Curiosity-Driven Exploration via Latent Bayesian Surprise

Pietro Mazzaglia, Ozan Catal, Tim Verbelen, Bart Dhoedt

IDLab, Ghent University
{firstname}.{lastname}@ugent.be

Abstract

The human intrinsic desire to pursue knowledge, also known as curiosity, is considered essential in the process of skill acquisition. With the aid of artificial curiosity, we could equip current techniques for control, such as Reinforcement Learning, with more natural exploration capabilities. A promising approach in this respect has consisted of using Bayesian surprise on model parameters, i.e. a metric for the difference between prior and posterior beliefs, to favour exploration. In this contribution, we propose to apply Bayesian surprise in a latent space representing the agent's current understanding of the dynamics of the system, drastically reducing the computational costs. We extensively evaluate our method by measuring the agent's performance in terms of environment exploration, for continuous tasks, and looking at the game scores achieved, for video games. Our model is computationally cheap and compares positively with current state-of-the-art methods on several problems. We also investigate the effects caused by stochasticity in the environment, which is often a failure case for curiosity-driven agents. In this regime, the results suggest that our approach is resilient to stochastic transitions.

1 Introduction

Agents can be trained with Reinforcement Learning (RL) to successfully accomplish tasks by maximising a reward signal that encourages correct behaviors and penalizes wrong actions. For instance, agents can learn to play video games by maximizing the game score (Mnih et al. 2015) or achieve robotic manipulation tasks, such as solving a Rubik's cube (OpenAI et al. 2019), by following human-engineered rewards. However, how to correctly define reward functions to develop general skills remains an unsolved problem, and it is likely to stumble across undesired behaviours when designing rewards for complex tasks (Amodei et al. 2016; Clark and Amodei 2016; Krakovna et al. 2020; Popov et al. 2017).

In contrast to RL agents, humans can learn behaviors without any external rewards, due to the intrinsic motivation that naturally drives them to be active and explore the environment (Larson and Rusk 2011; Legault 2016). The design of similar mechanisms for RL agents opens up possibilities for training and evaluating agents without external rewards (Matusch, Ba, and Hafner 2020), fostering more self-supervised strategies of learning.

The idea of instilling intrinsic motivation, or 'curiosity', into artificial agents has raised a large interest in the RL community (Oudeyer, Kaplan, and Hafner 2007; Schmidhuber 1991), where curiosity is used to generate intrinsic rewards that replace or complement the external reward function. However, what is the best approach to generate intrinsic bonuses is still unsettled and current techniques underperform in certain domains, such as stochastic or ambiguous environments (Wauthier et al. 2021).

Several successful approaches modeled intrinsic rewards as the 'surprisal' of a model. In layman's terms, this can be described as the difference between the agent's belief about the environment state and the ground truth, and can be implemented as the model's prediction error (Achiam and Sastry 2017; Pathak et al. 2017). However, searching for less predictable states suffers from the 'NoisyTV problem', where watching a screen outputting white random noise appears more interesting than other exploratory behaviours (Schmidhuber 2010). This because the noise of the TV is stochastic and thus results generally more interesting than the rest of the environment (Burda et al. 2019a).

In contrast, Bayesian surprise (Itti and Baldi 2006) measures the difference between the posterior and prior beliefs of an agent, after observing new data. As we also show in this work, this means that for stochastic transitions of the environment, which carry no novel information to update the agent's beliefs, low intrinsic bonuses are provided, potentially overcoming the NoisyTV issue. Previous work adopting Bayesian surprise for exploration has mostly focused on evaluating surprise in the model's parameter space (Houthooft et al. 2016), which suffers from being computationally-expensive.

Contributions. In this work, we present a new curiosity bonus based on the concept of Bayesian surprise. Establishing a latent variable model in the task dynamics, we derive **L**atent **B**ayesian **S**urprise (**LBS**) as the difference between the posterior and prior beliefs of a latent dynamics model. Our dynamics model uses the random variable in latent space to predict the future, while at the same time capturing any uncertainty in the dynamics of the task.

The main contributions of the work are as follows: (*i*) a latent dynamics model, which captures the information about the dynamics of the environment in an unobserved variable that is used to predict the future state, (*ii*) a new Bayesian surprise inspired exploration bonus, derived as the information gained with respect to the latent variable in the dynamics model, (*iii*) evaluation of the exploration capabilities on several continuous-actions robotic simulation tasks and on discrete-actions video games, and comparison with other exploration strategies, and (*iv*) assessment of the robustness to stochasticity, by comparing to the other baselines on tasks with stochastic transitions in the dynamics.

The results empirically show that our LBS method either performs on par and often outperforms state-of-the-art methods, when the environment is mostly deterministic, making it a strongly valuable method for exploration. Furthermore, similarly to methods using Bayesian surprise in parameter space, LBS is resilient to stochasticity, and actually explores more in-depth than its parameter space counterparts in problems with stochastic dynamics, while also being computationally cheaper. Further visualization, is available on the project webpage.[1]

2 Background

We focus on exploration bonuses to incentivize exploration in RL. To foster the reader's understanding, we first introduce standard notation and common practices.

Markov Decision Processes. The RL setting can be formalized as a Markov Decision Process (MDP), which is denoted with the tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, T, R, \gamma\}$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, T is the state transition function, also referred to as the dynamics of the environment, R is the reward function, which maps transitions into rewards, and γ is a discount factor. The dynamics of the task can be described as $p(s_{t+1}|s_t, a_t)$ that is the probability that action a_t brings the system to state s_{t+1} from state s_t, at the next time step $t+1$. The objective of the RL agent is to maximize the expected discounted sum of rewards over time, also called return, and indicated as $G_t = \sum_{k=t+1}^{T} \gamma^{(k-t-1)} r_k$.

