Active Inference for Robotic Manipulation

Master thesis by Tim Schneider
Date of submission: October 20, 2021

1. Review: Boris Belousov
2. Review: Hany Abdulsamad
3. Review: Prof. Jan Peters
Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

Computer Science Department

Intelligent Autonomous Systems

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 und §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Tim Schneider, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß §23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 20. Oktober 2021

Tim Schneider

Abstract

Despite significant advances in robotics and machine learning in the last decades, robotic manipulation stands as a largely unsolved problem. One of the central challenges of manipulation is partial observability, as the agent usually does not know all physical properties of the objects it is manipulating in advance. A recently emerging theory that deals with partial observability in an explicit manner is Active Inference. It does so by driving the agent to act in a way that is not only goal directed, but also informative about the environment. In this work, we apply Active Inference to simulated robotic manipulation tasks. We show that the information seeking behavior induced by Active Inference allows the agent to systematically explore challenging sparse environments. Furthermore, we propose to replace the information seeking objective of Active Inference by one that is based on Lautum Information and argue that this choice facilitates a more sample efficient approximation. Our experimental results indicate that this choice of objective achieves performance comparable to the original objective on simulated robotic manipulation tasks. Additionally, we provide an in-depth formal analysis of the origin of information seeking behavior in Active Inference. In this analysis, we cast doubt on the mathematical soundness of a central claim of Active Inference, namely that it explains curiosity and thereby resolves the exploration-exploitation dilemma. Finally, we conclude that using an information seeking objective is beneficial in sparse environments and allows the agent to solve tasks in which methods that do not exhibit directed exploration fail.

Contents

List of Abbreviations

AI	Active Inference
BOED	Bayesian Optimal Experimental Design
CEM	Cross Entropy Method
EFE	Expected Free Energy
EM	Expectation Maximization
FEEF	Free Energy of the Expected Future
FEF	Free Energy of the Future
FEP	Free Energy Principle
i.i.d.	independent and identically distributed
KL	Kullback-Leibler
LI	Lautum Information
MC	Monte Carlo
MCTS	Monte Carlo Tree Search
MDP	Markov Decision Process
MI	Mutual Information
MPC	Model Predictive Control
MSE	Mean Squared Error
NMC	Nested Monte Carlo
POMDP	Partially Observable Markov Decision Process
RBF	Radial Basis Function
RL	Reinforcement Learning
SAC	Soft Actor Critic
SGD	Stochastic Gradient Descent
SVGD	Stein Variational Gradient Descent
VFE	Variational Free Energy

1 Introduction

A common belief 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 mankind in general [2]. Our manipulation skills allow us to interact with our environment in a highly sophisticated way, setting us humans apart from all other known species on this planet. Performing manipulation is cognitively highly demanding, forcing the actor to reason not only about the impact of its actions on itself, but also about the impact on its environment. This inherent complexity leaves autonomous robotic manipulation as a largely unsolved topic, despite significant advances in robotics and machine learning in the last decades.

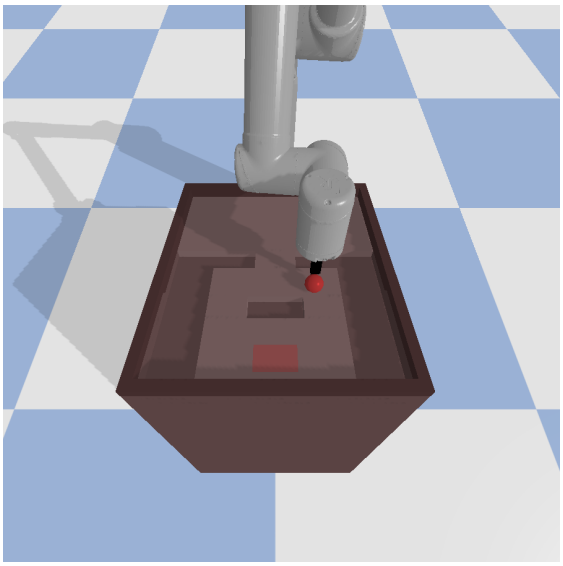


Figure 1.1: Robot using Active Inference to solve a challenging manipulation task.

One of the central challenges of manipulation is partial observability. While we are manipulating an object, we rarely know all of its physical properties in advance. Instead, we must resort to inferring those properties based on observations and touch. To deal with this issue as effectively as possible, humans have developed a variety of active haptic exploration strategies that they constantly apply during manipulation tasks [3, 4].

