Active Exploration for Robotic Manipulation

Schneider, Tim; Belousov, Boris; Chalvatzaki, Georgia; Romeres, Diego; Jha, Devesh K.; Peters, Jan

TR2022-139 November 01, 2022

Abstract

Robotic manipulation stands as a largely un- solved problem despite significant advances in robotics and machine learning in recent years. One of the key challenges in manipulation is the exploration of the dynamics of the environment when there is continuous contact between the objects being manipulated. This paper proposes a model-based active exploration approach that enables efficient learning in sparse-reward robotic manipulation tasks. The proposed method estimates an information gain objective using an ensemble of probabilistic models and deploys model predictive control (MPC) to plan actions online that maximize the expected reward while also performing directed exploration. We evaluate our proposed algorithm in simulation and on a real robot, trained from scratch with our method, on a challenging ball pushing task on tilted tables, where the target ball position is not known to the agent a-priori. Our real-world robot experiment serves as a fundamental application of active exploration in model-based reinforcement learning of complex robotic manipulation tasks.

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2022

Active Exploration for Robotic Manipulation

Tim Schneider[1], Boris Belousov[1], Georgia Chalvatzaki[1], Diego Romeres[2], Devesh K. Jha[2] and Jan Peters[1]

***Abstract*— Robotic manipulation stands as a largely unsolved problem despite significant advances in robotics and machine learning in recent years. One of the key challenges in manipulation is the exploration of the dynamics of the environment when there is continuous contact between the objects being manipulated. This paper proposes a model-based active exploration approach that enables efficient learning in sparse-reward robotic manipulation tasks. The proposed method estimates an information gain objective using an ensemble of probabilistic models and deploys model predictive control (MPC) to plan actions online that maximize the expected reward while also performing directed exploration. We evaluate our proposed algorithm in simulation and on a real robot, trained from scratch with our method, on a challenging ball pushing task on tilted tables, where the target ball position is not known to the agent a-priori. Our real-world robot experiment serves as a fundamental application of active exploration in model-based reinforcement learning of complex robotic manipulation tasks. Project page `https://sites.google.com/view/aerm`.**

I. Introduction

A common view in cognitive science is that the evolution of dexterous manipulation capabilities was one of the major driving factors in the development of the human mind [1] and the success of humankind in general [2]. Performing manipulation is cognitively highly demanding, forcing the agent to reason not only about the impact of its actions on itself, but also on the environment. This inherent complexity leaves autonomous robotic manipulation a largely unsolved problem, despite significant advances in robotics and machine learning in the last decades [3].

One of the central challenges of manipulation is the uncertainty about the environment. When an object is manipulated, its physical properties are rarely known in advance. Instead, they must be inferred from observations and touch. To deal with such inference problems effectively, humans have developed various active haptic exploration strategies [4, 5].

Prominent approaches in robotic manipulation span from motion planning methods [6] to imitation [7] and reinforcement learning [8]. Motion planning usually suffers from ill-defined task descriptions that combined with uninformed prior trajectory distribution lead to suboptimal behaviors. Learning-based methods, on the other hand, overfit single solutions, and collapse to low-entropy behaviors that fail to generalize to unseen variations of the same task. We believe that for robots to reach human-level manipulation skills, they must *actively explore* and adapt to new instances of a task.

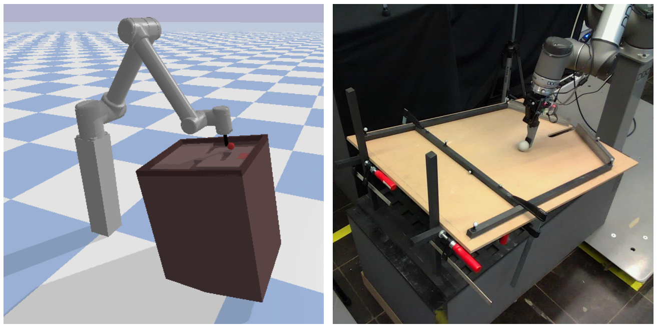


Fig. 1: Our *active exploration* strategy evaluated on a challenging *Tilted Pushing* task in simulation (left) and on the real robot (right). The agent needs to learn the dynamics of the task and identify a sparse reward model in order to bring the ball to a target location. The robot learns to solve the task online in the real setting from scratch.

We define *active exploration* as the directed search of the agent, during the learning process, for unvisited state-action pairs that would maximize the agent's performance. In this work, we draw inspiration from the Active Inference (AI) [9] field of studies to propose an active exploration framework for model-based reinforcement learning of challenging robotic manipulation tasks. In our problem formulation, we consider a fully observable environment with unknown world model whose dynamics we need to learn.

Unlike other approaches, that either couple model-free and model-based reinforcement learning to achieve better model learning through costly maximum entropy exploration [10–12], or introduce intrinsic signals related to the learned model variance to promote exploration [13, 14], we take an information-theoretic perspective, under the umbrella of AI. Starting from a common framework for model learning, namely, using an ensemble of neural networks that allow the estimation of epistemic uncertainty, we propose to use an information-seeking Model Predictive Control (MPC), that systematically explores in environments with sparse rewards.

Our information-seeking MPC tries to select actions that maximize the agent's information gain, balancing between highly exploratory actions when the dynamics model is unknown and the task performance when the agent has better confidence about its knowledge of the world model. We provide a thorough theoretical analysis regarding the implementation of our active exploration framework in model-

[1] Tim Schneider, Boris Belousov, Georgia Chalvatzaki, and Jan Peters are with the Intelligent Autonomous Systems Lab, Technical University of Darmstadt, 64289 Darmstadt, Germany, {`tim.schneider1`, `name.surname`}`@tu-darmstadt.de`

[2] Diego Romeres, Devesh K. Jha are with the Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA, {`romeres`, `jha`}`@merl.com`

This project has received funding from BMWSB ZukunftBau under grant Nr. 10.08.18.7-21.34 and the funded programs Aristotle by BMBF and iROSA by DFG. Calculations for this research were partially conducted on the Lichtenberg high performance computer of the TU Darmstadt.

based reinforcement learning. We evaluate our algorithmic contribution in simulated tasks of increased difficulty, where a 6 degrees of freedom (dof) robotic arm aims to solve a task where it has to push a ball on a tilted table to reach a goal position (Fig. 1). This task, though seemingly simple, introduces many challenges, as the robot has to balance the ball at the tip of its end-effector, and push the ball over a tilted table to reach an unknown for the agent goal. Moreover, we provide proof of concept results on a real robotic system that learns online, and we provide our insights regarding real-world robot learning with challenging dynamics.

