Intrinsically Motivated Exploration for Data-Efficient MBRL

Ishani Ghose
Department of Computer Science
Rutgers University
New Jersey, NJ 08854
ig225@scarletmail.rutgers.edu

Abstract

In this work, we propose a new approach towards exploration during reinforcement learning and demonstrate that it can be used to improve the sample efficiency of Model-Based Reinforcement Learning (MBRL) algorithms. Unlike recent state of the art approaches toward exploration that use ensembles and can be heavy on the compute resources, our method uses a single forward model to compute the information gained from a transition. Specifically, we reward transitions found to be informative while training this forward model. Our method explores actively by leveraging planning in latent space to estimate future novelty. Eventually, it estimates the intrinsic reward from a transition as zero when the forward model does not learn from it. This intrinsic reward is used to optimize the agent's policy along with task-specific extrinsic rewards. We demonstrate a significant improvement in the performance of the RL agent on continuous control tasks with intrinsic rewards estimated by our method over the underlying MBRL algorithm and other existing approaches.

1 Introduction

Exploration is a crucial challenge in model-based learning, particularly because diverse data is required to train the model, often from high-dimensional input. A well-studied approach toward solving this challenge has been to use intrinsic rewards in addition to the task-rewards as learning signals to train the behavior policy in reinforcement learning algorithms. Many prior approaches build a predictive model that is trained alongside the world model of the MBRL algorithm while the model's prediction error is used as the intrinsic reward signal. Furthermore, intrinsic rewards provide feedback to the policy when extrinsic rewards are spare or absent[10].

Other existing formulations of intrinsic rewards include estimating the novelty from the state-visitation count [12] wherein the policy attempts to maximize the visits to the infrequently visited states, action entropy, information gained from the transition (s, a, r, s')[2, 3, 4, 8, 10] and empowerment[13, 14, 15]. These methods either attempt to reduce uncertainty in the internal state of the agent or increase coverage the state-space[32].

Model based deep RL [5, 6, 7, 17] usually involves learning a latent representation of the state space and dynamics, which together form a world model for the agent. The policy is trained through experience gained in the latent state space, shortening interacting with the environment. An important issue that persists in both model-based and model-free learning is sample complexity forming a significant hurdle in being able to deploy these algorithms in the real world. This makes any step towards improving the sample complexity of an algorithm particularly valuable. With recent advances, MBRL has demonstrated better performance in terms of sample efficiency over model-free methods.

Preprint. Under review.

MBRL algorithms learn latent states to encode information about the task, environment and even the reward function in some cases. These states filter out aleatoric uncertainty. Aleatoric uncertainty is the inherent stochastic nature of the environment that cannot be learnt from gathering more data. This circumvents issues that arise from noisy-environment-observations (the noisy TV problem) etc. as explicated in [4]. Furthermore, the dynamics of the world model provides a supervision signal from which a separate transition model can be trained alongside the world model. Our method uses this transition model to estimate the novelty of a transition.

Our contributions are as follows: We propose a new active learning approach towards exploration. Our method trains policies in a world model, to collect data in the real environment guided by the information gained from a transition in the world model. In comparison to recent active learning approaches for exploration that leverage ensembles, our method demonstrates significant improvement in sample efficiency while being more compute-efficient.

The paper is structured as follows. In Section 2 we describe the basic concepts relevant to our method. In Section 3 we present the formulation of our method's intrinsic reward and its relation to active learning. In Section 4 we demonstrate the effectiveness of our approach across a variety of continuous control tasks. In Section 5 we discuss Related Work, concluding in Section 6 where we point out links to possible future work.

2 Background

2.1 Model-Based Reinforcement Learning

MBRL algorithms aim to learn a world model from data observed in the real environment. The behavior policy to solve a task is usually trained in this world model. The necessity to learn a precise model of the environment plays an important role in determining which data samples should be collected while minimizing interaction with the environment.

Many MBRL algorithms train a greedy policy and use it to collect data to further improve the world model. This introduces redundancy among the state transitions observed in the real environment and reduces sample efficiency. Addressing this issue, our method focuses on collecting data that will improve the world model.

We use the Dreamer model as the underlying MBRL to demonstrate the efficacy of our method. The Dreamer algorithm learns a world model that contains the following components: Recurrent model: $h_t = f_{\phi'}(h_{t-1}, s_{t-1}, a_{t-1})$, Representation model: $s_t \sim p_{\phi'}(s_t|h_t, o_t)$ Transition predictor: $\hat{s_t} \sim p_{\phi'}(\hat{s_t}|h_t)$ Image predictor: $\hat{x_t} \sim p_{\phi'}(\hat{x_t}|h_t, s_t)$ Reward predictor: $\hat{r_t} \sim p_{\phi'}(\hat{r_t}|h_t, s_t)$ Discount predictor: $\hat{Y_t} \sim p_{\phi'}(\hat{Y_t}|h_t, s_t)$[5].