Policy Optimization. In order to maximize the returns, the agent should condition its actions on the environment's current state. The policy function $\pi(a_t|s_t)$ is used to represent the probability of taking action a_t when being in state s_t. Several policy-optimization algorithms also evaluate two value functions, $V(s_t)$ and $Q(s_t, a_t)$, to estimate and predict future returns with respect to a certain state or state-action pair, respectively.

Intrinsic Motivation. Curious agents are designed to search for novelty in the environment and to discover new behaviours, driven by an intrinsically motivated signal. Practically, this comes in the form of self-generated rewards $r^{(i)}$ that can complement or replace the external rewards $r^{(e)}$ of the environment. The combined reward at time step t can be represented as: $r_t = \eta_e r_t^{(e)} + \eta_i r_t^{(i)}$, where η_e and η_i are factors adopted to balance external and intrinsic rewards. How to optimally balance between exploration with intrinsic motivation and exploitation of external rewards is still an unanswered question, which we do not aim to address with our method. Instead, similarly to what done in other works (Shyam, Jaśkowski, and Gomez 2019; Burda et al. 2019a; Pathak, Gandhi, and Gupta 2019; Ratzlaff et al. 2020; Tao, Francois-Lavet, and Pineau 2020), we focus on the exploration behaviour emerging from the self-supervised intrinsic motivation signal.

Surprisal and Bayesian Surprise. The surprisal, or information content, of a random variable is defined as the negative logarithm of its probability distribution. In an MDP, at time step t, we can define the surprisal with respect to next-step state as $-\log p(s_{t+1}|s_t, a_t)$. By using a model with parameters θ to fit the transition dynamics of the task, we can define surprisal in terms of the probability estimated by the model, namely $-\log p_\theta(s_{t+1}|s_t, a_t)$. Such surprisal signal has been adopted for exploration in several works (Achiam and Sastry 2017; Pathak et al. 2017). One shortcoming of these methods is that a stochastic transition, e.g. rolling a die, will always incur into high surprisal values, despite the model having observed the same transition several times. This problem has been treated in literature as the 'NoisyTV problem' (Schmidhuber 2009, 2010).

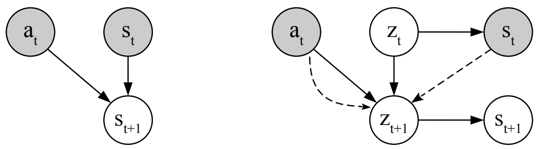
In contrast, Bayesian surprise (Itti and Baldi 2006) can be defined as the information gained about a random variable, by observing another random variable. For instance, we can compute the information gained about the parameters of the model θ by observing new states as $\mathcal{I}(\theta; s_{t+1}|s_t, a_t)$. Such signal has been used for exploration exploiting Bayesian neural networks (Houthooft et al. 2016), where Bayesian surprise is obtained by comparing the weights distribution before and after updating the model with newly collected states. However, this procedure is extremely expensive, as it requires an update of the model for every new transition. Alternatively, an approximation of Bayesian surprise is obtainable by using the variance of an ensemble of predictors (Pathak, Gandhi, and Gupta 2019; Sekar et al. 2020), though this method still requires to train several models.

3 Latent Bayesian Surprise

Our method provides intrinsic motivation through a Bayesian surprise signal that is computed with respect to a latent variable. First, we describe how the latent dynamics model works and how it allows the computation of Bayesian surprise in latent space. Then, we present an overview of the different components of our model and explain how they are concurrently trained to fit the latent dynamics, by exploiting variational inference. Finally, we show how the intrinsic reward signal for LBS is obtained from the model's predictions and discuss connections with other methods.

Latent Dynamics. The transition dynamics of an MDP can be summarized as the probability of the next state, given the current state and the action taken at the current time step, namely $p(s_{t+1}|s_t, a_t)$. The associated generative process is presented in Figure 1a. In the case of deterministic dynamics, the next state is just a function of the current state and action. For non-deterministic dynamics, there would be a distribution over the next state, from which samples are drawn when the state-action pair is triggered. The entropy of such distribution determines the uncertainty in the dynamics.

[1]https://lbsexploration.github.io/



(a) MDP dynamics (b) Latent dynamics (ours)

Figure 1: **Dynamics graphical models.** The model observes s_t and a_t. Solid lines indicate generative processes and dashed lines indicate the inference ones.

With the aim of capturing the environment's uncertainty and to compute the Bayesian surprise given by observing new states, we designed the latent dynamics model in Figure 1b. The intermediate latent variable z_{t+1} should contain all the necessary information to generate s_{t+1}, so that by inferring the latent probability distribution as $p(z_{t+1}|s_t, a_t)$ from previous state and action, we can then estimate future state probability as $p(s_{t+1}|z_{t+1})$.

As we discuss later in this Section, we can train a model to maximize an evidence lower bound on the future states likelihood that matches our latent variable model. Then, the most appealing aspect for exploration is that we can now compute Bayesian surprise in latent space as $\mathcal{I}(z_{t+1}; s_{t+1}|s_t, a_t)$, which is the information gained with respect to the latent variable by observing the actual state.

Model Overview. A dynamics model with parameters θ can be trained to match the environment dynamics (as in Figure 1a) by maximizing the log-likelihood $\log p(s_{t+1}|s_t, a_t)$ of its predictions.

Similarly, given our latent variable model, we can train a dynamics model to maximize an evidence lower bound on the log-likelihood of future states. For this purpose, the LBS model is made of the following components:

$$\begin{aligned} &\text{Latent Prior:} && p_\theta(z_{t+1}|s_t, a_t), \\ &\text{Latent Posterior:} && q_\theta(z_{t+1}|s_t, a_t, s_{t+1}), \\ &\text{Reconstruction model:} && p_\theta(s_{t+1}|z_{t+1}), \end{aligned}$$

which are displayed in Figure 2. The latent prior component represents prior beliefs over the next state's latent variable.

