I}(s_{t+1}; s_t, a_t) = H(s_{t+1}) - H(s_{t+1} \mid s_t, a_t), \tag{13}$$

which emphasizes that a useful latent representation should both cover a diverse range of future states and make those future states predictable from the current state–action pair.

Equivalently, the same quantity can be written as

$$\mathrm{I}(s_{t+1}; s_t, a_t) = H(s_t, a_t) - H(s_t, a_t \mid s_{t+1}), \tag{14}$$

which provides a backward-looking interpretation in terms of how informative the next latent state is about its predecessor state–action pair. In the following subsections, we use these decompositions to derive practical objectives for learning the latent representation.

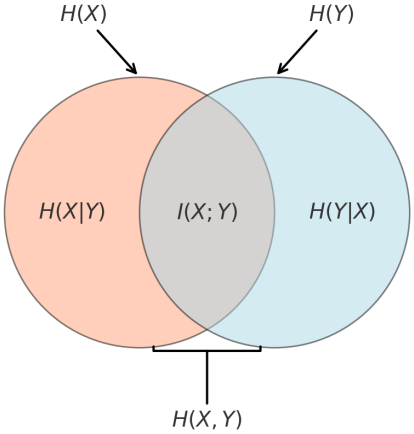

Figure 2. Mutual information decomposition. The two circles represent marginal entropies $H(X)$ and $H(Y)$; their overlap is $\mathrm{I}(X;Y)$, and the non-overlapping parts are conditional entropies. In our setting, $Y \equiv s_{t+1}$ and $X \equiv (s_t, a_t)$, yielding $\mathrm{I}(s_{t+1}; s_t, a_t) = H(s_{t+1}) - H(s_{t+1} \mid s_t, a_t) = H(s_t, a_t) - H(s_t, a_t \mid s_{t+1})$.

3.4.2. First Decomposition Method: Using Prediction Error to Approximate Predictive Information

For the decomposition

$$\mathrm{I}(s_{t+1}; s_t, a_t) = H(s_{t+1}) - H(s_{t+1} \mid s_t, a_t), \tag{15}$$

we optimize the two entropy terms through practical surrogate objectives. Intuitively, the first term encourages the latent dynamics to cover a diverse set of future states, while the second term requires those future states to remain predictable from the current state–action pair.

Encouraging large $H(s_{t+1})$ via latent dispersion.

To prevent the latent representation from collapsing to a narrow region of the space, we introduce a dispersion regularizer based on pairwise distances within a mini-batch. Given a batch $\{s_i\}_{i=1}^{B}$, we define

$$\mathcal{L}_{\text{unif}} = \log\left(\frac{2}{B(B-1)} \sum_{1 \le i < j \le B} e^{-t\|s_i - s_j\|_2^2}\right), \tag{16}$$

 doi:10.20944/preprints202604.0739.v1

where $t > 0$ is a temperature (equivalently, bandwidth) hyperparameter controlling the sensitivity to pairwise distances. Minimizing $\mathcal{L}_{\text{unif}}$ encourages samples in the latent space to be more spread out on average, which serves as a practical surrogate for increasing the marginal entropy of future latent states.

From another perspective, using only transition-prediction losses may admit degenerate solutions in which different observations are mapped to overly similar latent codes. The dispersion term counteracts this effect by encouraging the latent representation to occupy a broader region of the space, thereby improving both expressiveness and the quality of subsequent novelty estimation.

Reducing $H(s_{t+1} \mid s_t, a_t)$ via forward transition prediction.

To make the next latent state predictable from the current state and action, we learn a transition model T_ψ operating directly in latent space (see Figure 3):

$$\hat{s}_{t+1} = T_\psi([s_t, \mathbf{a}_t]). \tag{17}$$

A smaller prediction error corresponds to lower conditional uncertainty about the future latent state given (s_t, a_t), and therefore acts as a practical surrogate for minimizing $H(s_{t+1} \mid s_t, a_t)$.

To improve consistency over multiple steps and to reduce the effect of short-horizon fitting, we train T_ψ using an n-step rollout objective. Starting from the encoded state $s_t = E_\phi(o_t)$, we recursively predict future latent states according to

$$\hat{s}_{t+i+1} = T_\psi([\hat{s}_{t+i}, \mathbf{a}_{t+i}]), \qquad i = 0, \ldots, n-1. \tag{18}$$

Let $s_{t+i} = E_\phi(o_{t+i})$ denote the encoded target latents. We then minimize the terminal-masked n-step transition loss

$$\mathcal{L}^{(n)}_{\text{trans}} = \frac{1}{n} \sum_{i=0}^{n-1} (1 - d_{t+i}) \, \|\hat{s}_{t+i+1} - s_{t+i+1}\|_2^2, \tag{19}$$

where the terminal mask prevents the predictor from propagating across episode boundaries.

Optionally, we also consider a normalized variant that rescales each step loss by the magnitude of the predicted transition:

$$\tilde{\mathcal{L}}^{(n)}_{\text{trans}} = \frac{1}{n} \sum_{i=0}^{n-1} (1 - d_{t+i}) \, \frac{\|\hat{s}_{t+i+1} - s_{t+i+1}\|_2^2}{\|\hat{s}_{t+i+1} - \hat{s}_{t+i}\|_2 + \epsilon}. \tag{20}$$

This normalization reduces sensitivity to scale differences across rollout steps and can improve training stability when latent transitions vary substantially in magnitude.