To summarize our contributions in this work,

- we derive a novel algorithm for information-seeking model-based reinforcement learning,
- we investigate different measures of curiosity and information seeking strategies that can promote exploration for better dynamics model learning in simulated ball-pushing tasks with different difficulty levels, and
- we demonstrate a real-world execution of our actively exploring model-based controller for 7-dof manipulator learning to push a ball over a tilted table.

II. Related Work

This field of research attempts to solve Reinforcement Learning (RL) problems by predicting actions as an a posteriori estimate of trajectory rollouts given a prior distribution of actions [15]. Model-based RL learns the transition dynamics of the Markov decision process (MDP) and solves an optimal control problem, usually employing MPC. On the one hand, model-based RL is a promising way for learning reactive robot control strategies, benefiting from the integrated planner that can additionally integrate constraints about the problem. Applications of model-based RL can be found for robot in-hand manipulation [11], human-robot interaction [16, 17] and robot manipulation skill learning [18, 19].

On the other hand, learning the dynamics of the environment, especially in robotics, is a very challenging problem. Real-world dynamics, particularly the dynamics of complex manipulation tasks, like the task of interest in this paper, are characterized by multimodality that function approximators cannot easily capture. A prominent approach in the last years for model-based RL is the use of an ensemble of neural networks that can learn from different instances of the collected dataset to capture the epistemic and aleatoric uncertainty of the probabilistic model, thus, attempting to learn more accurate models [20]. These probabilistic ensembles are coupled with trajectory sampling (PETS) [20] via the cross-entropy method to estimate the best actions to apply to the environment. However, PETS suffers from the local exploration in the model rollouts that do not drive the agent to unknown states in the environment, therefore, it learns suboptimal dynamics models. Following methods attempt to couple model-free exploration with model learning, as in model-based policy optimization (MBPO) [10], but the purpose is to accelerate policy learning by utilizing model-based approximate samples. While this method is faster in terms of convergence compared to its pure model-free competitor Soft Actor Critic (SAC), it still needs an impractical amount of samples to learn a good control policy. MoPAC [11] improved over MBPO by employing model-predictive rollouts in the approximate MDP that is learned through the model, incentivizing the agent to explore areas of the state-space where model predictions are inefficient. A parallel line of works aims to learn world models from images, i.e., for observations, using a variational autoencoder to encode the image features and a probabilistic model to encode the dynamics with Cross-Entropy Method (CEM) for planning, like PlaNet [21]. Unlike PETS and PlaNet, which greedily select the actions that the CEM predicts to yield the highest reward, we propose a novel framework for model-based RL using an information-seeking objective in our MPC, that balances exploration-exploitation during learning, by promoting maximization of the information again when the model is still suboptimal and optimizes for the end-task once the agent is confident about its knowledge about the world dynamics.

Exploration for efficient model learning is an open research topic, with intrinsic motivation being a well-known method for exploration towards model learning [22, 23]. However, intrinsic rewards usually rely on hand-crafted metrics of learning progress, making them difficult to apply in a wide range of tasks [24]. In the era of the deep probabilistic ensembles, a self-supervised way of encouraging exploration of states that will improve model learning are methods based on ensemble disagreement, where the variance in the predictions of the different ensemble models is used as intrinsic reward for a model-free policy [14, 25, 26]. This policy, thus, learns to collect data in areas of the state-space where the disagreement between the predictions of the ensemble models is higher. However, advantages of these methods compared to MBPO, MoPAC and other model-based methods without explicit exploration bonus [12] are not well-established. Even if disagreement promotes exploration in early stages, it is more beneficial to vision-based RL settings, where the variance between the ensembles can be high due to the high image reconstruction errors, but its benefit to trajectory-based MDPs is incremental, as the ensembles are bounded by the statistics of the marginal state distribution of the dataset. Other methods for inducing curiosity in RL rely on prediction error [13], epistemic uncertainty [27] or state visitation counts [28] as reward signals during roll-outs. Contrary to these methods that introduce heuristic intrinsic rewards or rely on a model-free policy for model learning, our method employs a principled information-theoretic approach that can be traced back to early works on Bayesian experiment design [29]. Specifically, we utilize MPC to trade off between actions that yield a high expected extrinsic reward and actions that maximize the expected information gain of the observed states for our model in real time. The use of an expected information gain term inside MPC allows our agent to plan even beyond the space of states it has visited so far to obtain observations that it expects to be beneficial for model learning.

III. Active Exploration for Model Learning

We assume that the environment is fully observable, governed by unknown dynamics $P(x_\tau \,|\, x_{\tau-1}, a_\tau)$, and provides the agent with an a-priori unknown reward $P(r_\tau \,|\, x_\tau, a_\tau)$ in every time step. Here, $x_\tau \in \mathbb{R}^{N_x}$ denotes the state of the environment at time τ, $a_\tau \in \mathbb{R}^{N_a}$ the action the agent can take, and $r_\tau \in \mathbb{R}$ the reward it is receiving. The agent's objective is the maximization of the cumulative reward over a fixed time horizon of T discrete time steps, that is

$$\max_{\pi} \quad \mathbb{E}_{P(r_{1:T} \,|\, \pi)}\left[\sum_{\tau=1}^{T} r_\tau\right]$$

where $\pi(a_\tau \,|\, x_{\tau-1})$ is the agent's policy.

Since the real dynamics and reward distributions are unknown, the agent maintains approximations $p(x_\tau \,|\, a_\tau, s_{\tau-1}, \theta)$ and $p(r_\tau \,|\, x_\tau, a_\tau, \theta)$ of them, where θ are the model parameters. Hence, the agent assumes the following generative model of the environment:

$$p(x_{0:T}, a_{1:T}, r_{1:T}, \theta) = p_k(x_0)\, p(\theta) \prod_{\tau=1}^{T} \Big(p(r_\tau \,|\, x_\tau, a_\tau, \theta) \swarrow \\ p(x_\tau \,|\, x_{\tau-1}, a_\tau, \theta)\, \pi(a_\tau \,|\, x_{\tau-1}) \Big)$$

where $p_k(\theta)$ is the agent's belief over the correct model parameters in episode k.