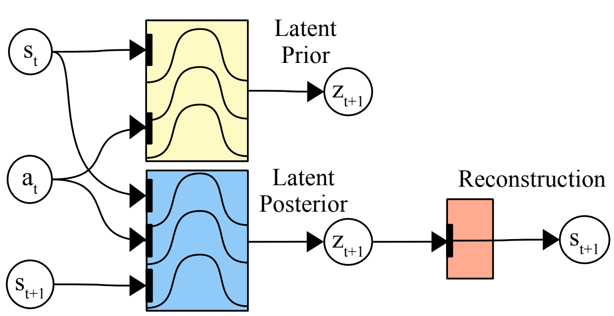


Figure 2: **LBS overview.** The modules of LBS, with input and output variables. Latent Prior and Posterior output distributions, while the Reconstruction model outputs point estimates.

The latent posterior $q(z_{t+1})$ represents a variational distribution that approximates the true posterior of the latent variable, given the observed data s_{t+1}. Finally, the reconstruction module allows to generate the next state from the corresponding latent. Overall, the model resembles a conditional VAE (Kingma and Welling 2014), trained to autoencode the next states, conditioned on current states and actions.

All the components parameters θ are jointly optimized by maximizing the following variational lower bound on future states log-likelihood:

$$\begin{aligned} \mathcal{J} =& \mathbb{E}_{z_{t+1}\sim q(z)}[\log p_\theta(s_{t+1}|z_{t+1})] \\ &- \beta D_{\mathrm{KL}}[q_\theta(z_{t+1}|s_t, a_t, s_{t+1}) \| p_\theta(z_{t+1}|s_t, a_t)] \end{aligned} \tag{1}$$

where β is introduced to control disentanglement in the latent representation, as in (Higgins et al. 2017). The derivation of the objective is available in the Appendix.

Intrinsic Rewards. In our method, we are interested in measuring the amount of information that is gained by the model when facing a new environment's transition and using that as an intrinsic reward to foster exploration in RL. Every time the agent takes action a_t while being in state s_t, it observes a new state s_{t+1} that completes the transition and brings new information to the dynamics model.

Such information gain can be formulated as the KL divergence between the latent prior and its approximate posterior and adopted as an intrinsic reward for RL as follows:

$$\begin{aligned} r_t^{(i)} &= \mathcal{I}(z_{t+1}; s_{t+1}|s_t, a_t) \\ &\approx D_{\mathrm{KL}}[q_\theta(z_{t+1}|s_t, a_t, s_{t+1}) \| p_\theta(z_{t+1}|s_t, a_t)] \end{aligned} \tag{2}$$

The above term can be efficiently computed by comparing the distributions predicted by the latent prior and the latent posterior components. The signal provided should encourage the agent to collect transitions where the predictions are more uncertain or erroneous.

The intrinsic motivation signal of LBS can also be reformulated as (conditioning left out for abbreviation):

$$\begin{aligned} &D_{\mathrm{KL}}[q_\theta(z_{t+1}) \| p_\theta(z_{t+1})] = \\ &= \mathbb{E}_{q_\theta(z_{t+1})}[\log q_\theta(z_{t+1}) - \log p_\theta(z_{t+1})] \\ &= -H[q_\theta(z_{t+1})] + H[q_\theta(z_{t+1}), p_\theta(z_{t+1})] \end{aligned} \tag{3}$$

where the left term is the entropy of the latent posterior and the right term is the cross-entropy of p relatively to q. Maximizing our bonus can thus be interpreted as searching for states with minimal entropy of the posterior and a high cross-entropy value between the posterior and the prior.

Assuming the LBS posterior correctly approximates the true posterior of the system dynamics, the cross-entropy term closely resembles the 'surprisal' bonus adopted in other works (Achiam and Sastry 2017; Pathak et al. 2017; Burda et al. 2019a). Using LBS can then be seen as maximizing the 'surprisal', while trying to avoid high-entropy, stochastic states.

4 Experiments

The aim of the experiments is to compare the performance of the LBS model and intrinsic rewards against other approaches for exploration in RL.

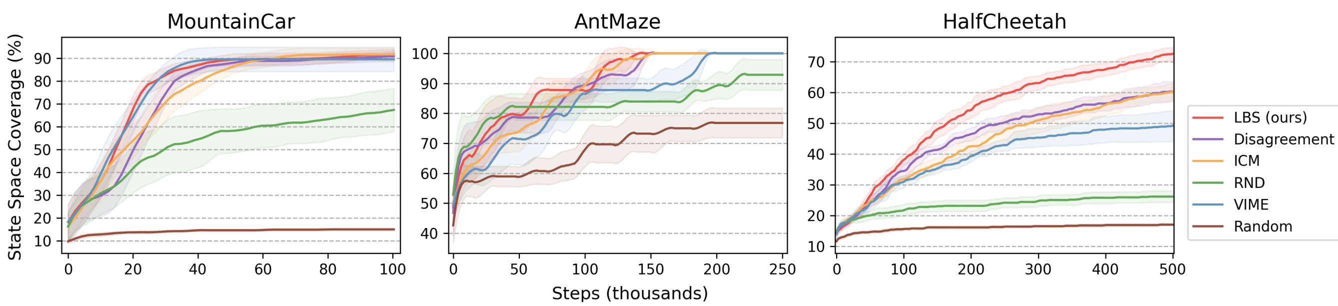


Figure 3: **Continuous Control results.** A comparison of our method against several baselines on continuous control tasks. Lines show the average state-space coverage (standard deviations in shade) in terms of percentage of bins visited by the agents.