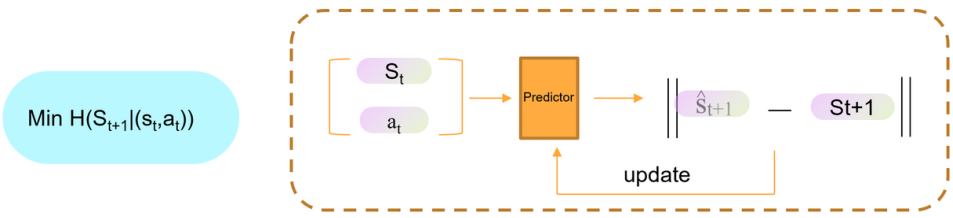

Figure 3. Minimizing the conditional entropy term $H(s_{t+1} \mid s_t, a_t)$. A predictor f_ψ takes (s_t, a_t) and outputs $\hat{s}_{t+1}$. We minimize $\|\hat{s}_{t+1} - s_{t+1}\|_2^2$ as a surrogate for the conditional entropy.

3.5. Second Decomposition Method: Backward Transition Predictor

Besides the forward decomposition, the predictive mutual information can also be written as

$$\mathrm{I}(s_{t+1}; s_t, a_t) = H(s_t, a_t) - H(s_t, a_t \mid s_{t+1}). \tag{21}$$

 doi:10.20944/preprints202604.0739.v1

This form admits a backward-looking interpretation: instead of asking how well the current state–action pair predicts the future, we ask how informative the next latent state is about its predecessor state–action pair. In this view, maximizing predictive mutual information can also be approached by reducing uncertainty over plausible predecessors.

To operationalize this idea, we introduce a stochastic backward predictor B_ξ that maps the next latent state s_{t+1} and a noise variable ϵ to candidate predecessor state–action embeddings:

$$\hat{y}_t^{(k)} = B_\xi(s_{t+1}, \epsilon^{(k)}), \qquad \epsilon^{(k)} \sim \mathcal{N}(0, I), \qquad k = 1, \dots, K. \tag{22}$$

These K samples induce an empirical conditional distribution over predecessors. We evaluate the ground-truth predecessor y_t under a kernel density estimate formed from the sampled candidates:

$$\hat{q}_\xi(y_t \mid s_{t+1}) = \frac{1}{K} \sum_{k=1}^{K} \exp\left(- \frac{\left\| y_t - \hat{y}_t^{(k)} \right\|_2^2}{2\sigma_b^2} \right), \tag{23}$$

where σ_b controls the smoothness of the induced distribution in the joint state–action space. The corresponding backward loss is defined as

$$\mathcal{L}_{\text{back}} = \mathbb{E}\Big[- \log \hat{q}_\xi(y_t \mid s_{t+1}) \Big]. \tag{24}$$

Minimizing Eq. (24) encourages the model to assign high probability mass to the true predecessor, thereby reducing $H(y_t \mid s_{t+1})$ without separating state and action into distinct inverse-dynamics terms. Because the predictor is stochastic, it can also represent multi-modal predecessor hypotheses, which is useful in settings where inverse dynamics are many-to-one.

For consistency with the forward n-step rollout objective, we extend this loss along trajectory segments:

$$\mathcal{L}_{\text{back}}^{(n)} = \frac{1}{n} \sum_{i=0}^{n-1} (1 - d_{t+i}) \Big[- \log \hat{q}_\xi(y_{t+i} \mid s_{t+i+1}) \Big], \tag{25}$$

where $\hat{q}_\xi(y_{t+i} \mid s_{t+i+1})$ is computed by sampling from $B_\xi(s_{t+i+1}, \epsilon)$ as in Eq. (22) and applying Eq. (23). The terminal mask $(1 - d_{t+i})$ prevents the loss from propagating across episode boundaries.

We include this backward formulation mainly as a conceptual alternative to the forward predictive objective. In preliminary experiments, activating a separate backward predictor did not improve exploration efficiency, and in some cases interfered with learning accurate forward dynamics. For this reason, the backward loss is not used in the final implementation reported in the main results. Nevertheless, we retain it here because it provides an alternative decomposition of predictive mutual information and suggests a possible direction for future work, especially if forward and backward dynamics can be learned through a shared architecture rather than separate predictors.

3.6. Matrix-Based Entropy Estimation as an Alternative to Prediction-Error Surrogates

As an alternative to predictor-based surrogates, we also consider a matrix-based estimator of predictive mutual information. Unlike explicit density estimators, this approach computes entropy directly from kernel similarity matrices constructed from samples [18,44]. In our setting, it provides a non-parametric way to quantify the dependence between the next latent state s_{t+1} and the current state–action pair (s_t, a_t), and can therefore be used as a direct regularizer for representation learning.

Given n samples $\{x_i\}_{i=1}^n$, we first construct a Gram matrix $K \in \mathbb{R}^{n \times n}$ with entries

$$K_{ij} = \kappa(x_i, x_j), \tag{26}$$

 doi:10.20944/preprints202604.0739.v1

where $\kappa(\cdot,\cdot)$ is a positive-definite kernel function. In practice, we use a Gaussian kernel, which also satisfies the infinite-divisibility condition commonly assumed in the matrix-based entropy literature [18,45,46]. The Gram matrix is then normalized to have unit trace:

$$A = \frac{K}{\mathrm{tr}(K)}. \tag{27}$$

The eigenvalues of A are nonnegative and sum to one, which allows entropy to be defined through the spectrum of the normalized matrix. The matrix-based Rényi entropy of order α is

$$H_\alpha(A) = \frac{1}{1-\alpha}\log\big(\mathrm{tr}(A^\alpha)\big), \tag{28}$$

where A^α denotes the matrix power defined through spectral decomposition [18].