Instead of greedily optimizing the expected reward directly, we propose to optimize the sum of the expected reward and an intrinsic term that encourages the agent to make observations informative w.r.t. its model. Hence, at time t, we choose π such that it optimizes the problem

$$\max_{\pi} \; \mathbb{E}_{p(r_{t+1:t+H} \,|\, \pi)}\left[\sum_{\tau=t+1}^{t+H} r_\tau\right] + \beta \mathrm{I}\left(\pi, s_t\right)$$

where $H \in \mathbb{N}$ is the planning horizon, $\beta \in \mathbb{R}$ a weighting factor, and $\mathrm{I}\left(\pi, s_t\right)$ the intrinsic term.

In this work, we choose the intrinsic term to be the information gain between the model parameters and the expected states and rewards:

$$\begin{aligned} \mathrm{I}\left(\pi, s_t\right) &\coloneqq \mathrm{MI}\left(\left(\mathbf{s}, \mathbf{r}\right), \theta \mid \pi, s_t\right) \\ &= \mathbb{E}_{p(\mathbf{s}, \mathbf{r} \,|\, \pi, s_t)}\left[D_{\mathrm{KL}}[p(\theta \,|\, \mathbf{s}, \mathbf{r}, \pi, s_t) \,\|\, p(\theta)]\right] \end{aligned}$$

where $\mathbf{s} \coloneqq s_{t+1:t+H}$ and $\mathbf{r} \coloneqq r_{t+1:t+H}$ is the sequence of states and rewards up to the planning horizon. Note that in the literature, the expected information gain is also known as Mutual Information (MI), which is why we denoted it accordingly in the equation above.

The expected information gain can be seen as a measure of how much the agent expects to learn about the environment by following policy π in state s_t. Specifically, this measure becomes maximal if the agent expects to make observations that will change its belief about the correct choice of model parameters drastically. Hence, an agent maximizing this term will be curious about its environment and explore it systematically, even in the total absence of extrinsic reward. In combination with the expected reward, we obtain an agent that is acting both information-seeking and goal-directed, with the trade-off being explicitly controlled by the weighting factor β.

The optimization of the objective can now be performed by any planner that is capable of handling continuous action spaces. In this work, we use a variant of the Cross-Entropy Method to find an open loop sequence of actions $a_{t+1:T}$ that maximizes the objective.

A. Approximation of the planning objective

A major challenge in computing the joint objective is that neither the expected reward nor the intrinsic term can be computed in closed form. While the expected reward can straightforwardly be approximated via Monte Carlo (MC) [20, 21], the intrinsic term is known to be notoriously difficult to compute [30–33]. Thus, instead of maximizing Mutual Information (MI) directly, many methods maximize a variational lower bound of it [33]. However, due to the high-dimensional nature of θ, these approaches are too expensive to be executed during planning in real time.

Hence, instead we propose to use a Nested Monte Carlo (NMC) estimator that reuses samples from the outer estimator in the inner estimator:

$$\begin{aligned} &\mathrm{MI}\left(\left(\mathbf{s}, \mathbf{r}\right), \theta \mid \pi, s_t\right) \\ &= \mathbb{E}_{p(\theta)}\left[\mathbb{E}_{p(\mathbf{s}, \mathbf{r} \,|\, \pi, s_t, \theta)}\left[\ln p(\mathbf{s}, \mathbf{r} \,|\, \pi, s_t, \theta) - \ln p(\mathbf{s}, \mathbf{r})\right]\right] \\ &\approx \underbrace{\frac{1}{n}\sum_{i=1}^{n}\Big(\ln p\big(s^i, r^i \,\big|\, \theta^i\big) - \ln \underbrace{\frac{1}{n}\sum_{\substack{k=1 \\ k \neq i}}^{n} p\big(s^i, r^i \,\big|\, \theta^k\big)}_{\text{inner estimator}}\Big)}_{\text{outer estimator}} \end{aligned}$$

where

$$\theta^i \sim p(\theta)\,, \quad \big(\mathbf{s}^i, \mathbf{r}^i\big) \sim p\big(\mathbf{s}, \mathbf{r} \,\big|\, \theta^i\big)\,, \quad \forall i \in \{1, \ldots, n\}\,.$$

Although using the same samples $\theta_1, \ldots, \theta_n$ in the inner estimator as in the outer estimator violates the i.i.d. assumption, we found this reuse of samples can substantially increase the sample efficiency.

B. Choice of model

We approximate the dynamics and the reward models with Gaussian distributions where the mean and covariance are given by neural networks with weights θ

$$\begin{aligned} p(x_\tau \,|\, a_\tau, s_{\tau-1}, \theta) &\coloneqq \mathcal{N}\left(s_\tau \mid \mu^x_\theta\left(s_{\tau-1}, a_\tau\right), \Sigma^x_\theta\left(s_{\tau-1}, a_\tau\right)\right) \\ p(r_\tau \,|\, x_\tau, a_\tau, \theta) &\coloneqq \mathcal{N}\left(r_\tau \mid \mu^r_\theta\left(s_\tau, a_\tau\right), \sigma^r_\theta\left(s_\tau, a_\tau\right)\right). \end{aligned}$$

There are multiple options for representing distributions over neural network parameters. Common choices are particle-based representations [34, 35], Gaussian distributions with diagonal covariance matrix [36] or a combination of both [37]. Since our approximation of the planning objective only requires samples of θ, we choose to represent $p_k(\theta)$ by a set of particles $\theta_1, \ldots, \theta_n$, making our model a neural network ensemble.