Environments. Main results are presented with respect to three sets of environments: continuous control tasks, discrete-action games, and tasks with stochastic transitions. The continuous control tasks include the classic Mountain Car environment (Moore 1990), the Mujoco-based Half-Cheetah environment (Todorov, Erez, and Tassa 2012), and the Ant Maze environment used in (Shyam, Jaśkowski, and Gomez 2019). The discrete-action games include 8 video games from the Atari Learning Environment (ALE; Bellemare et al. (2013)) and the Super Mario Bros. game, which is a popular NES platform game. The stochastic tasks include an image-prediction task with stochastic dynamics and two stochastic variants of Mountain Car, including a NoisyTV-like component.

In this Section, we consider curious agents that only optimize their self-supervised signal for exploration. This means that we omit any external rewards, by setting $\eta_e = 0$ (see Background). This focuses the agents solely on the exploratory behaviors inspired by the curiosity mechanisms. For all tasks, we update the policy using the Proximal Policy Optimization algorithm (PPO; Schulman et al. (2017)). For all model's components, we use neural networks. For the model's latent stochastic variable, we use distributional layers implemented as linear layers that output the means and standard deviations of a multivariate gaussian.

Zero-shot Adaptation. We present additional experiments on the Deep Mind Control suite (Tassa et al. 2018) in the Appendix. As in Plan2Explore (Sekar et al. 2020), we use intrinsic motivation to train an exploration policy, which collects data to improve the agent's model. Then, the model is used to train an exploitative policy on the environment's rewards and its zero-shot performance is evaluated. In these visual control tasks, we show that the intrinsic motivation bonus of LBS combines well with model-based RL, achieving similar or higher performance than Plan2Explore and requiring no additional predictors to be trained.

4.1 Continuous Control

In our continuous control experiments, we discretize the state-space into bins and compare the number of bins explored, in terms of coverage percentage. An agent being able to visit a certain bin corresponds to the agent being able to solve an actual task that requires reaching that certain area of the state space. Thus, it is important that a good exploration method would be able to reach as many bins as possible.

We compare against the following baselines:

- *Disagreement* (Pathak, Gandhi, and Gupta 2019): an ensemble of models is trained to match the environment dynamics. The variance of the ensemble predictions is used as the curiosity signal.
- *Intrinsic Curiosity Model* (ICM; Pathak et al. (2017)): intrinsic rewards are computed as the mean-squared error (MSE) between a dynamics model's predictions in feature space and the true features. States are processed into features using a feature network, trained jointly with the model to optimize an inverse-dynamics objective.
- *Random Network Distillation* (RND; Burda et al. (2019b)): features are obtained with a fixed randomly initialized neural network. Intrinsic rewards for each transition are the prediction errors between next-state features and the output of a distillation network, trained to match the outputs of the random feature network.
- *Variational Information Maximizing Exploration* (VIME; Houthooft et al. (2016)): the dynamics is modeled as a Bayesian neural network (BNN; Bishop (1997)). Intrinsic rewards for single transitions are shaped as the information gain computed with respect to the BNN's parameters before and after updating the network, using the new transition's data.
- *Random*: an agent that explores by performing a series of random actions. Note that employing random actions is equivalent to having a policy with maximum entropy of actions, for each state. Thus, despite its simplicity, the random baseline provides a metric of how in-depth do maximum entropy RL methods explore, when receiving no external rewards (Haarnoja et al. 2018).

We found LBS to be working best in this benchmark, as it explores the most in-depth and the most efficiently in all tasks. The training curves are presented in Figure 3, averaging over runs with eight different random seeds. Further comparison against RIDE (Raileanu and Rocktäschel 2020) and NGU (Badia et al. 2020b), which employ episodic counts to modulate exploration, are presented in Appendix.

Mountain Car. In Mountain Car, the two-dimensional state space is discretized into 100 bins. Figure 3 shows that LBS, VIME, ICM and Disagreement all reach similar final performance, with around 90% of coverage. In particular,

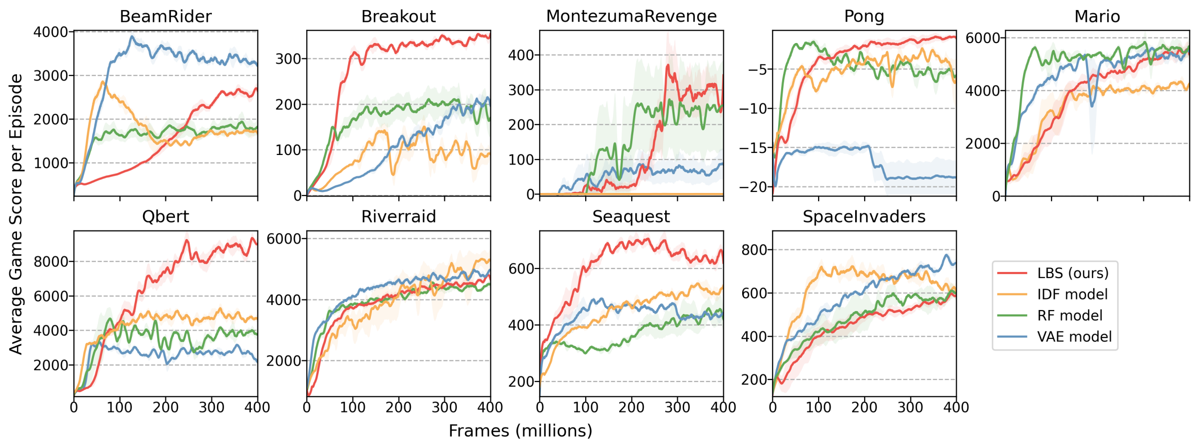


Figure 4: **Arcade Games results.** A comparison of LBS against surprisal-based models, using different sets of features, on 8 selected Atari and the Super Mario Bros. games. Lines show the average game score per episode (standard deviations in shade).