To measure dependence between two variables, we construct normalized Gram matrices for each of them. Suppose A corresponds to samples from one variable and B to samples from another. Their joint entropy is defined through the normalized Hadamard product

$$C = \frac{A \circ B}{\mathrm{tr}(A \circ B)}, \tag{29}$$

which yields

$$H_\alpha(A,B) = H_\alpha(C). \tag{30}$$

The corresponding matrix-based mutual information is then

$$I_\alpha(A;B) = H_\alpha(A) + H_\alpha(B) - H_\alpha(A,B), \tag{31}$$

and the associated conditional entropy is

$$H_\alpha(A \mid B) = H_\alpha(A,B) - H_\alpha(B). \tag{32}$$

Because these quantities are differentiable with respect to the Gram-matrix entries, they can be incorporated directly into gradient-based representation learning [21].

In our framework, we apply this construction to the predictive setting by forming one matrix from samples of the next latent state and another from samples of the current state–action pair. To encourage predictive dependence, we minimize the negative matrix-based mutual information:

$$\mathcal{L}_{\mathrm{matMI}} = -I_\alpha(A;B). \tag{33}$$

This term serves as a direct alternative to the decomposition-based surrogates introduced above. Instead of separately approximating marginal and conditional entropy terms through latent dispersion and forward prediction, it evaluates predictive dependence from pairwise sample similarities in mini-batches.

The matrix-based estimator is appealing in our setting for two reasons. First, it avoids explicit probability-density modeling in latent space, which can be unstable in high dimensions [18]. Second, it complements predictor-based objectives: the matrix functional captures dependence in a purely non-parametric manner, while transition models provide a more explicit notion of latent dynamics. Prior work has shown that matrix-based information measures can be effective as differentiable regularizers in neural representation learning, especially when combined with model-based objectives rather than used in isolation [19,21,47]. In the experiments, we therefore study this term as an alternative information-theoretic objective within our curiosity-driven framework.

 doi:10.20944/preprints202604.0739.v1

3.7. Reward Prediction Loss

Our agent selects actions by planning in a learned latent dynamics model rather than relying only on model-free value estimates. Given the current observation o_t, we encode it into a latent state $s_t = E_\phi(o_t)$ and use the transition model to roll out hypothetical futures $\hat{s}_{t+1}, \ldots, \hat{s}_{t+H}$ under candidate action sequences. Because the curiosity reward is computed from the predicted state–action embedding, planning requires scoring imagined trajectories using the predicted novelty signal. To turn each imagined step into a scalar return, we additionally learn a reward model that predicts the task reward from the same latent state–action representation, so the planner can evaluate the total predicted return and choose the next action accordingly.

Concretely, the reward model is a small MLP R_ω that takes the concatenation of latent state and action vector

$$y_t \triangleq [\, s_t, \mathbf{a}_t \,], \tag{34}$$

and outputs a scalar reward prediction

$$\hat{r}_t = R_\omega(y_t). \tag{35}$$

During planning, we score a candidate rollout by combining the predicted task reward with the curiosity reward computed on predicted embeddings:

$$\hat{G}_t = \sum_{i=0}^{H-1} \gamma^i \Big(R_\omega(\hat{y}_{t+i}) + \eta\, r^{\text{int}}(\hat{y}_{t+i}) \Big), \qquad \hat{y}_{t+i} \triangleq [\, \hat{s}_{t+i}, \mathbf{a}_{t+i} \,], \tag{36}$$

where $r^{\text{int}}(\cdot)$ follows Eq. (11) and is evaluated on the imagined state–action embedding produced by the transition rollout.

We train R_ω with a terminal-masked n-step regression loss on replay sequences:

$$\mathcal{L}_R^{(n)} = \frac{1}{n} \sum_{i=0}^{n-1} (1 - d_{t+i}) \left\| R_\omega([\, s_{t+i}, \mathbf{a}_{t+i} \,]) - r_{t+i} \right\|_2^2, \qquad s_{t+i} = E_\phi(o_{t+i}), \tag{37}$$

3.8. Putting It Together: Total Training Losses

The encoder and auxiliary prediction modules are trained jointly through a weighted combination of objectives, each corresponding to a different component of the proposed framework. Specifically, we optimize

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{trans}}^{(n)} + \lambda_{\text{back}} \mathcal{L}_{\text{back}}^{(n)} + \lambda_{\text{unif}} \mathcal{L}_{\text{unif}} + \lambda_R \mathcal{L}_R^{(n)} + \lambda_{\text{mat}} \mathcal{L}_{\text{matMI}}, \tag{38}$$

where $\mathcal{L}_{\text{trans}}^{(n)}$ encourages predictive latent dynamics, $\mathcal{L}_{\text{unif}}$ promotes dispersion in the latent space, $\mathcal{L}_R^{(n)}$ trains the reward predictor used for planning, and $\mathcal{L}_{\text{matMI}}$ provides a matrix-based alternative for encouraging predictive dependence. The backward term $\mathcal{L}_{\text{back}}^{(n)}$ corresponds to the reverse-time decomposition discussed earlier.

The coefficients λ_{back}, λ_{unif}, λ_R, and λ_{mat} determine which components are active and how strongly they influence training. This makes the framework modular: predictor-based, matrix-based, and auxiliary reward-modeling terms can be enabled or disabled depending on the experimental setting.