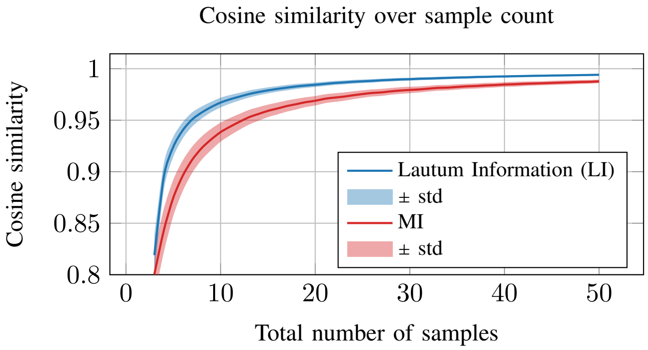


Fig. 2: Comparison of the cosine similarity between the sample-reusing NMC approximations of MI and LI to their respective exact values. For each number of samples, we conducted 1000 experiments on randomly generated discrete generative models $p(s, \theta \,|\, \pi)$. To assess the approximation quality independently of scale, we compute the cosine similarity between the approximated and exact information vectors $(\mathrm{I}\,(s, \theta \mid \pi_1), \ldots, (s, \theta \mid \pi_m))^T$, where I is either MI or LI and m is the number of policies in the model. *Number of samples* refers to the sum of s and θ samples. Note that the optimal value possible under cosine similarity is 1.

C. Lautum Information

An alternative to our choice of the intrinsic term is to use the reverse KL divergence, yielding the LI [38], which in our case is defined as

$$\begin{aligned} \mathrm{I}\,(\pi, s_t) &\coloneqq \mathrm{LI}\,((\mathbf{s}, \mathbf{r})\,, \theta \mid \pi, s_t) \\ &= \mathbb{E}_{p(\mathbf{s},\mathbf{r} \,|\, \pi, s_t)}[D_{\mathrm{KL}}[p(\theta) \parallel p(\theta \,|\, \mathbf{s}, \mathbf{r}, \pi, s_t)]]\,. \end{aligned}$$

LI, similar to MI, measures how much information we are expected to gain about θ by observing $(\mathbf{s}, \mathbf{r})$. However, it does it in a different way and leads to a different result, similar to how reverse KL leads to mode-seeking behavior and the forward KL to the moment matching behavior.

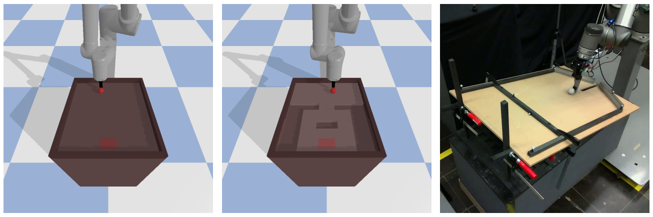
To gain an intuition about the difference between MI and LI, it is useful to consider which policy π maximizes each of them. For LI, the prior $p(\theta)$ is in the numerator in the KL divergence term, therefore LI becomes maximal for policies that are expected to produce observations which make a-priori likely θ have a low probability in the posterior. In a sense, LI encourages the agent to seek out observations that disprove the optimality of those θ that the agent assigned a high prior probability to. Analogously, MI encourages the agent to make observations that cause θ with a low prior probability to have a high posterior probability.

One advantage of LI is the independence of $(\mathbf{s}, \mathbf{r})$ and θ in the outer expectations

$$\mathrm{LI} = \mathbb{E}_{p(\theta)}\big[\mathbb{E}_{p(\mathbf{s},\mathbf{r} \,|\, \pi, s_t)}[\ln p(\mathbf{s}, \mathbf{r}) - \ln p(\mathbf{s}, \mathbf{r} \,|\, \pi, s_t, \theta)]\big]$$

which allows for a more efficient reuse of samples. Contrary to the MI approximation, when approximating the inner expectation $\mathbb{E}_{p(\mathbf{s},\mathbf{r} \,|\, \pi, s_t)}[.]$, we can reuse the same samples $(\mathbf{s}^i, \mathbf{r}^i)$ for all samples θ^j from the outer expectation. Hence, the resulting NMC approximation of LI is given as

$$\mathrm{LI} \approx \frac{1}{n}\sum_{i=1}^{n} \ln\left(\frac{1}{n}\sum_{\substack{k=1\\k\neq i}}^{n} p\big(\mathbf{s}^i, \mathbf{r}^i \,\big|\, \theta^k\big)\right) - \frac{1}{n}\sum_{\substack{j=1\\j\neq i}}^{n} \ln p\big(\mathbf{s}^i, \mathbf{r}^i \,\big|\, \theta^j\big)\,.$$



(a) *Tilted Pushing* (b) *Tilted Pushing Maze* (c) *Tilted Pushing Real*

Fig. 3: Visualization of the three environment configurations we test our methods on. The target zone (marked in red in the simulated environments) is always at the top center of the table and its location is not observed by the agent.

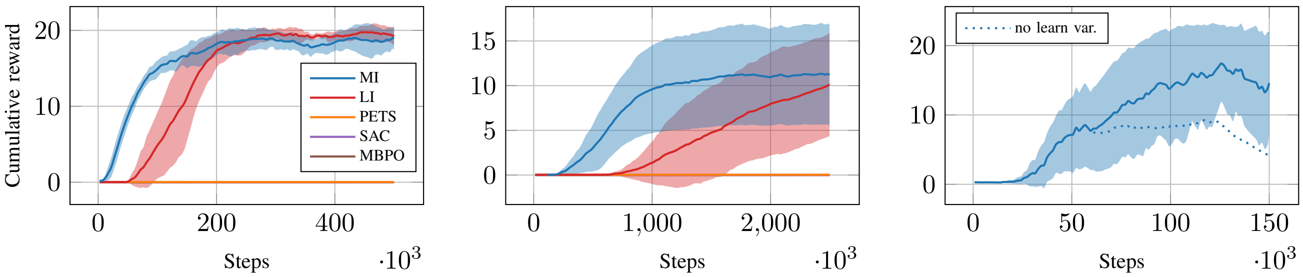
An empirical comparison of the stochastic estimators of MI and LI is shown in Fig. 2, which suggest a clear advantage in sample efficiency of the LI approximation in comparison to the MI approximation. Theoretical analysis of these estimators may be of interest for future work. The practical influence of the choice of the information gain term is evaluated in Section IV. To our knowledge, LI has not been used in the context of exploration before.

Further implementation details as well as the link to our code can be found on our project page:
`https://sites.google.com/view/aerm`.