In order to train the world model, Dreamer samples batches of episodes from the real environment. The policy is trained by rolling out sequences of fixed length in the world model. The latent state transitions and predicted rewards are used to update the policy π_θ using an Actor Critic algorithm. A signal indicating whether the current state is terminal is predicted by the Discount predictor.

We use the stochastic component of the latent state = s_t as the supervision signal for the forward model q_ϕ in Figure 1.

2.2 Active Learning

Active learning deals with the problem of choosing a subset of data to be labeled by an oracle from a collection of unlabeled examples such that the model M is most improved.

In the Active Learning setting, we do not have access to the labels y_i for each x_i. Methods in active learning usually iterate over the available $(x_i)_{i=1}^n$ and determines which subset $(x_i^{`})_{i=1}^k$ will result in greatest improvement of the performance of M. The chosen subset is used to train M. our method is inspired by the idea that the data $(x_i^{`})_{i=1}^k$ chosen can be cast as the problem of selecting data that maximizes the expected information gain[1]. [1] is built upon the work in [41], which formalized the amount of information obtained from an experiment.

However, in Reinforcement Learning, we do not have access to a static data set. This requires the active learning method to adapt while gaining experience in the environment such that the data it

collects will significantly improve the model M. Specifically, the data observed in the environment is a series of (s_t, a, r, s^{t+1}) tuples where S is a POMDP[44] state and M is a model of policy. In MBRL, M can also include the dynamics model being learnt. The policy used to sample data from the environment can be trained such that the agent will observe transitions in the future that it expects to improve M the most.

Concretely, our method generates an intrinsic reward for transitions that the forward model cannot predict well in the latent space learned by the dreamer model and encourages the agent to perform these transitions in the real environment to learn them better. Instead of estimating disagreement among an ensemble of models [2, 3, 4], another active learning approach, a single forward model that starts a different initial state from the world model, is used to identify uncertainty. Thus the agent is encouraged to explore parts of the state space that it has not learnt well.

2.3 Online Reinforcement Learning

Online Reinforcement learning methods involve interacting with the environment to gather more experience while improving it's behaviour policy. In this setting, the RL agent is able to collect data specifically suitable for the task it is attempting to solve, while continually updating its policy. Elaborating this, the agent collects data with the current version of it's policy - π_t. On collecting a sequence of data $o_1, a_1, r_1, o_2, a_2, r_2, ...o_n$, the agent receives feedback about it's decisions or policy. Based on this feedback, the agent updates it's strategy of collecting data in the future.

In our approach, the agent optimizes its policy to maximize the task-rewards it collects from the environment while it also executes actions whose outcomes the agent cannot predict well. The exploratory intrinsic reward is a part of the feedback received by the agent to update it's policy.