In our final implementation, we do not activate the backward loss. Preliminary experiments suggested that the current backward-prediction design was less effective than the forward predictive objective, and including it did not improve exploration performance. Accordingly, all main results are obtained without $\mathcal{L}_{\text{back}}^{(n)}$.

 doi:10.20944/preprints202604.0739.v1

3.9. Action Selection via Model Predictive Control

At decision time, actions are selected by planning in the learned latent dynamics using model predictive control (MPC). Given the current observation o_t, we first encode it into the latent state

$$s_t = E_\phi(o_t). \tag{39}$$

Starting from this latent state, we evaluate candidate action sequences over a planning horizon H by rolling out the learned transition model:

$$\hat{s}_t = s_t, \tag{40}$$

$$\hat{s}_{t+i+1} = T_\psi([\hat{s}_{t+i}, \mathbf{a}_{t+i}]), \qquad i = 0, \ldots, H-1. \tag{41}$$

This produces imagined latent trajectories under hypothetical future actions, allowing planning to be carried out directly in the learned representation space rather than in the raw observation space.

Each imagined step is assigned a score consisting of two parts: the predicted task reward from the reward model and the intrinsic reward defined on the latent state–action representation. The resulting cumulative return of a candidate action sequence is

$$\hat{G}(s_t, a_{t:t+H-1}) = \sum_{i=0}^{H-1} \gamma^i \Big(R_\omega([\hat{s}_{t+i}, \mathbf{a}_{t+i}]) + \eta\, r^{\text{int}}([\hat{s}_{t+i}, \mathbf{a}_{t+i}]) \Big), \tag{42}$$

where γ is the discount factor and η controls the contribution of the intrinsic reward during planning. In this way, MPC favors action sequences that are predicted to be both task-relevant and informative for exploration.

To convert these rollout scores into action selection, we define a softmax distribution over candidate sequences:

$$p(a_{t:t+H-1} \mid s_t) = \frac{\exp\big(\hat{G}(s_t, a_{t:t+H-1})/\tau\big)}{\sum_{a'_{t:t+H-1}} \exp\big(\hat{G}(s_t, a'_{t:t+H-1})/\tau\big)}, \tag{43}$$

where τ is a temperature parameter controlling the sharpness of the induced planning distribution. We then sample one sequence $\tilde{a}_{t:t+H-1} \sim p(\cdot \mid s_t)$ and execute only its first action:

$$a_t \;=\; [\tilde{a}_{t:t+H-1}]_0. \tag{44}$$

After executing a_t, the agent observes o_{t+1}, re-encodes the updated latent state, and repeats the planning procedure. This receding-horizon strategy allows the planner to continually revise its decisions as new observations become available.

4. Experiments and Results

4.1. Implementation Details and Reproducibility

Replay buffer, optimization, and action representation.

Transitions are stored in a circular replay buffer with capacity 10^6. During training, mini-batches of size $B = 32$ are sampled uniformly from the buffer. Unless otherwise stated, we use a discount factor of $\gamma = 0.9$ and optimize all trainable modules with a learning rate of 5×10^{-3}.

For discrete-action environments, each action is represented as a one-hot vector $\mathbf{a}_t \in \{0,1\}^{|\mathcal{A}|}$. The state–action representation used for novelty estimation and reward prediction is then formed by concatenating the latent state and action vector, i.e., $y_t = [s_t, \mathbf{a}_t]$, as introduced in Eq. (5). For continuous-action settings, the same construction can be used by concatenating the latent state with the raw action vector.

 doi:10.20944/preprints202604.0739.v1

Network architectures.

The latent dimensionality d is controlled by the hyperparameter `internal_dim`, which is set to either 2 or 4 depending on the experiment. For low-dimensional vector observations, the encoder E_ϕ is implemented as a multilayer perceptron with hidden widths $[200, 100, 50, 10]$, followed by a final linear layer that maps to the d-dimensional latent state. For image observations of size 32×32, we use a lightweight convolutional encoder consisting of three convolutional layers with intermediate pooling operations, followed by a linear projection to the latent space.

The forward transition model T_ψ operates on the concatenated state-action input $(s_t, \mathbf{a}_t)$ and is implemented as a residual MLP with hidden width 10 and dropout probability $p = 0.5$. The reward predictor R_ω takes the state–action embedding y_t as input and is implemented as an MLP with layer widths $[10, 50, 20, 1]$, producing a scalar reward estimate for planning.

Unless explicitly noted otherwise, the same architectural choices are used across all experiments to ensure comparability between the different objective variants studied in this work.

Matrix-based objective.

For experiments involving matrix-based entropy estimation, we construct Gram matrices within each mini-batch using a Gaussian kernel. Unless otherwise specified, the matrix-based Rényi entropy uses order $\alpha = 0.99$ and Gaussian-kernel scale parameter $\sigma = 2$, and these settings are kept fixed across runs.

Intrinsic reward hyperparameters.

The combined reward uses the intrinsic scaling $\eta = 1$. For the hybrid intrinsic signal $\alpha = 0.5$ trades off KDE novelty and prediction-error bonus. The prediction-error bonus uses a running mean $\bar{e}$ for normalization and a small constant ϵ for numerical stability.

4.2. Performance Comparison in Acrobot

Environment and protocol.