IV. Experimental Results

A central feature that sets our method apart from other purely model-based approaches [20, 21] is the intrinsic term, that explicitly drives the agent to explore its environment in a systematic manner. Since to our knowledge, there exists no well established benchmark for hard-to-explore continuous-control robotic manipulation tasks yet, we designed two such tasks in simulation, *Tilted Pushing* and *Tilted Pushing Maze*), and one in the real world, *Tilted Pushing Real* (see Fig. 3). In all tasks, the agent has to push a ball up a tilted table into a target zone to receive a reward. The agent can move the gripper in a plane parallel to the table and, in the simulated tasks, also rotate the black end-effector around the Z-axis (Z-axis being orthogonal to the brown table and pointing up). In the real world experiments we disabled end-effector rotation to make the task complexity and thereby also the training time more manageable. As input, the agent receives the 2D positions and velocities of both the gripper and the ball, and the angular position and velocity of the end-effector. For the real robot training we rely on motion capture for measuring the state of the environment. To test the limits of our methods, in the *Tilted Pushing Maze* task we add holes to the table, that irrecoverably trap the ball if it falls in.

There are two aspects that make these tasks particularly challenging. First, the reward is sparse, meaning that the only way the agent can learn about the reward at the top of the table is by moving the ball there and exploring it. The agent receives 1 when pushing the ball in the target goal area, and 0 otherwise, except for a small penalty for large actions. Second, balancing the ball on the finger and moving it around is non-trivial and requires dexterity, especially given the low control frequency of 4 Hz we operate our agent on.



(a) *Tilted Pushing*

(b) *Tilted Pushing Maze*

(c) *Tilted Pushing Real*

Fig. 4: Cumulative reward in three different experiments for both versions of our agent (MI and LI) in comparison to three baselines: PETS [20], SAC [39], and MBPO [10]. The graphs display the evaluation reward, which is obtained by rolling out the learned model or policy without adding intrinsic reward or action noise. In both simulated tasks (*Tilted Pushing* and *Tilted Pushing Maze*), all baselines failed to discover the reward at the top of the table and converged to local minima. Our configurations found the reward consistently within 1,500,000 steps except for one of the five LI runs on the *Tilted Pushing Maze* task. On the real robot we evaluate only the MI configuration, since each experiment of 150,000 steps corresponds to approximately 21 hours of training time. The dashed blue line in (c) indicates a continuation of the experiment where we did not start to learn transition variances after 60,000 steps. Note that the episode length of the simulation experiments is 50 steps, while we only use 30 steps for the real-world experiments to speed up the training.

Note that the robot actions have to be slow, always balancing the ball at the tip of the end-effector, and it doesn't try to achieve the goal by flicking the ball. This makes the combined control and exploration task significantly harder than simple pushing tasks. On top of that, the table is tilted, meaning that the gravity affects the balancing of the ball by the robot, making the dynamics of the task very hard to learn. Once the agent drops the ball, this cannot be recovered, giving the agent no choice but to wait for the episode to terminate to continue exploring. Both of these aspects make solving these tasks with conventional, undirected exploration methods like Boltzmann exploration or Gaussian action noise extremely challenging. Consequently, the agent has to learn to balance the ball without receiving any extrinsic reward, purely driven by its own curiosity.

In both simulated experiments, we compare against three baselines: PETS [20], SAC [39], and MBPO [10]. We chose these three methods, because they are popular examples of each of the three main categories that most modern RL algorithms fall into: SAC is completely model-free, PETS is fully model-based and MBPO is a hybrid approach where the model is used to generate additional data for an underlying model-free agent. Note that we do not compare to other methods that use other types of intrinsic rewards to induce exploration [14], since, as we described in Section II, those are not pure model-based methods but rely on learning a policy to explore for learning a model. For a comprehensive overview of exploration methods in RL, refer to [40].

For all tasks, our method uses an ensemble of 5 fully-connected neural networks for both dynamics and reward models. Furthermore, in simulation, we found it beneficial to use a constant standard deviation of 0.001 for both models instead of learning it from the data.

As shown in Fig. 4, our method is able to solve the *Tilted Pushing* task consistently with both MI and LI as intrinsic terms. All baselines fail to find the reward within 300,000 steps and converge to local minima. Also, the *Tilted Pushing Maze* task is solved consistently within 1,500,000 steps by our method, with only a single agent of the LI configuration being unable to find the reward in time. Note, that the holes of *Tilted Pushing Maze* make this environment significantly harder to explore, as the ball has to be maneuvered around two corners in order to reach the target zone. As can be seen in Fig. 5, the reason for the high performance of our method on this task is a much better state space coverage compared to PETS. While our agents systematically maneuver the ball around the holes in unseen locations, the non-intrinsic agent rarely passes the lower holes and leaves the upper half of the table completely unexplored. Given that PETS is purely driven by the extrinsic reward, this behavior is not surprising, as these environments initially provide the agents with no direct incentive to learn to balance the ball. Note that SAC and MBPO with their maximum entropy exploration strategy did not manage to learn the tasks, revealing that hard exploration problems require directed information seeking strategies, as our method does.

To show that our method is principally capable of solving real-world tasks, we re-created the *Tilted Pushing* in the real world as shown in Fig. 3c. Applying our approach to the real world, for online training with the robot from scratch, yields a number of additional challenges that the agent has to deal with. A central challenge we faced is the occurrence of unobserved variables in our environment that violate our Markov assumption. One such example is that we only observe the position of the ball and not the spin. Hence, the agent has no way of knowing whether the ball is currently sliding or rolling, yet the future trajectory of the ball may largely depend on this information. A typical example where information about the ball's spin becomes important is when the robot does some left-right jittering motion. In this situation, the ball will remain stable as long as it does not start rolling along the finger. Without knowing the spin, reliably predicting whether the ball will stay on the finger is likely impossible.

We tackled this issue by learning not only the means, but also the variances of the transition model. In the case of this environment, we found that learned variances cause the planner to avoid jittering motions, as they tend to lead to a high state uncertainty and, thus, a lower expected reward. However, we also found that learning the variances too early in the training leads to pessimistic behavior, where the agent stops moving at some point in the training despite having found the reward multiple times before. So instead of learning the variances directly from the start, we start learning them after 60,000 steps on, when we have collected enough data to predict sensible variances.