LBS and VIME are on average faster at exploring in the first 30k steps. RND struggles behind with about 67% of visited bins, doing better only than the Random baseline (∼15%).

Ant Maze. In the Ant Maze environment, the agent can explore up to seven bins, corresponding to different aisles of a maze. LBS, ICM and Disagreement perform best in this environment, reaching the end of the maze in all runs and before 150k steps. VIME also reaches 100% in all runs but takes longer. RND and the Random baselines saturate far below 100% coverage.

Half-Cheetah. In the Half-Cheetah environment, the state space is discretized into 100 bins. In this task, which has the most complex dynamics compared to the others, LBS reaches the highest number of bins, with around 73% of coverage. ICM and Disagreement follow with ∼60%, and VIME with ∼49%. RND lacks behind by doing slightly better than the Random baseline (∼26% vs ∼17%).

4.2 Arcade Games

For the arcade games, the environments chosen are designed in a way that either requires the player to explore in order to succeed, e.g. Qbert, or to survive as long as possible to avoid boredom, e.g. Pong. For this reason, agents are trained only with curiosity but evaluated on the game score they achieve in one episode, or, in the case of Super Mario Bros., on the distance traveled from the start position. Higher scores in this benchmark translate either into an higher number of enemies killed, objects collected, or transitions/areas/levels of the game explored, meaning that methods that perform better on this benchmark are more likely to discover meaningful skills in the environment. Combining curiosity with environment's rewards, performance in these games could be significantly improved with respect to using only curiosity but we do not compare to that setting in order to completely focus on the exploration performance.

We follow the setup of (Burda et al. 2019a) and compare against their baselines, which use MSE prediction error in feature space as the intrinsic motivation signal, aka surprisal in feature space. The feature space is obtained by projecting states from the environment into a lower-dimensional space using a feature model, i.e. next-state features can be expressed as $\phi_{t+1} = f(s_{t+1})$.

The different baselines use different feature models, so that the Variational Autoencoder, or *VAE model*, trains an autoencoder, as in (Kingma and Welling 2014), concurrently with the dynamics model; the Random Features, or *RF model*, uses a randomly initialized network; the Inverse Dynamics Features, or *IDF model*, uses features that allow to model the inverse dynamics of the environment.

For LBS, we also found that working in a reduced feature space, compared to the high-dimensional pixel space, is beneficial. For this purpose, we project the states from the environment into a low-dimensional feature space using a randomly initialized network, similarly to the RF model. We believe more adequate features than random could be found, though we leave this idea for future studies. In this setup, the reconstruction model predicts next-state features instead of next-state pixels:

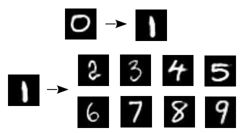
$$\text{Reconstruction model:} \qquad p_\theta(\phi_{t+1}|z_{t+1}).$$

A performance comparison between using pixel and feature reconstruction is provided in the Appendix.

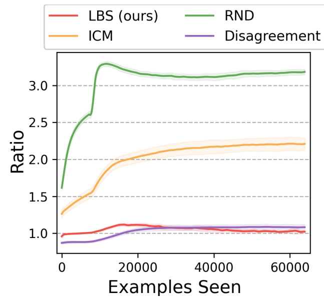
The training curves are shown in Figure 4, presenting the original results from (Burda et al. 2019a) for the baselines and an average of five random seed runs for LBS. The empirical results are favorable towards the LBS model, which achieves the best average final score in 5 out 9 games: Montezuma Revenge, Pong, Seaquest, Breakout, and Qbert, with a large margin for the latter three; and performs comparably to the other baselines in all other games.

4.3 Stochastic Environments

Our stochastic benchmark is composed of three tasks: an image-prediction task, where we quantitatively assess the intrinsic rewards assigned by each method for deterministic and stochastic transitions, and two stochastic variants of the Mountain Car control problem, presenting an additional state that is randomly controlled by an additional action. The additional low-dimensional state in Mountain Car can



(a) Task Dynamics



(b) Baselines comparison

Figure 5: **Stochastic MNIST.** (a) Image prediction stochastic task based on the MNIST dataset samples. (b) Average intrinsic motivation ratio over training samples, in ten runs. The closer the ratio to the unity, at convergence, the better.

be seen as a one-pixel NoisyTV that is controlled by the additional action's remote.

Image Task. Similarly to (Pathak, Gandhi, and Gupta 2019), we employ the Noisy MNIST dataset (LeCun et al. 1995) to perform an experiment on stochastic transitions. Taking examples from the test set of MNIST, we establish a fictitious dynamics that always starts either from an image of a zero or a one: a 0-image always transitions to a 1-image, while a 1-image transitions into an image representing a digit between two and nine (see Figure 5a).

We assess the performance in terms of the ratio between the intrinsic motivation provided for transitions starting from 1-images and transitions starting from 0-images. After having seen several samples starting from the 1-image, the agent should eventually understand that results associated with this more stochastic transition do not bring novel information about the task dynamics, and should lose interest with respect to it. Thus, the expected behavior is that the ratio should eventually lean to values close to the unity.

We train the models uniformly sampling random transitions in batches of 128 samples and run the experiments with ten random seeds. In Figure 5b, we compare LBS to Disagreement, ICM and RND. We observe that LBS and Disagreement are the only methods that eventually overcome the stochasticity in the transitions starting from 1-images, maintaining a ratio of values close to one at convergence. Both ICM and RND, instead, keep finding the stochastic transition more interesting at convergence.