We evaluate in the Acrobot environment, a two-link underactuated pendulum where only the elbow joint is actuated. The agent applies bounded torques to pump energy into the system until the tip rises above the horizontal target line (Figure 4). Following [20], we use an exploration-focused setting with long-horizon episodes, pixel-based observations, and extrinsic rewards ignored so that progress depends primarily on intrinsic motivation and representation quality.

Metric and baselines.

We report steps-to-goal, where lower values indicate better exploration efficiency, and provide the mean and standard error across random seeds. Baselines reproduced from [20] include ICM [4], RND [14], and the Novelty method NSRS [20]. Our method uses an information-bottleneck-shaped latent space and computes intrinsic reward on state–action novelty.

Figure 4. Acrobot environment and goal. A two-link underactuated pendulum in which only the middle elbow joint is actuated. The agent applies a torque u_t to swing the system upward; success occurs once the tip rises above the horizontal target line.

 doi:10.20944/preprints202604.0739.v1

Table 1. Comparing our performance with baselines in Acrobot. *ICM*—intrinsic reward from next-state prediction error [4]; *RND*—prediction error against a fixed random target network [14]; *Novelty (NSRS)*—k-NN distance-based novelty bonus [20]. Baseline numbers for ICM, RND, and Novelty are reproduced from Table 1 of [20]. Under the standard Gym convention (lower is better), *Ours* (4 seeds) achieves the lowest steps.

Method	Avg	StdErr
ICM	932.8	141.54
RND	953.8	85.98
Novelty	576.0	66.13
Ours	290.0	45.72

Our method reaches the goal in 290 steps on average, improving over the Novelty baseline (NSRS), which requires 576 steps, by 49.7%. It also achieves the smallest standard error (45.72), indicating more consistent discovery across seeds. A mixture of prediction-error-based and density-based rewards outperforms either reward alone, which can be interpreted as an ensembling effect.

4.3. Results on MountainCar

The **MountainCar** environment (see Figure 5) consists of a car moving in a one-dimensional valley between two hills. At each time step the agent chooses a discrete acceleration to the left or right. The engine is not powerful enough to drive straight up the right hill; instead, the agent must first move back and forth to build sufficient momentum before finally climbing to the goal region.

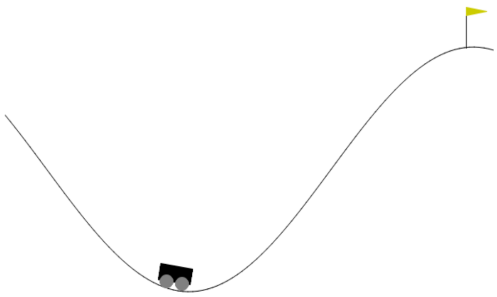

Figure 5. MountainCar environment and goal. A car must build up momentum by moving back and forth before reaching the goal on the right hill. Sparse rewards make exploration difficult.

4.3.1. Experimental results

We evaluate the k-NN based novelty bonus of [20] (*Novelty*) and our IB-based curiosity module (*Ours*). Table 2 reports the average steps to goal (lower is better) and standard error.

Table 2. Performance comparison on MountainCar. Average number of environment steps required to reach the goal (lower is better). *Novelty* is averaged over 4 seeds. *Ours* aggregates two Rényi-based variants over 8 seeds. For runs that did not reach the goal within the training budget (more than 1900), we conservatively treat the last observed step count as completion time when computing statistics.

Method	Avg. steps to goal	StdErr
Novelty	1158.5	172.8
Ours	1456.9	125.7

On MountainCar, our method does not outperform the k-NN based *Novelty* baseline. One plausible explanation is that MountainCar has a low-dimensional state space and simple dynamics, so the k-NN novelty bonus already provides a strong intrinsic signal. In this setting, compressing the state through an information bottleneck and reshaping the intrinsic reward via matrix-based Rényi quantities may introduce additional optimization difficulty without clear benefits.

 doi:10.20944/preprints202604.0739.v1

Taken together with the positive results on Acrobot, these findings suggest that our IB-based curiosity signal is not uniformly superior to simple k-NN novelty bonuses; its advantages appear more pronounced in higher-dimensional or more structurally complex environments.

4.4. Comparing Entropy Estimators for $H(s_{t+1})$

To estimate $\mathrm{I}(s_{t+1}; s_t, a_t) = H(s_{t+1}) - H(s_{t+1} \mid s_t, a_t)$, we compute the conditional term $H(s_{t+1} \mid s_t, a_t)$ with the forward predictor loss in Eq. (19), and compare two surrogates for maximizing the marginal entropy $H(s_{t+1})$ on the same Acrobot setting.

(1) Pairwise Gaussian-potential loss.

Given a mini-batch $\{s_{t+1}^{(i)}\}_{i=1}^{B}$ and an embedding head $\hat{e}(\cdot;\theta_{\hat{e}})$, define $z_i \triangleq \hat{e}(s_{t+1}^{(i)};\theta_{\hat{e}})$. We minimize the expected pairwise Gaussian potential (cf. Eq. (16)):

$$\mathcal{L}_{\mathrm{GP}} = \log\left(\frac{2}{B(B-1)} \sum_{1\le i<j\le B} e^{-t\|z_i - z_j\|_2^2}\right), \tag{45}$$

where $t > 0$ controls the sensitivity to pairwise distances. Minimizing $\mathcal{L}_{\mathrm{GP}}$ encourages larger pairwise distances and thus higher $H(s_{t+1})$.

(2) Matrix-based Rényi entropy.