Fig. 4c shows the results of training with our method on the real-world setup. Note that we do not pre-train in simulation, but learn the model from scratch in the real world. Despite the state not being fully observable, our method solves the *Tilted Pushing* in the real-world with a performance comparable to the simulated equivalent.

These experiments show that our method is able to systematically explore a complex, contact-rich environment with many dead-ends. Without any extrinsic feedback, our agents learned to balance the ball on the end-effector and to systematically move it around the environment until the target zone is found. The sole reason for this behavior to occur in the first place is that our agents understood they could only explore the entire state-space, if they kept balancing the ball and move it to unseen locations. Our final experiment shows that our method generalizes to the real world, while only requiring some simple algorithmic changes. These results serve as a proof of concept for using model-based active exploration to learn challenging robotic manipulation tasks.

V. Connections to Active Inference

Our approach is related to the Active Inference (AI) [9] framework, which we briefly summarize in the following. AI is an implementation of the Free Energy Principle (FEP) [41], which attempts to explain intelligent behavior from a cognitive science perspective. The fundamental idea behind FEP is that any organism's objective is to restrict the states it is visiting to a manageable amount. From this basic principle, a process theory is derived that reproduces features of intelligent behavior and curiosity [42].

Mathematically, AI implements the FEP's objective as follows: An agent maintains a generative model p of the world and avoids sensations o that are surprising, i.e., that have a low marginal log-probability $\ln p(o)$. Thus, the objective can be written as

$$\min_{\pi} \; -\ln p(o)$$

where o is generated by an external process that can be influenced by changing the policy π.

The agent's generative model is assumed to consist not only of observations o, but also include hidden states x, giving $p(o) = \int p(o,x)\,dx = \int p(o\,|\,x)\,p(x)\,dx$. To make optimization tractable, variational inference is invoked to obtain the Evidence Lower Bound using Jensen's inequality:

$$\begin{aligned} -\ln p(o) &= -\ln \int p(o,x)\,dx = -\ln \int \frac{q_\phi(x)}{q_\phi(x)} p(o,x)\,dx \\ &\leq D_{\mathrm{KL}}[q_\phi(x) \,\|\, p(x\,|\,o)] - \ln p(o) =: \mathcal{F}(o,\phi) \end{aligned}$$

where $\mathcal{F}(o,\phi)$ is termed the Variational Free Energy (VFE).

Minimizing $\mathcal{F}(o,\phi)$ w.r.t. the variational parameters ϕ corresponds to minimizing the KL divergence between the variational posterior $q_\phi(x)$ and the true posterior $p(x\,|\,o)$. In other words, by minimizing the VFE w.r.t. ϕ, the agent is solving the perception problem of mapping its observations to their latent causes.

To facilitate planning into the future, the VFE can be modified to incorporate an expectation over future states, yielding the Expected Free Energy (EFE) [42]:

$$\begin{aligned} G_\pi(\phi) &= -\mathbb{E}_{q_\phi(\mathbf{o},\mathbf{x}\,|\,\pi)}[\ln p(\mathbf{o},\mathbf{x}) - \ln q_\phi(\mathbf{x}\,|\,\pi)] \\ &\approx -\sum_{\tau=t+1}^{t+H} \mathbb{E}_{q_\phi(o_\tau,x_\tau\,|\,\pi)}[\ln p(o_\tau,x_\tau) - \ln q_\phi(x_\tau\,|\,\pi)] \\ &\approx -\sum_{\tau=t+1}^{t+H} \Big(\underbrace{\mathbb{E}_{q_\phi(o_\tau\,|\,\pi)}[\ln p(o_\tau)]}_{\text{extrinsic term}} \swarrow \\ &\qquad + \underbrace{\mathbb{E}_{q_\phi(x_\tau\,|\,\pi)}[D_{\mathrm{KL}}[q_\phi(o_\tau\,|\,x_\tau,\pi) \,\|\, q_\phi(o_\tau\,|\,\pi)]]}_{\text{intrinsic term (expected information gain)}} \Big) \end{aligned}$$

where a mean-field assumption is made in the second step and definitions $\mathbf{o} := o_{t+1:t+H}$ and $\mathbf{x} := x_{t+1:t+H}$ are used.

The minimization of the EFE w.r.t. the policy π causes the agent to act in a way that maximizes both the information gain and the extrinsic term. Here, the extrinsic term acts as an external signal expressing the preferences of the agent over observations. While it is common in RL to use a reward function to give the agent a notion of "good" and "bad" behavior, in the AI framework, one defines a prior distribution over the target observations $p(o)$ that the agent tries to match. By making the reward part of the observation and setting the maximum reward as the target observation, one can transform any reward-based task to fit into the AI framework [35].

In our implementation, the agent observes the state of the environment s_τ and the reward r_τ at every time step τ. The only unobserved variables are the model parameters θ. Consequently, the hidden state is given by $x_\tau = (s_\tau, r_\tau)$. Setting the preference distribution to $p(o_\tau) \propto \exp(\beta r_\tau)$ and dropping constant terms makes the EFE at time t become

$$\begin{aligned} G_\pi(\phi) \propto -\sum_{\tau=t+1}^{t+H} \Big(&\beta \mathbb{E}_{q_\phi(r_\tau\,|\,\pi)}[r_\tau] \swarrow \\ &+ \mathbb{E}_{q_\phi(\theta\,|\,\pi)}[D_{\mathrm{KL}}[q_\phi(x_\tau, r_\tau\,|\,\theta,\pi) \,\|\, q_\phi(x_\tau, r_\tau\,|\,\pi)]] \Big). \end{aligned}$$

The only difference between the above objective and our planning objective is that we do not make the mean-field assumption over time. Hence, our method can be understood as an implementation of AI where the agent encodes its belief about the dynamics of the environment in its hidden state. As such our method is related to [35], where a variant of AI was used for model learning and the results were reported for simple benchmark environments, such as *Mountain Car* [43] and *Cup Catch* [44]. To our knowledge, our method is the first demonstration of AI used for active model learning on a real robotic system on a sparse-reward manipulation task.