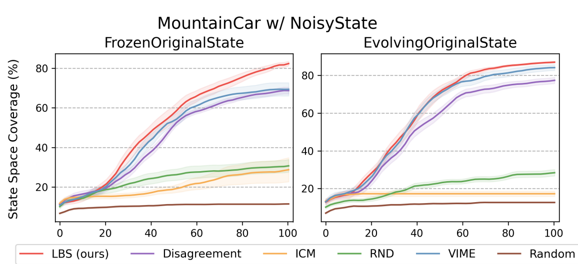
Stochastic Mountain Car. The original Mountain Car continuous control problem is made of a two-dimensional state space, position and velocity of the car, which we refer to as the OriginalState, and a one-dimensional action space, controlling the force to apply to the car to move. We extended the environment to be stochastic by adding a one-dimensional state, referred to as the NoisyState, and a one-dimensional action, ranging from $[-1, 1]$ which works as a remote for the NoisyState. When this action's value is higher than 0, the remote is triggered, updating the NoisyState value, by sampling uniformly from the $[-1, 1]$ interval. Otherwise, the task works like the standard Mountain Car, and the agent can explore up to 100 bins.

We experiment with two versions of the environment:

- *FrozenOriginalState*: when the remote is triggered, the OriginalState is kept frozen, regardless of the force applied to the car. This allows the agent to focus on the NoisyState changes, whilst not losing the velocity and the momentum of the car.
- *EvolvingOriginalState*: when the remote is triggered, the OriginalState is updated but the force applied is zero. This means the agent has to decide whether giving up on the original task to focus on the NoisyState varying.

We hypothesized that, in the Frozen scenario, a surprisal-based method, like ICM, would sample the NoisyState but also widely explore the OriginalState, as the gravity normally pushing down the car is frozen when the agent is distracted by the noisy action, representing no impediment to exploration. In practice, we see that ICM's average performance on the Frozen problem is better than on the Evolving setup but is still strongly limited by stochasticity.

Average state space coverage for several baselines is displayed in Figure 6. As also highlighted in the Figure's table, LBS remains the best performing method in both the variants of the stochastic environment, being strongly robust to NoisyTV-like stochasticity. Disagreement and VIME also show to be resilient to stochasticity, though exploring less than LBS. Both ICM and RND's performance are strongly undermined by the randomness in the task. The tabular results also show that LBS is the method that least reduced its exploration performance, compared to the original non-stochastic Mountain Car experiment and that ICM is the method that suffered the presence of noise the most.



	State Space Coverage (%)			Reduction (%)	
	NoStoch	Frozen	Evolving	Frozen	Evolving
LBS (ours)	**91.75**	**82.38**	**87.0**	**-10.21**	**-5.18**
Disagreement	90.88	68.75	77.38	-24.35	-14.85
ICM	**91.75**	28.75	17.25	-68.66	-81.20
RND	67.38	30.75	28.38	-54.36	-57.88
VIME	89.57	69.38	84.12	-22.54	-6.08
Random	15.0	11.5	12.62	-23.33	-15.87

Figure 6: **Stochastic Mountain Car.** On the left, training curves on the two variants of the stochastic Mountain Car problem are displayed, showing the average state space coverage over eight random seeds (standard deviations in shade). On the right, the table compares the final performance with the original non-stochastic environment, highlighting the reductions in performance.

Table 1: We summarize and compare several exploration methods, highlighting similarities and differences.

Algorithm	Objective	Model Loss	Distributions	Ensemble	Episodic
ICM	$-\log p(\phi_{t+1}\|\phi_t, a_t)$	Forward + Inverse Dynamics	✗	✗	✗
RND	$-\log p(\phi_{t+1}\|s_{t+1})$	Knowledge Distillation	✗	✗	✗
VIME	$D_{\mathrm{KL}}[q(\theta'\|s_t, a_t)\\|q(\theta\|s_t, a_t)]$	ELBO (variational weights)	✓(weights θ)	✗	✗
Disagreement	$\approx IG(s_{t+1}; \theta_{1:k}\|s_t, a_t)$	Forward Dynamics (Ensemble)	✗	✓	✗
Plan2Explore	$\approx IG(h_{t+1}; \theta_{1:k}\|s_t, a_t)$	Forward Dynamics (Ensemble)	✗	✓	✗
RIDE	$\\|\phi_{t+1} - \phi_t\\|_2/\sqrt{N_{ep}(s_{t+1})}$	Forward + Inverse Dynamics	✗	✗	✓
NGU	$\approx \alpha_t/\sqrt{N_{ep}(\phi_{t+1})}$	Inverse Dynamics	✗	✗	✓
LBS (ours)	$IG(z_{t+1}; s_{t+1}\|s_t, a_t)$	ELBO (variational latent)	✓(latent z)	✗	✗

$\phi = f(s)$: features; θ: model parameters; θ': θ after model update; h: hidden state of a RNN (part of the model); IG: information gain; k: ensemble models; $N_{ep}(s)$: episodic (pseudo)count of visits to s; α_t: normalized RND's objective; z: latent variable in the model.

5 Related Work

In Table 1, we compare LBS to all the methods we benchmark against (both in main text and Appendix).

Reinforcement Learning. Value-based methods in RL use the Q-value function to choose the best action in discrete settings (Mnih et al. 2015; Hessel et al. 2018). However, the Q-value approach cannot scale well to continuous environments. Policy Optimization techniques solve this by directly optimizing the policy, either learning online, using samples collected from the policy (Schulman et al. 2015, 2017), or offline, reusing the experience stored in a replay buffer (Lillicrap et al. 2016; Haarnoja et al. 2018).

Latent Dynamics. In complex environments, the use of latent dynamics models has proven successful for control and long-term planning, either by using VAEs to model locally-linear latent states (Watter et al. 2015), or by using recurrent world models in POMDPs (Buesing et al. 2018; Hafner et al. 2019, 2020).

Intrinsic Motivation. Several exploration strategies use a dynamics model to provide intrinsic rewards (Pathak et al. 2017; Burda et al. 2019b; Houthooft et al. 2016; Pathak, Gandhi, and Gupta 2019; Kim et al. 2019). Latent variable dynamics have also been studied for exploration (Bai et al. 2020; Bucher et al. 2019; Tao, Francois-Lavet, and Pineau 2020). Maximum entropy in the state representation has been used for exploration, through random encoders, in RE3 (Seo et al. 2021), and prototypical representations, in ProtoRL (Yarats et al. 2021).