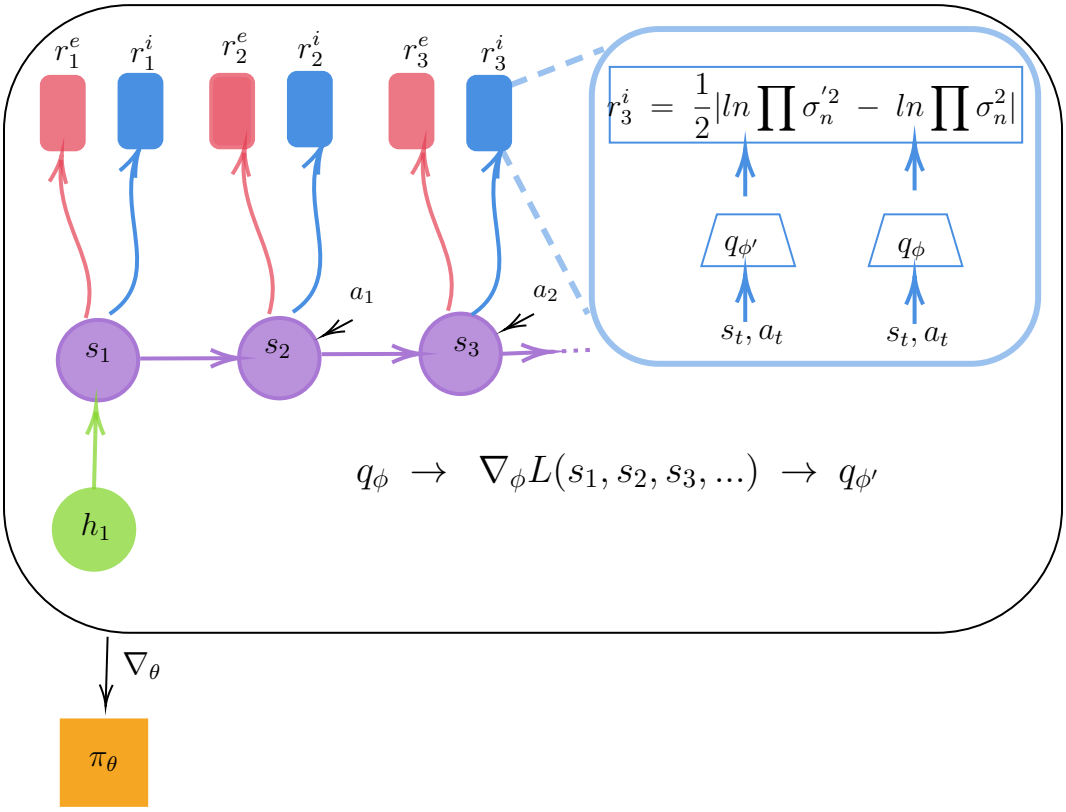

Figure 1: Illustration of Variance Based Exploration. The intrinsic reward for a state transition is computed by first, finding the log-product of variances of the distribution over the next state along each dimension of the state space: V_1. This distribution is modeled by the encoder q_ϕ. On observing a sequence of transitions, the encoder is updated with the derivative of the log-likelihood loss. Next, we compute the log-product of variances of the updated distribution over the next state in a similar way = V_2. The absolute difference of these products, $\frac{1}{2}|V_2 - V_1|$ is returned as the intrinsic reward.

About the latent dynamics: Each observation o_t at time t is first encoded into features h_t which is then used to infer a latent state s_t. A recurrent model predicts the next state s_{t+1} conditioned on h_t, s_t, a_t.

3 Method

3.1 Predictive model

Usually, MBRL algorithms learn a model of the environment's dynamics to train a policy without interacting with the environment for either the whole[5, 6] or a part of the training period[20]. This makes the accuracy of the dynamics model crucial for providing a true representation of the real environment to train the behavior policy in.

In our method, a forward predictive model, separate from the world model, learns a distribution of the next state conditioned on the current state and action. This model is essentially a dynamics model, trained in parallel to the world model. The initial state of this model differs from the dynamics model of the underlying MBRL's world model.

Forward predictor: $\hat{S_{t+1}} \sim \mathcal{N}(\mu, \sigma)$ where μ and σ are modeled by q_ϕ.

$\mathcal{N}(\mu, \sigma)$ is a multivariate normal distributions with independent variables. This predictor is trained in a supervised manner. For each input - (s_t, a_t), the model is trained by the log-likelihood loss of s_{t+1}. While the predictions of the forward model is not used directly, it eventually learns the transition function of the environment.

3.2 Estimating the intrinsic reward: variance based exploration

Our work is inspired by the curiosity driven objective in [2], where the information gained from each transition - estimated from ensemble disagreement - is used as an intrinsic reward. However, we estimate the information gained from an observation as the change in entropy of the distribution of the world parameters on being updated by that observation[1].

Concretely, we measure change in entropy of the parameters as

$$H(P^{N+1}(W)) - H(P^N(W)) \tag{1}$$

where $P^{N+1}(W)$ is the probability distribution of W on being updated with the $N + 1^{th}$ observation.

If we consider the random variable S_{t+1} i.e. the next state predicted by q_ϕ as world parameters W in equation (1) then we essentially need to calculate the change in differential entropy of the distribution of S_{t+1} to estimate the information gained about S_{t+1} from an observation. We take each observation to be a sequence of transitions executed by the policy π_θ in latent space. From a sequence of k transitions executed in latent space we get k tuples; $(s_t, a_t, \hat{r_t}, s_{t+1})_{t=1}^{k-1}$ which are used to update q_ϕ.