For the same batch, we build a Gram matrix with a positive-definite kernel κ (Gaussian in our experiments):

$$K_{ij} = \kappa(s_{t+1}^{(i)}, s_{t+1}^{(j)}), \qquad A = \frac{K}{\mathrm{tr}(K)}. \tag{46}$$

The matrix-based Rényi entropy is

$$\widehat{H}_\alpha(s_{t+1}) = \frac{1}{1-\alpha}\log\big(\mathrm{tr}(A^\alpha)\big), \qquad \mathcal{L}_{\mathrm{matH}} \triangleq -\widehat{H}_\alpha(s_{t+1}). \tag{47}$$

Experiment and interpretation.

We keep all other components fixed and only swap the marginal-entropy regularizer: Eq. (45) vs. Eq. (47). Figure 6 plots the corresponding loss values during training; a lower value indicates that the estimator more effectively spreads s_{t+1} in latent space under the same optimization setup.

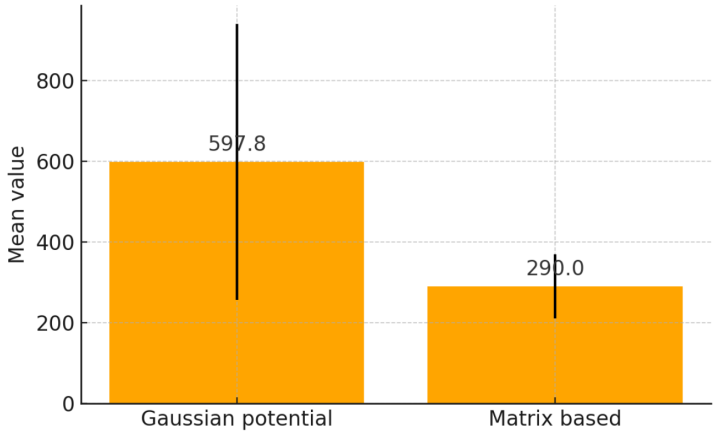

Figure 6. Gaussian-potential vs. matrix-based objective for $H(s_{t+1})$. Comparison of two ways to encourage large next-state entropy on *Acrobot*. The matrix-based objective attains a substantially lower value, indicating faster diversification of s_{t+1}.

 doi:10.20944/preprints202604.0739.v1

On Acrobot, the matrix-based objective decreases faster, suggesting quicker diversification of next-state latents, which benefits density-based novelty estimation. Directly computing mutual information with matrix-based estimators can help the agent diversify states faster in this environment.

4.5. Comparing KDE and KNN

Our intrinsic reward in Eq. (7) depends on estimating the density of continuous state–action embeddings $y_t = [s_t, \mathbf{a}_t]$. Two common non-parametric choices are k-nearest-neighbor scores as used in NSRS [20] and kernel density estimation. We compare them for two reasons. First, to remain comparable with prior work that uses KNN as a strong baseline for novelty bonuses. Second, because in a learned latent space the reward is used at every step, so estimator smoothness affects reward variance and training stability.

KNN-based novelty relies on a hard neighbor set. Small changes in y_t can swap which samples fall into the K nearest neighbors, which can create abrupt changes in the intrinsic reward. KDE assigns soft kernel weights to nearby samples as in Eq. (6), so the estimated density and the resulting $-\log \rho$ reward change more gradually as y_t moves in latent space and as the replay buffer evolves.

Figure 7 shows that KDE converges faster than KNN in our setting. We attribute this mainly to the smoother reward landscape induced by KDE. As the replay buffer grows and embeddings evolve, KDE updates the density estimate gradually, whereas KNN can change abruptly when neighbor identities switch. In practice, this yields a more stable intrinsic bonus and more sample-efficient learning, while preserving the same qualitative novelty ordering.

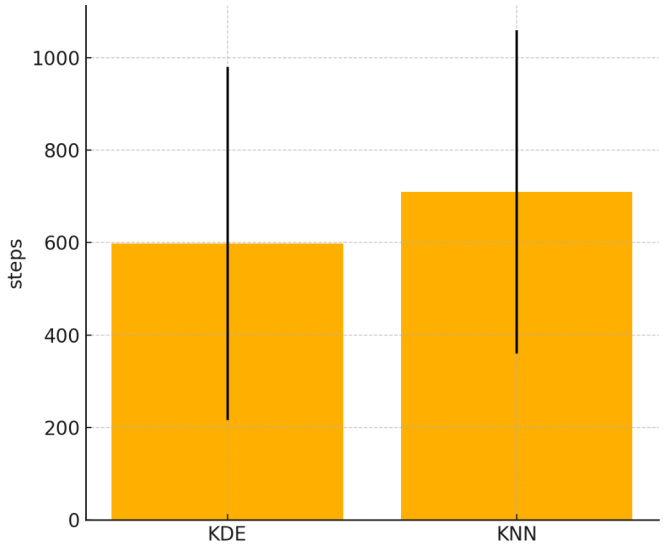

Figure 7. KDE vs. KNN for curiosity density estimation. Under the same latent representation and training setup, KDE reaches a stable intrinsic reward signal in fewer steps than KNN. The smoother kernel weighting reduces reward variance for continuous state–action embeddings, improving optimization stability and accelerating convergence.

5. Conclusions

In this paper, we presented a curiosity-driven exploration framework that combines density-based intrinsic reward with predictive representation learning in a latent space. Our approach defines intrinsic motivation through a hybrid signal that integrates two complementary cues: the rarity of state–action pairs, estimated by KDE in the learned representation space, and the prediction error of the latent dynamics model. We further examined two practical ways to encourage predictive structure in the latent representation, namely a decomposition-based objective and a matrix-based Rényi-entropy formulation.