A recently emerging theory from cognitive science that tries to explain this notion of constant active exploration is Active Inference (AI) [5]. Active Inference formulates both

action and perception as the minimization of a single free-energy functional, called the Variational Free Energy. In doing so, Friston et al. [6] derive an objective function that consists of an extrinsic, goal-directed term and an intrinsic, information seeking term. The combination of these two terms drive the agent to act in a way that is both goal directed and informative. E.g. in the context of a pick-and-place task, goal directed behavior would be to pick the object up and move it to the target location. Information seeking behavior, on the other hand, would be to haptically explore the object, learning more about its shape, texture, pose and weight. A combination of both could be realized by picking the object up and exploring it on the way to the target location.

In this work, we show how Active Inference can be used to learn challenging robotic manipulation tasks without prior knowledge. For now, we assume that the environment is fully observable and only consider epistemic uncertainty[1]. To implement Active Inference in practice, we utilize an ensemble of neural network models and deploy Model Predictive Control for action selection. We show that agents driven by Active Inference explore their environments in a directed and systematic way. These exploratory capabilities allow the agents to solve complex sparse manipulation tasks, on which agents that are not explicitly information seeking fail.

A significant challenge we address in this work is the approximation of the intrinsic term. To obtain an efficient estimate, we propose to nest two Monte Carlo (MC) approximators and share samples among them. Although reusing samples violates the i.i.d. assumption of the approximators, we show empirically that it improves sample-efficiency substantially. Furthermore, we propose to use Lautum Information [7] as a novel intrinsic term and show empirically that it allows a more efficient reuse of samples than our approximator of the original intrinsic term.

Another contribution of this work is an in-depth formal analysis of the origin of the intrinsic term in Active Inference. We argue that the intrinsic term cannot be naturally obtained, but rather arises from an assumption that is not fulfilled in the overwhelming majority of cases. By empirically showing that only those agents that make this assumption exhibit information seeking behavior on a small example environment, we cast further doubt on the soundness of the derivation of the intrinsic term. In other words, we argue that if the mathematical derivations leading to the Active Inference objective are done exactly, then the agents do not exhibit information gain maximizing behavior.

This thesis is structured as follows. In Chapter 2, we summarize approaches related to

[1] Epistemic uncertainty is the uncertainty the agent has over its model of the world. In contrast, aleatoric uncertainty is uncertainty over the agent's state.

ours and point out their similarities and differences. We provide a summary of the Active Inference formalism in Chapter 3. In Chapter 4, we cast doubt on the mathematical soundness of a central claim of Active Inference, namely that it explains curiosity and thereby resolves the exploration-exploitation dilemma. Chapter 5 contains the formal derivation and a detailed description of our method. In Chapter 6, we present four different tasks we evaluated our method on and discuss its performance. Finally, in Chapter 7 we elaborate on merits and limitations of our method and discuss potential future research directions.

2 Related Work

In this chapter, we discuss prior work that is related to our approach. For a better overview, we separate the discussed publications into three categories. In the first category, we investigate how and to which extent Active Inference has been used in practical applications related to robotics or Reinforcement Learning. In the second category, we briefly summarize current model-based Reinforcement Learning methods and relate them to our approach. Finally, in the third category, we investigate other approaches in the field of Bayesian Optimal Experimental Design and Artifical Curiosity.

Practical applications of Active Inference

In the recent years, Active Inference (AI) has gained a lot of attention and started to find practical application in different fields. One field relevant to this work is robotics, in which AI has been used for a variety of tasks, ranging from control [8–13] to planning [14], navigation [15], and learning of dynamic models [14, 16–18].

Few approaches have yet used AI to solve Reinforcement Learning (RL) problems. One of those approaches was proposed by Millidge [19], who applies AI to three small RL benchmarks [20]. Unlike our approach, they do not use the Expected Free Energy (EFE) functional for planning, but rather use it to learn a neural network policy and value function. Another difference is that they do not assume that the model parameters are part of the hidden state, but rather train the model jointly with the variational distribution. Consequently, their method does not compute the intrinsic term w.r.t. the model parameters, but only w.r.t. the state.

Similar to the approach of Millidge [19] are the approaches of Ueltzhöffer [21] and Himst and Lanillos [22]. Ueltzhöffer [21] is different from Millidge [19], in that they deploy an evolutionary algorithm to optimize their models, which limited the method to very small

neural networks. Himst and Lanillos [22] use a similar overall approach as Millidge [19], but operate on pixel input instead of the numeric state vectors of the environments.