Alternative approaches to modelling the environment's dynamics are based on pseudo-counts (Bellemare et al. 2016; Ostrovski et al. 2017; Tang et al. 2017), which use density estimations techniques to explore less seen areas of the environment, Randomized Prior Functions (Osband, Aslanides, and Cassirer 2018), applying statistical bootstrapping and ensembles to the Q-value function model, or Noisy Nets (Fortunato et al. 2018), applying noise to the value-function network's layers.

Some methods combine model-based intrinsic motivation with pseudo-counts, such as RIDE (Raileanu and Rocktäschel 2020), which rewards the agent with for transitions that have an impact on the state representation, and NGU (Badia et al. 2020b), which modulates a pseudo-count bonus with the intrinsic rewards provided by RND. Remarkably, combining NGU with an adaptive exploration strategy over the agent's lifetime led Agent57 to outperform human performance in all Atari games (Badia et al. 2020a).

Planning Exploration. Recent breakthroughs concerning exploration in RL have also focused on using the learned environment dynamics to plan to explore. This is the case in (Shyam, Jaśkowski, and Gomez 2019) and (Ratzlaff et al. 2020), where they use imaginary rollouts from their dynamics models to plan exploratory behaviors, and (Sekar et al. 2020), where they combine a model-based planner in latent space (Hafner et al. 2020) with the Disagreement exploration strategy (Pathak, Gandhi, and Gupta 2019).

6 Discussion

In this work, we introduced LBS, a novel approach that uses Bayesian surprise in latent space to provide intrinsic rewards for exploration in RL. Our method has proven successful in several continuous-control and discrete-action settings, providing reliable and efficient exploration performance in all the experimental domains, and showing robustness to stochasticity in the dynamics of the environment.

The experiments in low-dimensional continuous-control tasks, where we evaluate the coverage of the environment's state space, have shown that our method provides more in-depth exploration than other methods. LBS provided the most effective and efficient exploration in the Mountain Car and Ant Maze tasks, and strongly outperformed all methods in the more complex HalfCheetah task. Comparing LBS to VIME and Disagreement, we showed that Bayesian surprise in a latent representional space outperforms information gain in parameter space.

In the arcade games results, we showed that LBS works well in high-dimensional settings. By performing best in 5 out of 9 games, compared to several surprisal-based baselines, we demonstrate that the curiosity signal of LBS, based on Bayesian surprise, generally works better than surprisal.

We also tested LBS to be resilient to stochasticity in the dynamics, both qualitatively and quantitatively. While other methods based on the information gained in parameter space also showed to be robust in the stochastic settings, the exploration performance of LBS are unmatched in both variants of stochastic Mountain Car. We believe stochasticity is an important limitation that affects several exploration methods and future work should focus on understanding to which extent limitations apply and how to overcome them.

Acknowledgements

This research received funding from the Flemish Government (AI Research Program). Ozan Catal is funded by a Ph.D. grant of the Flanders Research Foundation (FWO).

References

Achiam, J.; and Sastry, S. 2017. Surprise-Based Intrinsic Motivation for Deep Reinforcement Learning. arXiv:1703.01732.

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete Problems in AI Safety. arXiv:1606.06565.

Badia, A. P.; Piot, B.; Kapturowski, S.; Sprechmann, P.; Vitvitskyi, A.; Guo, D.; and Blundell, C. 2020a. Agent57: Outperforming the Atari Human Benchmark. arXiv:2003.13350.

Badia, A. P.; Sprechmann, P.; Vitvitskyi, A.; Guo, Z. D.; Piot, B.; Kapturowski, S.; Tieleman, O.; Arjovsky, M.; Pritzel, A.; Bolt, A.; and Blundell, C. 2020b. Never Give Up: Learning Directed Exploration Strategies. In *8th International Conference on Learning Representations, ICLR 2020*.

Bai, C.; Liu, P.; Wang, Z.; Liu, K.; Wang, L.; and Zhao, Y. 2020. Variational Dynamic for Self-Supervised Exploration in Deep Reinforcement Learning. arXiv:2010.08755.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *J. Artif. Int. Res.*, 47(1): 253–279.

Bellemare, M. G.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016. Unifying Count-Based Exploration and Intrinsic Motivation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 1479–1487.

Bishop, C. M. 1997. Bayesian Neural Networks. *Journal of the Brazilian Computer Society*, 4.

Bucher, B.; Arapin, A.; Sekar, R.; Duan, F.; Badger, M.; Daniilidis, K.; and Rybkin, O. 2019. Perception-Driven Curiosity with Bayesian Surprise. *RSS Workshop on Combining Learning and Reasoning for Human-Level Robot Intelligence*.

Buesing, L.; Weber, T.; Racaniere, S.; Eslami, S. M. A.; Rezende, D.; Reichert, D. P.; Viola, F.; Besse, F.; Gregor, K.; Hassabis, D.; and Wierstra, D. 2018. Learning and Querying Fast Generative Models for Reinforcement Learning. arXiv:1802.03006.

Burda, Y.; Edwards, H.; Pathak, D.; Storkey, A. J.; Darrell, T.; and Efros, A. A. 2019a. Large-Scale Study of Curiosity-Driven Learning. In *7th International Conference on Learning Representations, ICLR*.

Burda, Y.; Edwards, H.; Storkey, A. J.; and Klimov, O. 2019b. Exploration by random network distillation. In *7th International Conference on Learning Representations, ICLR 2019*.

Clark, J.; and Amodei, D. 2016. Faulty Reward Functions in the Wild.