It is well known that the differential entropy of a single variable Gaussian distribution is $\frac{1}{2} + ln(\sqrt{2\pi}\sigma)$. In the multi-variable case, when all the variables are independent, the differential entropy is [43]

$$H = \frac{n}{2} ln 2\pi e(\sigma_1^2 \sigma_2^2 ... \sigma_n^2)^{1/n} \tag{2}$$

We define S_{t+1} as a random variable with a multi-variable Gaussian probability density function modeled by q_ϕ. The change in entropy on observing the transitions is

$$H(P(S_{t+1})|q_{\phi'}) - H(P(S_{t+1})|q_\phi) = \frac{1}{2} ln \frac{\prod \sigma_i^{'2}}{\prod \sigma_i^2} \tag{3}$$

To estimate $H(S_{t+1}|q_{\phi'})$ we update the parameters of q_ϕ with the log-likelihood loss on observing a sequence $s_1, s_2, ..., s_k$, in latent space:

$$q_{\phi'} \leftarrow q_\phi + \alpha \nabla_\phi log(p(s_2, s_3, ..., s_k)) \tag{4}$$

An alternative information measure is the cross-entropy between $P^N(W)$ and $P^{N+1}(W)$. [1] intuitively described the difference between these two measures of information: change in entropy

and cross entropy. The former quantifies how much the probability bubble P(W) shrinks and the latter measures how much the probability bubble shifts. In the latter case, even if certainty about W has not improved, we still learn something about W if the probability distribution of W has moved. [1] also shows us that in expectation, these two measures of difference are equal. Thus taking all changes in the entropy of P(W) as a signal of learning, we ensure that the intrinsic reward is positive. In other words, we take the absolute value of r_t^i

$$\frac{1}{2}|ln \prod \sigma_i^{'2} - ln \prod \sigma_i^2| \tag{5}$$

as the formulation for the intrinsic reward used in our method. We refer to this as Variance Based Exploration.

We do not use a hyper-parameter to trade off between exploration and exploitation. Over time, as the agent learns an increasingly accurate dynamics model, the predictions of the forward model and the dynamics model of the MBRL converge and the intrinsic rewards become negligible or 0 alleviating the need of a decay schedule.

Algorithm 1: Variance based intrinsic reward estimation

Initialize parameters of q_ϕ;
Input : Batch of sequences of length *k* executed in latent space
Output : Reward values for a batch of sequences of length *k*
$L_\phi \leftarrow logq_\phi(s_2, s_3, ..s_k)$;
$\phi^{'} \leftarrow \phi + \alpha \nabla L_\phi$;
for $t = 1$ *to* k **do**
  $V \leftarrow \ln\prod_{i=1}^{n} \sigma_i$ where σ_i is the variance along the i^{th} dimension of the Gaussian modeled by q_ϕ;
  $V^{'} \leftarrow \ln\prod_{i=1}^{n} \sigma_{i}^{'}$ where $\sigma_i^{'}$ is the variance along the i^{th} dimension of the Gaussian modeled by $q_{\phi'}$;
  $r_t^i \leftarrow \frac{1}{2}|V^{'} - V|$;
  $r_t^{total} \leftarrow r_t^e + r_t^i$;
end

3.3 Efficient exploration

In order to train a policy that performs the task well in the real environment while limiting interaction with the environment as much as possible, in the MBRL setting, we identify that learning an accurate and complete world model is crucial. Our method actively executes transitions in the real environment which the latent dynamics model is poor at modeling. As a result, the RL agent collects experience in a way that accelerates training the world model which in turn aids training a policy that performs well in the real environment.

Furthermore, collecting experience using a policy that learns from both extrinsic and intrinsic rewards is more efficient for solving the task at hand as it spends less time visiting regions of the state space irrelevant for solving the task while exploring.

4 Experiments

The primary goal of out experiments is to determine if the intrinsic reward for exploration computed by our method for MBRL makes the underlying MBRL algorithm more efficient. We also compare our method to 2 state-of-the-art exploration methods - Planning to explore[2] and Model based active exploration[3].