Related to our approach is Tschantz et al. [23], who also use Model Predictive Control (MPC) for control and use an ensemble to track model uncertainty. A difference to our approach is the free energy functional used for planning. While we rely on the Expected Free Energy, they use the Free Energy of the Expected Future, which we describe in detail in Section 3.3. The difference between the two functionals is the resulting extrinsic term, which is expressed as a KL divergence in case of the Free Energy of the Expected Future (FEEF) and a cross-entropy in case of the EFE. Furthermore, they chose a different approximation of their intrinsic term, which requires them to make a mean-field assumption over consecutive states. Similar to our approach, they also assume a Markov Decision Process (MDP) as their generative model. They evaluate their approach on multiple RL benchmarks, including *Mountain Car* [20] and *Cup Catch* [24].

Model-Based Reinforcement Learning

Since our method learns a model of its environment, it is related to methods in the field of Model-Based RL. A popular model-based RL approach is MBPO [25], which we also compare against in this work. MBPO uses samples of the environment to train an ensemble of dynamics models, which are used to generate novel artificial data. This artificial data is then used to train a SAC agent. A difference to our approach is that the ensemble model is not used during roll-outs, as the actions are sampled from the SAC policy.

Another approach is PILCO [26], where a Gaussian Process model is trained on data from the environment. Then PILCO uses the differentiability of its model to compute the policy gradient analytically. Similarly to MBPO, only the policy is used during roll-outs.

An approach similar to ours in PETS [27]. Like our method, PETS trains ensemble models for both the transition and reward distributions and selects actions with a Cross Entropy Method (CEM) planner. Another approach that learns a probabilistic model and uses CEM for planning is PlaNet [28]. Unlike our method, however, PlaNet learns a latent representation, which makes it capable of operating on Partially Observable Markov Decision Process (POMDP) environments. The key difference to our approach is that both PETS and PlaNet do not use intrinsic terms and instead greedily select the actions they predict to yield the highest reward.

Bayesian Optimal Experimental Design and intrinsic motivation

Our method is related to Bayesian Optimal Experimental Design (BOED). In Bayesian Optimal Experimental Design (BOED), the objective is to choose experiment configurations in such a way that is their results will be maximally informative. Typically, this notion is formulated as the maximization of the expected information gain [24, 29–31].

Particularly relevant to our work is the approach of Shyam, Jaśkowski, and Gomez [32], who use active exploration for model learning in a dynamic environment. Similar to our approach, they make use of the expected information gain and approximate it with MC. Contrary to our method, they do not use MPC and instead use SAC to learn a policy that maximizes information gain. Crucially, during exploration, their method does not consider extrinsic reward and purely acts upon the information gain as objective. They show that their method is capable of solving *Mountain Car* and evaluate it on *Half Cheetah* [24]. Similar to the approach of Shyam, Jaśkowski, and Gomez [32] is the approach of Sun, Gomez, and Schmidhuber [30], who use policy iteration to obtain an information-gain maximizing policy.

Another, more general concept that is related to our approach is the use of intrinsic motivation. Intrinsic motivation can be understood as any kind of active drive that is not caused by an extrinsic reward signal. Schmidhuber [33] and Chentanez, Barto, and Singh [34] propose to formulate intrinsic motivation in terms of the disagreement over outcomes of multiple agents. Hence, instructions become intrinsically interesting if the agents predict different outcomes. Pathak et al. [35] take this idea one step further and use model disagreement as reward signal for A3C [36], to obtain an information seeking policy. Other approaches [29, 37] use an uncertainty aware model and rate the informativeness of actions by the entropy of the resulting prediction. Thus, actions for which the model prediction exhibits a high entropy receive a higher intrinsic value than actions that result in a low prediction entropy.

3 Active Inference

Active Inference (AI) [5, 38] is a process theory that implements the Free Energy Principle (FEP), a concept that recently emerged in the field of cognitive science. The Free Energy Principle (FEP) offers a unified brain theory that accounts for action, perception and learning. At its core lies the idea that any living organism has to restrict the space of possible states it is visiting in order to resist the second law of thermodynamics. For example, fish restrict themselves to remain underwater, while many species of birds migrate each year to ensure that the surrounding temperature is restricted to some acceptable levels. By formulating this notion of the purposeful restriction of the state space as the minimization of Bayesian surprise, Friston et al. [5] obtain a formulation that they claim explains intelligent behavior.

In this chapter, we summarize four formulations that attempt to implement the FEP mathematically. We describe the Variational Free Energy (VFE) [5], which implements the FEP for stationary systems without any notion of past and future, in Section 3.1, and show how it can be reformulated to facilitate operation on Partially Observable Markov Decision Processs (POMDPs). In Section 3.2, we derive the Expected Free Energy (EFE) [6, 39] which allows agents to plan their actions into the future, while still relying on the Variational Free Energy (VFE) for perception and learning. For simplicity, we will use the term Active Inference (AI) to refer to the process theory that results from the combined minimization of VFE and EFE, although in the literature the term is also used to refer to other process theories in the context of free energy minimization [5, 38, 39]. Finally, we briefly highlight an alternative formulation to the combination of VFE and EFE that also operates on POMDPs in Section 3.3.