The empirical results show that this design is effective in environments where exploration depends strongly on representation quality. In Acrobot, the proposed method clearly improves exploration

 doi:10.20944/preprints202604.0739.v1

efficiency over ICM, RND, and a strong k-NN novelty baseline, while KDE also provides a smoother and more stable novelty signal than hard neighbor-based estimation. In MountainCar, by contrast, the advantage is less evident, indicating that the proposed machinery is not uniformly beneficial in simpler low-dimensional settings.

Taken together, these findings suggest that density-based curiosity becomes more informative when novelty is evaluated on compact state–action representations rather than raw observations. They also indicate that matrix-based information objectives can serve as a useful alternative to predictor-based entropy surrogates, particularly when the environment exhibits richer observation structure or more challenging latent dynamics. Future work may further investigate how to better adapt the intrinsic reward and representation objective to environments of different complexity.

Author Contributions: Conceptualization, M.Z. and C.Y.; Methodology, M.Z.; Software, M.Z.; Validation, M.Z.; Writing—original draft preparation, M.Z.; Writing—review & editing, M.Z. and C.Y.; Supervision, C.Y.

Funding: This research was funded by the Key Science and Technology Project of the Ministry of Emergency Management of the People's Republic of China, grant number 2024EMST131302. The APC was funded by the same project (grant number 2024EMST131302).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new datasets were generated or analyzed in this study. All results were obtained in simulation environments, and no external dataset was used. The code for this work is available from the first author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sutton, R.S.; Barto, A.G.; et al. *Reinforcement learning: An introduction*; Vol. 1, MIT press Cambridge, 1998.
2. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *nature* **2015**, *518*, 529–533.
3. Ecoffet, A.; Huizinga, J.; Lehman, J.; Stanley, K.O.; Clune, J. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995* **2019**.
4. Pathak, D.; Agrawal, P.; Efros, A.A.; Darrell, T. Curiosity-driven Exploration by Self-supervised Prediction. In Proceedings of the Proceedings of the 34th International Conference on Machine Learning. PMLR, 2017, pp. 2778–2787.
5. Kayal, A.; Pignatelli, E.; Toni, L. The impact of intrinsic rewards on exploration in Reinforcement Learning. *Neural Computing and Applications* **2025**, pp. 1–35.
6. Aubret, A.; Matignon, L.; Hassas, S. An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey. *Entropy* **2023**, *25*, 327.
7. Cobbe, K.; Hesse, C.; Hilton, J.; Schulman, J. Leveraging procedural generation to benchmark reinforcement learning. In Proceedings of the International conference on machine learning. PMLR, 2020, pp. 2048–2056.
8. Raileanu, R.; Goldstein, M.; Yarats, D.; Kostrikov, I.; Fergus, R. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems* **2021**, *34*, 5402–5415.
9. Moure, P.; Cheng, L.; Ott, J.; Wang, Z.; Liu, S.C. Regularized parameter uncertainty for improving generalization in reinforcement learning. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 23805–23814.
10. Oudeyer, P.Y.; Kaplan, F. What is intrinsic motivation? A typology of computational approaches. *Frontiers in neurorobotics* **2007**, *1*, 108.
11. Oudeyer, P.Y.; Kaplan, F. How can we define intrinsic motivation? In Proceedings of the the 8th international conference on epigenetic robotics: Modeling cognitive development in robotic systems. Lund University Cognitive Studies, Lund: LUCS, Brighton, 2008.
12. Chen, X.; Twomey, K.E.; Westermann, G. Curiosity enhances incidental object encoding in 8-month-old infants. *Journal of Experimental Child Psychology* **2022**, *223*, 105508.