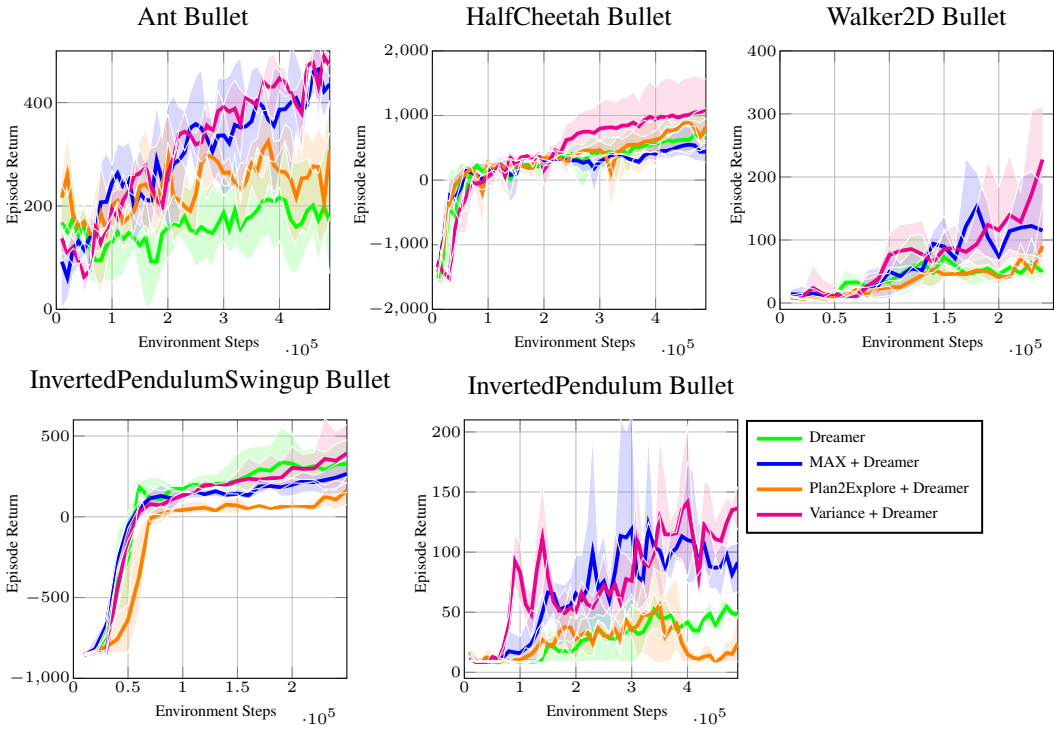

Figure 2: The performance of Dreamer and Dreamer + additional information gain based intrinsic rewards on 5 PyBullet continuous control tasks. The solid line and shaded region show the mean and deviation of the episode returns respectively. The variance based intrinsic reward is found to improve the performance of Dreamer consistently across the 5 tasks. Furthermore, Variance based exploration performs at par or better than MAX and Plan2Explore.

4.1 Environment

We used the PyBullet continuous control tasks by OpenAI[42] which are slightly harder versions of the tasks available in OpenAI Gym. These tasks are also more challenging than the DeepMind Control Suite tasks, because the episodes terminate on failure (e.g., when the walker falls over)[7].

Specifically, we demonstrate our results using Ant, Half Cheetah, Inverted Pendulum Swingup, Inverted Pendulum and Walker2D Bullet for the first 250k to 500k environment steps. For all experiments, we down sample the original pixel input from the environment and crop it to a window of 64x64x3 from the centre. The Dreamer agent repeats each action twice for all tasks. We used 3 seeds to run each experiment and evaluated the policy every 10^3 steps.The deviation from the mean is shown by the shaded region.

4.2 Baselines

We chose the Dreamer model[5] as the underlying MBRL algorithm as it is competitive with the state-of-the-art MBRL algorithms in terms of sample efficiency. The Dreamer model is able to simulate thousands of sequences in parallel on a single GPU while training the policy. Our experiments show a significant increase in task returns on using the variance based intrinsic reward formulated in section 3.2 when compared to the performance of Dreamer without intrinsic rewards.

Additionally, we also compare the our proposed method of exploration to two state-of-the-art self-supervised methods of exploration : Plan2Explore [2], Model-based Active Exploration [3], using the formulation of intrinsic rewards in these methods. Both, Plan2Explore and MAX use ensembles to measure uncertainty about the dynamics model.

Table 1: Compute-efficiency of Varaince Based Exploration. Here b_1 and b_2 are the batch sizes during policy training and model training respectively. M = number of add,multiplies during a forward pass

Method	FLOPS-Policy training	Flops-World model training
VBE	6Mb_1	6Mb_2
MAX	6Mb_1	18Mb_2
Plan2Explore	10Mb_1	30Mb_2

Like Variance Based Exploration, Plan2Explore and MAX eventually estimate negligible intrinsic rewards as the ensemble comes to a consensus and the dynamics model becomes accurate. Thus these methods do not require a hyper-parameter to trade off between exploration and exploitation as well.

4.3 Comparative evaluation

Variance based exploration consistently improves the sample-efficiency of Dreamer on multiple tasks. In particular, we observe over 100% improvement in the Walker2D, Ant and Inverted Pendulum Bullet tasks. Although MAX shows a similar extent of improvement for some tasks, it deteriorates the performance of Dreamer on Half Cheetah and Inverted Pendulum Swingup and Plan2Explore performs poorly as an exploratory bonus on most tasks.