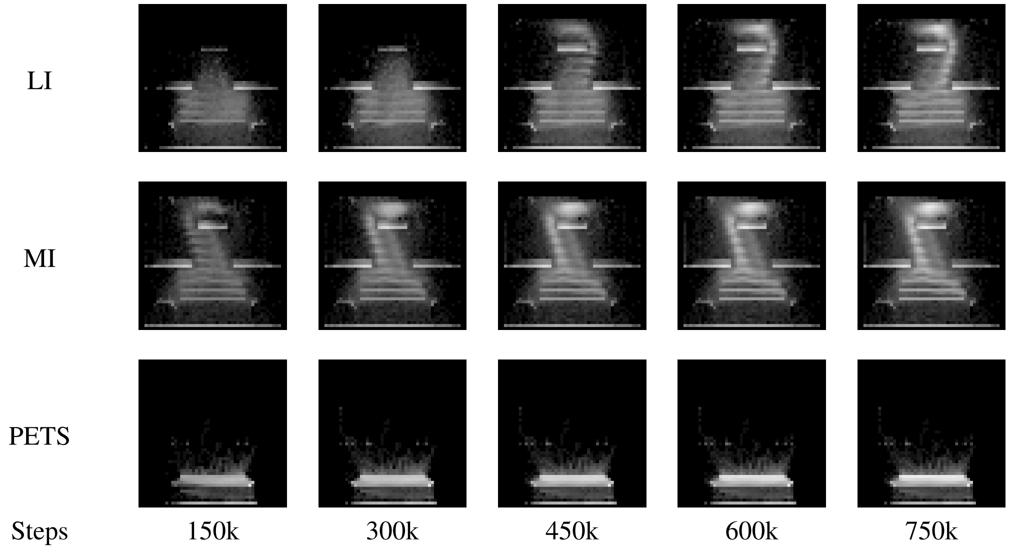


Fig. 5: Comparison of the states visited by our MI and LI agents and PETS [20] in the *Tilted Pushing Maze*. The brightness of each pixel indicates how often the ball has visited the respective point of the table at the given point in the training. The coordinate origin is at the bottom of each image, meaning that the images are rotated 180°compared to the top-down view in Fig. 3. Both the LI agent and the MI agent succeed in solving the task because they achieve a much better coverage than PETS [20].

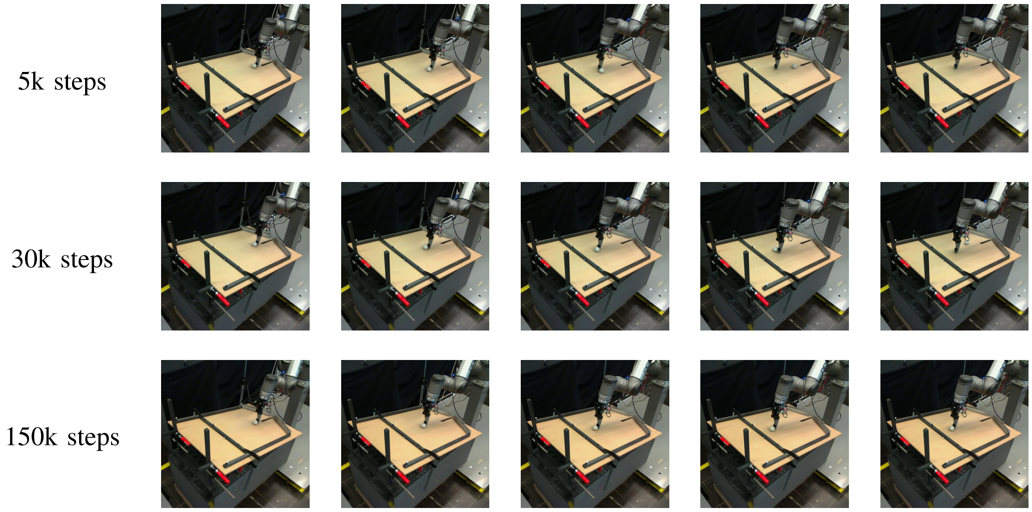


Fig. 6: Training process of our agent on the real system. At 5k steps, the agent has not found the target yet, and thus it is not guided by any external reward signal. Instead, purely driven by its intrinsic drive, it learns to balance the ball on the finger and systematically explores the table. At 30k steps, the agent understood how to obtain the reward in this task and the extrinsic signal causes it to focus on exploitation from now on. At 150k steps, the agent's model is accurate enough to repeatedly balance the ball at the goal location and solve the task.

VI. Conclusion

In this paper, we developed an active exploration method that is capable of solving complex robotic manipulation tasks. Our main algorithmic contribution is the introduction of an information-seeking strategy in model-based reinforcement learning, that balances between exploration of new states in the environment to improve the dynamics model and task performance. We evaluated our method on two simulated and one real-world tasks, all designed to be particularly hard to explore. Throughout our experiments, we showed that our method induces systematic exploratory behavior and learns to solve a manipulation task without dense extrinsic reward, but is driven by its own curiosity. Considering that none of the baselines were able to solve these problems, we conclude that the information-seeking behavior of our agents is beneficial for solving challenging exploration problems with sparse rewards, suitable for learning complex manipulation tasks in the real world.

In the future, we plan to incorporate tactile sensors into our setup. Tactile sensors would allow one to obtain more detailed feedback about the objects being manipulated. For example, the spin of the ball was not observed in our experiments, although it provides a useful signal for the manipulation task. Considering that humans deploy a variety of active haptic exploration strategies during manipulation, research on robotic active tactile exploration might bring us closer to human-level manipulation skills. One of the main limitations of our method that prevents it from being used on more dynamic tasks is that it is computationally comparably heavy and therefore limited to relatively slow tasks. Hence, an exciting future research direction is to tackle this issue with hierarchical controllers by combining our planning module with a learned low-level balancing controller running at a high frequency.

References

[1] R. MacDougall, "The significance of the human hand in the evolution of mind," *The American Journal of Psychology*, vol. 16, no. 2, pp. 232–242, 1905.

[2] R. W. Young, "Evolution of the human hand: The role of throwing and clubbing," *Journal of Anatomy*, vol. 202, no. 1, pp. 165–174, 2003.

[3] O. Kroemer, S. Niekum, and G. D. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *Journal of Machine Learning Research*, vol. 22, no. 30, 2021.