 doi:10.20944/preprints202604.0739.v1

13. Babik, I.; Galloway, J.C.; Lobo, M.A. Early exploration of one's own body, exploration of objects, and motor, language, and cognitive development relate dynamically across the first two years of life. *Developmental psychology* **2022**, *58*, 222.
14. Burda, Y.; Edwards, H.; Storkey, A.; Klimov, O. Exploration by Random Network Distillation. In Proceedings of the International Conference on Learning Representations, 2019.
15. Aubret, A.; Matignon, L.; Hassas, S. A survey on intrinsic motivation in reinforcement learning. *arXiv preprint arXiv:1908.06976* **2019**.
16. Beyer, K.S.; Goldstein, J.; Ramakrishnan, R.; Shaft, U. When Is "Nearest Neighbor" Meaningful? In Proceedings of the Proceeding of the 7th International Conference on Database Theory; Beeri, C.; Bruneman, P., Eds. Springer, 1999, Vol. 1540, *Lecture Notes in Computer Science*, pp. 217–235.
17. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. In Proceedings of the Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing, 1999, pp. 368–377.
18. Giraldo, L.G.S.; Rao, M.; Príncipe, J.C. Measures of entropy from data using infinitely divisible kernels. *IEEE Transactions on Information Theory* **2014**, *61*, 535–548.
19. Yu, S.; Sanchez Giraldo, L.; Principe, J. Information-Theoretic Methods in Deep Neural Networks: Recent Advances and Emerging Opportunities. In Proceedings of the Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21; Zhou, Z.H., Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 4669–4678. Survey Track, https://doi.org/10.24963/ijcai.2021/633.
20. Tao, R.Y.; François-Lavet, V.; Pineau, J. Novelty Search in Representational Space for Sample Efficient Exploration. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020.
21. Yu, S.; Alesiani, F.; Yu, X.; Jenssen, R.; Príncipe, J.C. Measuring Dependence with Matrix-based Entropy Functional. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 10725–10733.
22. Pathak, D.; Gandhi, D.; Gupta, A. Self-supervised exploration via disagreement. In Proceedings of the International conference on machine learning. PMLR, 2019, pp. 5062–5071.
23. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Proceedings of the Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016, Vol. 48, *PMLR*, pp. 1050–1059.
24. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In Proceedings of the Advances in Neural Information Processing Systems, 2017, Vol. 30, pp. 6402–6413.
25. Li, H.; Yu, S.; Francois-Lavet, V.; Principe, J.C. Reward-Free Exploration by Conditional Divergence Maximization. In Proceedings of the International Conference on Learning Representations, 2024.
26. Houthooft, R.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; Abbeel, P. VIME: Variational Information Maximizing Exploration. In Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 1109–1117.
27. Hong, Z.W.; Shann, T.Y.; Su, S.Y.; Chang, Y.H.; Fu, T.J.; Lee, C.Y. Diversity-driven exploration strategy for deep reinforcement learning. *Advances in neural information processing systems* **2018**, *31*.
28. Scott, P.D.; Markovitch, S. Learning Novel Domains Through Curiosity and Conjecture. In Proceedings of the Proceedings of International Joint Conference for Artificial Intelligence, Detroit, Michigan, 1989; pp. 669–674.
29. Seo, Y.; Chen, L.; Shin, J.; Lee, H.; Abbeel, P.; Lee, K. State entropy maximization with random encoders for efficient exploration. In Proceedings of the International conference on machine learning. PMLR, 2021, pp. 9443–9454.
30. Yarats, D.; Fergus, R.; Lazaric, A.; Pinto, L. Reinforcement learning with prototypical representations. In Proceedings of the International Conference on Machine Learning. PMLR, 2021, pp. 11920–11931.
31. Pei, Y.; Hou, X. Learning Representations in Reinforcement Learning: An Information Bottleneck Approach. *arXiv preprint arXiv:1911.05695* **2019**.
32. Bai, S.; Zhou, W.; Ding, P.; Zhao, W.; Wang, D.; Chen, B. Rethinking Latent Redundancy in Behavior Cloning: An Information Bottleneck Approach for Robot Manipulation. *arXiv preprint arXiv:2502.02853* **2025**. Accepted by ICML 2025.
33. You, B.; Liu, H. Multimodal Information Bottleneck for Deep Reinforcement Learning with Multiple Sensors. *Neural Networks* **2024**, *176*, –. Also available as arXiv preprint arXiv:2410.17551.

 doi:10.20944/preprints202604.0739.v1

34. Islam, R.; Zang, H.; Tomar, M.; Didolkar, A.; Islam, M.M.; Arnob, S.Y.; Iqbal, T.; Li, X.; Goyal, A.; Heess, N.; et al. Representation Learning in Deep RL via Discrete Information Bottleneck. In Proceedings of the Proceedings of the 26th International Conference on Artificial Intelligence and Statistics. PMLR, 2023, pp. 8699–8722.
35. Zou, Q.; Suzuki, E. Compact Goal Representation Learning via Information Bottleneck in Goal-Conditioned Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**. https://doi.org/10.1109/TNNLS.2023.3344880.
36. Kich, V.A.; Bottega, J.A.; Steinmetz, R.; Grando, R.B.; Yorozu, A.; Ohya, A. CURLing the Dream: Contrastive Representations for World Modeling in Reinforcement Learning. *arXiv preprint arXiv:2408.05781* **2024**.
37. Loewenstein, G. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin* **1994**, *116*, 75.
38. Burda, Y.; Edwards, H.; Pathak, D.; Storkey, A.; Darrell, T.; Efros, A.A. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355* **2018**.
39. Berlyne, D.E.; et al. A theory of human curiosity **1954**.
40. Kidd, C.; Hayden, B.Y. The psychology and neuroscience of curiosity. *Neuron* **2015**, *88*, 449–460.
41. Dubey, R.; Griffiths, T.L. Understanding exploration in humans and machines by formalizing the function of curiosity. *Current Opinion in Behavioral Sciences* **2020**, *35*, 118–124.
42. Schultz, W.; Dayan, P.; Montague, P.R. A neural substrate of prediction and reward. *Science* **1997**, *275*, 1593–1599.
43. Bellemare, M.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; Munos, R. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems* **2016**, *29*.
44. Tsuda, K.; Rätsch, G.; Warmuth, M.K. Matrix Exponentiated Gradient Updates for On-line Learning and Bregman Projection. *Journal of Machine Learning Research* **2005**, *6*, 995–1018.
45. Bhatia, R. Infinitely Divisible Matrices. *The American Mathematical Monthly* **2006**, *113*, 221–235. https://doi.org/10.2307/27641890.
46. Yu, S.; Sanchez Giraldo, L.G.; Jenssen, R.; Príncipe, J.C. Multivariate Extension of Matrix-Based Rényi's α-Order Entropy Functional. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**, *42*, 2960–2966. https://doi.org/10.1109/TPAMI.2019.2932976.
47. Yu, X.; Yu, S.; Príncipe, J.C. Deep Deterministic Information Bottleneck with Matrix-Based Entropy Functional. In Proceedings of the ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 3160–3164. https://doi.org/10.1109/ICASSP39728.2021.9414151.