4.4 Implementation

Here we elaborate on the architecture forward predictor used by our method and the ensemble used for the comparative methods - MAX and Plan2Explore. The forward predictor q_ϕ is implemented as a neural network of 4 layers of 400 units each, elu activation, which is used to transform the input (s_t, a_t) to the mean and standard deviation of the Gaussian distribution. The variables of the Gaussian are sampled independently.

The policy is trained by executing trajectories of fixed length in the latent space. Our method estimates an intrinsic reward for each of these transitions. Instead of updating q_ϕ separately for each transition, all transitions are used to update q_ϕ in a batch. Thus for each trajectory we perform one forward and one backward pass across the network.

An ensemble of networks with architecture identical to the one described above is used to implement MAX and Plan2Explore. These methods perform only a forward pass across the models while computing intrinsic rewards for each trajectory. In their original papers, MAX was implemented with an ensemble of 3 and Plan2Explore was implemented with an ensemble of 5 forward models.

4.5 Compute efficiency

A common methodology to compare compute efficiency when the exact architecture of the model is known, is the number of FLOPS [28] i.e. the number of addition and multiplication operations. A precise calculation is explained is [28]

$$2M(F + 2B)be \tag{6}$$

where M = number of add,multiplies during the forward pass, F is the number of forward passes, B is the number of backward passes, b is the batch size and e is the number of epochs. For our method, F=1 and B=1.

The bulk of the compute load is generated during the forward and backward passes, and serves as the basis for comparing the compute efficiency of our method with methods that use ensembles. During policy training in latent space, our method and MAX(F=3,B=0) executes the same number of FLOPS while Plan2Explore(F=5,B=0) executes 1.6x more FLOPS. In the real environment, for b_2 transitions sampled to train the model, our method executes 6Mb_2 flops while MAX(F=3,B=3) executes 3x more FLOPS and Plan2Explore (F=5,B=5) executes 5x FLOPS. Therefore, our method performs is more compute-efficient.

5 Related work

Recently, Model-based reinforcement learning[23, 21, 7, 18, 17] has seen large improvements in performance, this success being demonstrated using Atari, DM Control and Open AI gym benchmarks. Furthermore, Model-based reinforcement learning agents have been found to be more sample efficient than model free reinforcement learning while performing at par with model-free agents. Particularly, [22, 19, 20, 5, 6] improve sample efficiency by planning in latent space while improving the world model. Most of these model based agents learn a rich latent representation of the observations from the environment while learning the dynamics model. Our method can leverage this latent representation to estimate novelty, circumventing issues arising from uncertainty that cannot be controlled by the agent. For example, a noisy television[4].

Exploration plays a crucial role in the efficiency of reinforcement learning algorithms. Earlier approaches focused on tabular settings, estimating exploration bonuses based on state visitation counts while providing a theoretical perspective[33]. Bayesian approaches[45] were also studied but these methods do not scale to continuous, high-dimensional spaces such as pixel input. Some of these early ideas were then scaled to high dimensional data via pseudo-count state visitation[12].

A large number of prior works estimate novelty using predictive models. We see that Predictive errors about different aspects of the agent's model with respect to the state of the environment have been found to be effective. [2, 4] use the predictive error in the transition as the intrinsic reward. [8] predicts actions through inverse learning to encode useful features into the representations fed into the one-step forward predictive model. Our predictive forward model is used to identify missing knowledge about the dynamics of the environment from the world model. [36] performed a comparative study among different feature spaces: plain pixels, random features, inverse dynamics and Variational Autoencoder (VAE). [2, 3, 4, 35] use an ensemble of predictive models to estimate the uncertainty about the transitions that the agent has not learned well[9], [34] uses an ensemble of Q functions with randomized priors to predict the posterior Value function.

[11] estimates uncertainty using the parameters of a Bayesian Neural network to estimate novelty of the state space by calculating the cross entropy between the distributions of the weights of the Bayesian Neural network before and after observing a transition. However, this method was not evaluated on high dimensional environments. Our method estimates change in entropy of the distribution being modeled by a neural network and scales to image observations from complex environments.

Some approaches have used formulations of empowerment which aims to maximize the influence of the agent on its environment[13]. A model can be trained to predict the empowerment of a state[14]. This estimate can then be used as an intrinsic reward or as the sole exploratory objective. Empowerment can also be maximized along with task rewards though the policy gradient algorithm[16] and maximized along with information gain[15].