[4] A. Lacreuse and D. M. Fragaszy, "Manual exploratory procedures and asymmetries for a haptic search task: A comparison between capuchins (cebus apella) and humans," *Laterality: Asymmetries of Body, Brain and Cognition*, vol. 2, no. 3-4, pp. 247–266, 1997.

[5] M. T. Turvey and C. Carello, "Dynamic touch," *Perception of space and motion*, pp. 401–490, 1995.

[6] Z. LI, "On motion planning for dexterous manipulation, part i: The problem formulation," in *Proc. IEEE Conf. on Robotics and Automation, Scottsdale, AZ, 1989*, 1989, pp. 775–780.

[7] B. Fang, S. Jia, D. Guo, M. Xu, S. Wen, and F. Sun, "Survey of imitation learning for robotic manipulation," *International Journal of Intelligent Robotics and Applications*, vol. 3, no. 4, pp. 362–369, 2019.

[8] D. Kalashnikov *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning*, PMLR, 2018, pp. 651–673.

[9] T. Parr, G. Pezzulo, and K. J. Friston, *Active inference: the free energy principle in mind, brain, and behavior*. MIT Press, 2022.

[10] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," *arXiv preprint arXiv:1906.08253*, 2019.

[11] A. S. Morgan, D. Nandha, G. Chalvatzaki, C. D'Eramo, A. M. Dollar, and J. Peters, "Model predictive actor-critic: Accelerating robot skill acquisition with deep reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 6672–6678.

[12] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[13] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International conference on machine learning*, PMLR, 2017, pp. 2778–2787.

[14] D. Pathak, D. Gandhi, and A. Gupta, "Self-supervised exploration via disagreement," in *International conference on machine learning*, PMLR, 2019, pp. 5062–5071.

[15] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*, Citeseer, 2011, pp. 465–472.

[16] L. Roveda *et al.*, "Model-based reinforcement learning variable impedance control for human-robot collaboration," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 2, pp. 417–433, 2020.

[17] G. Chalvatzaki, X. S. Papageorgiou, P. Maragos, and C. S. Tzafestas, "Learn to adapt to human walking: A model-based reinforcement learning approach for a robotic assistant rollator," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3774–3781, 2019.

[18] V. Pong, S. Gu, M. Dalal, and S. Levine, "Temporal difference models: Model-free deep rl for model-based control," *arXiv preprint arXiv:1802.09081*, 2018.

[19] Y. Chebotar, K. Hausman, M. Zhang, G. Sukhatme, S. Schaal, and S. Levine, "Combining model-based and model-free updates for trajectory-centric reinforcement learning," in *International conference on machine learning*, PMLR, 2017, pp. 703–711.

[20] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *arXiv preprint arXiv:1805.12114*, 2018.

[21] D. Hafner *et al.*, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*, PMLR, 2019, pp. 2555–2565.

[22] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, "Intrinsically motivated reinforcement learning: An evolutionary perspective," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 70–82, 2010.

[23] T. Hester and P. Stone, "Intrinsically motivated model learning for developing curious robots," *Artificial Intelligence*, vol. 247, pp. 170–186, 2017.

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

[25] P. Shyam, W. Jaśkowski, and F. Gomez, "Model-based active exploration," in *International conference on machine learning*, PMLR, 2019, pp. 5779–5788.

[26] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak, "Planning to explore via self-supervised world models," in *International Conference on Machine Learning*, PMLR, 2020, pp. 8583–8592.

[27] S. Bechtle, Y. Lin, A. Rai, L. Righetti, and F. Meier, "Curious iLQR: Resolving Uncertainty in Model-based RL," *arXiv*, Apr. 2019. DOI: 10.48550/arXiv.1904.06786. eprint: 1904.06786.

[28] A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune, "Go-Explore: a New Approach for Hard-Exploration Problems," *arXiv*, Jan. 2019. DOI: 10.48550/arXiv.1901.10995. eprint: 1901.10995.

[29] D. V. Lindley, "On a measure of the information provided by an experiment," *The Annals of Mathematical Statistics*, vol. 27, no. 4, pp. 986–1005, 1956.

[30] D. B. F. Agakov, "The im algorithm: A variational approach to information maximization," *Advances in neural information processing systems*, vol. 16, no. 320, p. 201, 2004.

[31] A. Foster *et al.*, "Variational bayesian optimal experimental design," *arXiv preprint arXiv:1903.05480*, 2019.

[32] M. I. Belghazi *et al.*, "Mine: Mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[33] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *International Conference on Machine Learning*, PMLR, 2019, pp. 5171–5180.

[34] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, vol. 11, pp. 169–198, 1999.

[35] A. Tschantz, B. Millidge, A. K. Seth, and C. L. Buckley, "Reinforcement learning through active inference," *arXiv preprint arXiv:2002.12636*, 2020.

[36] J. Lampinen and A. Vehtari, "Bayesian approach for neural networks—review and case studies," *Neural networks*, vol. 14, no. 3, pp. 257–274, 2001.

[37] L. V. Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, "Hands-on bayesian neural networks–a tutorial for deep learning users," *arXiv preprint arXiv:2007.06823*, 2020.

[38] D. P. Palomar and S. Verdú, "Lautum information," *IEEE transactions on information theory*, vol. 54, no. 3, pp. 964–975, 2008.

[39] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, PMLR, 2018, pp. 1861–1870.

[40] S. Amin, M. Gomrokchi, H. Satija, H. van Hoof, and D. Precup, "A Survey of Exploration Methods in Reinforcement Learning," *arXiv*, Sep. 2021. DOI: 10.48550/arXiv.2109.00157. eprint: 2109.00157.

[41] K. J. Friston, J. Daunizeau, J. Kilner, and S. J. Kiebel, "Action and behavior: A free-energy formulation," *Biological cybernetics*, vol. 102, no. 3, pp. 227–260, 2010.

[42] K. Friston, F. Rigoli, D. Ognibene, C. Mathys, T. Fitzgerald, and G. Pezzulo, "Active inference and epistemic value," *Cognitive neuroscience*, vol. 6, no. 4, pp. 187–214, 2015.

[43] G. Brockman *et al.*, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[44] Y. Tassa *et al.*, *Dm_control: Software and tasks for continuous control*, 2020. arXiv: 2006.12983 [cs.RO].