Fortunato, M.; Azar, M. G.; Piot, B.; Menick, J.; Osband, I.; Graves, A.; Mnih, V.; Munos, R.; Hassabis, D.; Pietquin, O.; Blundell, C.; and Legg, S. 2018. Noisy Networks for Exploration. In *Proceedings of the International Conference on Representation Learning (ICLR 2018)*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1861–1870.

Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019. Learning Latent Dynamics for Planning from Pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 2555–2565.

Hafner, D.; Lillicrap, T.; Norouzi, M.; and Ba, J. 2021. Mastering Atari with Discrete World Models. arXiv:2010.02193.

Hafner, D.; Lillicrap, T. P.; Ba, J.; and Norouzi, M. 2020. Dream to Control: Learning Behaviors by Latent Imagination. In *8th International Conference on Learning Representations, ICLR 2020*.

Hessel, M.; Modayil, J.; van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M. G.; and Silver, D. 2018. Rainbow: Combining Improvements in Deep Reinforcement Learning. In *AAAI*, 3215–3222.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *5th International Conference on Learning Representations, ICLR 2017*.

Houthooft, R.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; and Abbeel, P. 2016. VIME: Variational Information Maximizing Exploration. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 1117–1125.

Itti, L.; and Baldi, P. 2006. Bayesian Surprise Attracts Human Attention. In *Advances in Neural Information Processing Systems*, volume 18, 547–554.

Kim, H.; Kim, J.; Jeong, Y.; Levine, S.; and Song, H. O. 2019. EMI: Exploration with Mutual Information. arXiv:1810.01176.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Krakovna, V.; et al. 2020. Specification gaming: the flip side of AI ingenuity.

Larson, R. W.; and Rusk, N. 2011. Chapter 5 - Intrinsic Motivation and Positive Development. In *Positive Youth Development*, volume 41 of *Advances in Child Development and Behavior*, 89–130. JAI.

LeCun, Y.; Jackel, L. D.; Bottou, L.; Cortes, C.; Denker, J. S.; Drucker, H.; Guyon, I.; Muller, U. A.; Sackinger, E.; Simard, P.; et al. 1995. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261(276): 2.

Legault, L. 2016. *Intrinsic and Extrinsic Motivation*, 1–4. Cham: Springer International Publishing.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In *ICLR*.

Matusch, B.; Ba, J.; and Hafner, D. 2020. Evaluating Agents without Rewards. arXiv:2012.11538.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Moore, A. 1990. Efficient memory-based learning for robot control.

OpenAI; Akkaya, I.; Andrychowicz, M.; Chociej, M.; Litwin, M.; McGrew, B.; Petron, A.; Paino, A.; Plappert, M.; Powell, G.; Ribas, R.; Schneider, J.; Tezak, N.; Tworek, J.; Welinder, P.; Weng, L.; Yuan, Q.; Zaremba, W.; and Zhang, L. 2019. Solving Rubik's Cube with a Robot Hand. arXiv:1910.07113.

Osband, I.; Aslanides, J.; and Cassirer, A. 2018. Randomized Prior Functions for Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 31, 8617–8629.

Ostrovski, G.; Bellemare, M. G.; van den Oord, A.; and Munos, R. 2017. Count-Based Exploration with Neural Density Models. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 2721–2730.

Oudeyer, P.; 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 *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 2778–2787.

Pathak, D.; Gandhi, D.; and Gupta, A. 2019. Self-Supervised Exploration via Disagreement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5062–5071.

Popov, I.; et al. 2017. Data-efficient Deep Reinforcement Learning for Dexterous Manipulation. arXiv:1704.03073.

Raileanu, R.; and Rocktäschel, T. 2020. RIDE: Rewarding Impact-Driven Exploration for Procedurally-Generated Environments. arXiv:2002.12292.

Ratzlaff, N.; Bai, Q.; Fuxin, L.; and Xu, W. 2020. Implicit Generative Modeling for Efficient Exploration. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 7985–7995.

Schmidhuber, J. 1991. Curious model-building control systems. In *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, 1458–1463 vol.2.

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 *Anticipatory Behavior in Adaptive Learning Systems*, 48–76.

Schmidhuber, J. 2010. Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3): 230–247.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 1889–1897.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Sekar, R.; Rybkin, O.; Daniilidis, K.; Abbeel, P.; Hafner, D.; and Pathak, D. 2020. Planning to Explore via Self-Supervised World Models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 8583–8592.

Seo, Y.; Chen, L.; Shin, J.; Lee, H.; Abbeel, P.; and Lee, K. 2021. State Entropy Maximization with Random Encoders for Efficient Exploration. arXiv:2102.09430.

Shyam, P.; Jaśkowski, W.; and Gomez, F. 2019. Model-Based Active Exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5779–5788.

Tang, H.; Houthooft, R.; Foote, D.; Stooke, A.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; and Abbeel, P. 2017. #Exploration : a study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 1–18.

Tao, R. Y.; Francois-Lavet, V.; and Pineau, J. 2020. Novelty Search in Representational Space for Sample Efficient Exploration. In *Advances in Neural Information Processing Systems*, volume 33, 8114–8126.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; de Las Casas, D.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; Lillicrap, T.; and Riedmiller, M. 2018. DeepMind Control Suite. arXiv:1801.00690.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033.

Watter, M.; Springenberg, J. T.; Boedecker, J.; and Riedmiller, M. 2015. Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, 2746–2754.

Wauthier, S. T.; Mazzaglia, P.; Çatal, O.; Boom, C. D.; Verbelen, T.; and Dhoedt, B. 2021. A learning gap between neuroscience and reinforcement learning. arXiv:2104.10995.

Yarats, D.; Fergus, R.; Lazaric, A.; and Pinto, L. 2021. Reinforcement Learning with Prototypical Representations. arXiv:2102.11271.