[28] estimates state entropy for exploration in a compute efficiency manner, without additional representation learning to improve the sample efficiency of Model-Based and Model-Free algorithms. [37] and [32] explicitly perform active learning to guide exploration. [2, 3, 4, 35] are also active approaches towards MBRL. Similarly, our work aims to improve model accuracy during exploration. Other approaches are exploration guided by tasks and skills [24, 25, 26], goal oriented strategies[29, 30, 31, 27].

6 Conclusion

In this paper, we present Variance Based Exploration (VBE), a simple exploration method for Model-Based Reinforcement Learning. VBE actively seeks to collect information from the environment that is required to improve the dynamics model of the MBRL algorithm. VBE identifies learnable unknowns in the environment by estimating information gained on observing poorly learnt dynamics. We showed that our method encourages exploration and significantly improves sample efficiency on multiple continuous control tasks. Furthermore, our method performs better than some of the state-of-the-art ensemble based approaches towards exploration while being more compute-efficient.

Combining our method with orthogonal approaches towards improving sample efficiency such as image augmentation [38, 39, 40] are possible directions for future work. Incorporating task based exploration would further increase the functional capacity of the model.

6.1 Broader impact

The presented work focuses on the actively studied problem of exploration and sample efficiency in the field of Reinforcement Learning. This area is crucial for closing the gap between research and real-world conditions (lack of crafted reward signals, image-based observations). The eventual goal of these algorithms is to model autonomous and intelligent systems for solving various tasks in the real world. The application of such systems may have positive and negative impacts on society depending on the field in which they are deployed and the tasks they are employed to perform. Thus the usage and development of these systems should be considered carefully.

References

[1] David J. C. MacKay (1992) Information-Based Objective Functions for Active Data Selection *Computation and Neural Systems, California Institute of Technology 139-74*.

[2] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, Deepak Pathak Planning to Explore via Self-Supervised World Models *arXiv:2005.05960 2020*

[3] Pranav Shyam, Wojciech Jaśkowski, Faustino Gomez Model-Based Active Exploration *arXiv:1810.12162 2019*

[4] Deepak Pathak, Dhiraj Gandhi, Abhinav Gupta Self-Supervised Exploration via Disagreement *arXiv:1906.04161 2019*

[5] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, Jimmy Ba MASTERING ATARI WITH DISCRETE WORLD MODELS *arXiv:2010.02193 2020*

[6] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, Mohammad Norouzi Dream to Control: Learning Behaviors by Latent Imagination *arXiv:1811.04551v1 2019*

[7] A. X. Lee, A. Nagabandi, P. Abbeel, S. Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv:1907.00953, 2019*

[8] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, Trevor Darrell Curiosity-driven Exploration by Self-supervised Prediction *arXiv:1705.05363 2017*

[9] Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Neural Information Processing Systems, 2017*

[10] Aleksandr Ermolov, Nicu Sebe Latent World Models For Intrinsically Motivated Exploration *Neural Information Processing Systems, 2020*

[11] Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Vime: Variational information maximizing exploration *Neural Information Processing Systems, 2016*

[12] Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation *Neural Information Processing Systems, 2016*

[13] Felix Leibfried, Sergio Pascual-Díaz, Jordi Grau-Moya A Unified Bellman Optimality Principle Combining Reward Maximization and Empowerment *Neural Information Processing Systems, 2019*

[14] Shakir Mohamed, Danilo Jimenez Rezende Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning *In Neural Information Processing Systems 2016*

[15] Ildefons Magrans de Abril, Ryota Kanai A unified strategy for implementing curiosity and empowerment driven reinforcement learning *arXiv:1806.06505 2018*

[16] Karol Gregor, Danilo Jimenez Rezende, Daan Wierstra Variational Intrinsic Control *In International Conference of Learning Representations 2017*

[17] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, Henryk Michalewski Model-Based Reinforcement Learning for Atari *In International Conference of Learning Representations 2020*

[18] Oleh Rybkin, Chuning Zhu, Anusha Nagabandi, Kostas Daniilidis, Igor Mordatch, Sergey Levine Model-Based Reinforcement Learning via Latent-Space Collocation In Neural Information Processing Systems 2020

[19] Julian Schrittwieser * 1 Thomas Hubert * 1 Amol Mandhane 1 Mohammadamin Barekatain 1 Ioannis Antonoglou 1 David Silver / Online and Offline Reinforcement Learning by Planning with a Learned Model arXiv:2104.06294 2021

[20] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, David Silver Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model arXiv:1911.08265 2019

[21] Matteo Hessel, Ivo Danihelka, Fabio Viola, Arthur Guez, Simon Schmitt, Laurent Sifre, Theophane Weber, David Silver, Hado van Hasselt Muesli: Combining Improvements in Policy Optimization *arXiv:2104.06159 2021*

[22] Masashi Okada and Tadahiro Taniguchi Dreaming: Model-based Reinforcement Learning by Latent Imagination without Reconstruction *arXiv:2007.14535, 2020*

[23] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, Pieter Abbeel Model-Ensemble Trust-Region Policy Optimization *In International Conference of Learning Representations 2018*

[24] Kevin Xie, Homanga Bharadhwaj, Danijar Hafner, Animesh Garg, Florian Shkurti Latent Skill Planning for Exploration and Transfer *In International Conference on Learning Representations 2021*

[25] Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, Sergey Levine Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables*In International Conference on Machine Learning, 2019.*

[26] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, Karol Hausman Dynamics-Aware Unsupervised Discovery of Skills *In International Conference on Learning Representations 2020*

[27] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, Sergey Levine Visual Foresight: Model-Based Deep Reinforcement Learning for Vision-Based Robotic Control *arXiv:1812.00568, 2018.*

[28] Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, Kimin Lee State Entropy Maximization with Random Encoders for Efficient Exploration *arXiv:2102.09430, 2021*

[29] Colas, Cédric, Fournier, Pierre, Chetouani, Mohamed, Sigaud, Olivier, and Oudeyer, PierreYves Curious: intrinsically motivated modular multi-goal reinforcement learning. *In International Conference on Machine Learning, 2019.*

[30] Nair, Ashvin, Pong, Vitchyr, Dalal, Murtaza, Bahl, Shikhar, Lin, Steven, and Levine, Sergey Visual reinforcement learning with imagined goals. *In Advances in Neural Information Processing Systems, 2018*

[31] Pong, Vitchyr H, Dalal, Murtaza, Lin, Steven, Nair, Ashvin, Bahl, Shikhar, and Levine, Sergey Skew-fit: State-covering self-supervised reinforcement learning. *In International Conference on Machine Learning, 2020.*

[32] Philip Ball, Jack Parker-Holder, Aldo Pacchiano, Krzysztof Choromanski, Stephen Roberts Ready Policy One: World Building Through Active Learning *arXiv preprint arXiv:2002.02693, 2020*

[33] Strehl, A. and Littman, M. An analysis of model-based interval estimation for markov decision processes. . *In Proceedings of The 22nd International Conference On Machine Learning, pp.856–863. ACM, 2005*

[34] Ian Osband, John Aslanides, Albin Cassirer Randomized Prior Functions for Deep Reinforcement Learning *In Advances in Neural Information Processing Systems, pp. 8617–8629, 2018*

[35] Mikael Henaff Explicit Explore-Exploit Algorithms in Continuous State Spaces In Advances in Neural Information Processing Systems 32, pages 9372–9382. Curran Associates, Inc., 2019

[36] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, Alexei A. Efros Large-Scale Study of Curiosity-Driven Learning *In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019.*

[37] T. Akiyama, H. Hachiya, and M. Sugiyama Efficient exploration through active learning for value function approximation in reinforcement learning *Neural Networks, 23(5):639 – 648, 2010*

[38] Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, Aravind Srinivas Reinforcement Learning with Augmented Data *In Neural Information Processing Systems 2020*

[39] Ilya Kostrikov, Denis Yarats, Rob Fergus Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels *In International Conference on Learning Representations 2021*

[40] Aravind Srinivas, Michael Laskin, Pieter Abbeel CURL: Contrastive Unsupervised Representations for Reinforcement Learning *Proceedings of the 37th International Conference on Machine Learning, PMLR 119:5639-5650, 2020.*

[41] Lindley, D. V. On a measure of the information provided by an experiment. The Annals of Mathematical Statistics, pp. 986–1005, 1956.

[42] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019

[43] McEliece, R.J. l977. The Theory of Information and Coding: A Mathematical Framework for Communication. Addison-Wesley, Reading, Mass.

[44] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence, 101(1):99 – 134, 1998. ISSN 0004-3702*

[45] Kolter, Z. and Ng, A. Near-bayesian exploration in polynomial time. *In International Conference on Machine Learning, 2009*

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Section: Conclusions
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Broader impact
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] It is present in the supplemental material and Section 4.4

(b) Did you specify all the training details (e.g., data splits, hyper-parameters, how they were chosen)? [Yes] In the supplemental material and Section 4.4

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] The deviation across multiple seeds is shown by the shaded region in the graphs

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In the supplemental material and Section 4.4

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] In the supplemental material

(b) Did you mention the license of the assets? [Yes]

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We did not obtain data from a third person

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not conduct research with human subjects

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]