3.1 Variational Free Energy

According to the FEP, any organism must restrict the states it is visiting to a manageable amount. Mathematically, AI implements this restriction as follows: every agent maintains a generative model p of the world and avoids sensations o that are surprising, hence have a low marginal log-probability $\ln p(o)$. Thus, the objective can be written as

$$\min_{\pi} \; -\ln p(o) \tag{3.1}$$

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

The agent's generative model is assumed to consist of not only observations o, but also contain hidden states x, making it a joint distribution over o and x: $p(o, x) = p(o \,|\, x)\, p(x)$. It is assumed that the agent can efficiently compute the likelihood $p(o \,|\, x)$ and the hidden state prior $p(x)$. Computing the objective in Eq. (3.1) requires to marginalize out the hidden states x, giving

$$-\ln p(o) = -\ln \int p(o, x)\, dx \tag{3.2}$$

However, $p(o, x)$ can be an arbitrarily complex distribution for which the integral in Eq. (3.2) does not have an analytical solution. We tackle this issue by introducing a variational posterior $q_\phi(x)$, which gives us the VFE formulation that the FEP owes its name to.

$$\begin{aligned}
-\ln p(o) &= -\ln \int p(o, x)\, dx \\
&= -\ln \int \frac{q_\phi(x)}{q_\phi(x)} p(o, x)\, dx && (3.3) \\
&\leq -\int q_\phi(x) \ln \frac{p(o, x)}{q_\phi(x)} dx && (3.4) \\
&= -\mathbb{E}_{q_\phi(x)}[\ln p(o, x) - \ln q_\phi(x)] \\
&=: \mathcal{F}(o, \phi)
\end{aligned}$$

where $\mathcal{F}(o, \phi)$ is the free energy, depending on the observation o and the parameters of the variational posterior ϕ. Note that to get from Eq. (3.3) to Eq. (3.4), we applied Jensen's inequality.

The free energy $\mathcal{F}(o, \phi)$ can now be decomposed as follows:

$$\begin{aligned}\mathcal{F}(o, \phi) &= -\mathbb{E}_{q_\phi(x)}\left[\ln \frac{p(o, x)}{q_\phi(x)}\right] \\ &= -\mathbb{E}_{q_\phi(x)}\left[\ln \frac{p(x \mid o)\, p(o)}{q_\phi(x)}\right] \\ &= D_{\mathrm{KL}}[q_\phi(x) \parallel p(x \mid o)] - \ln p(o) \end{aligned} \tag{3.5}$$

which shows that the difference between Eq. (3.3) and Eq. (3.4) is exactly the KL divergence between the variational posterior and the real posterior. This decomposition reveals that by minimizing $\mathcal{F}(o, \phi)$ w.r.t. the variational parameters ϕ, we are effectively minimizing the KL divergence between the variational posterior and the real posterior. Due to the positivity of the KL divergence, this minimization makes $\mathcal{F}(o, \phi)$ a better approximator of the surprise $-\ln p(o)$. To summarize, by minimizing $\mathcal{F}(o, \phi)$ w.r.t. o, we are minimizing an upper bound of the surprise $-\ln p(o)$, which we can tighten by minimizing $\mathcal{F}(o, \phi)$ w.r.t. ϕ. Note however, that we cannot control o directly, but rather choose actions that produce observations minimizing the free energy.

While Eq. (3.5) neatly demonstrates the relation between surprise and the free energy, it does not provide a straightforward way of computing the free energy. Eq. (3.5) is not the only decomposition of the free energy however, and we can find one that better facilitates computing $\mathcal{F}(o, \phi)$:

$$\begin{aligned}\mathcal{F}(o, \phi) &= -\mathbb{E}_{q_\phi(x)}\left[\ln \frac{p(o, x)}{q_\phi(x)}\right] \\ &= -\mathbb{E}_{q_\phi(x)}\left[\ln \frac{p(o \mid x)\, p(x)}{q_\phi(x)}\right] \\ &= \underbrace{D_{\mathrm{KL}}[q_\phi(x) \parallel p(x)]}_{\text{complexity}} - \underbrace{\mathbb{E}_{q_\phi(x)}[\ln p(o \mid x)]}_{\text{reconstruction}} \end{aligned} \tag{3.6}$$

If, for example, $p(x)$ and $q_\phi(x)$ are Gaussian, then the KL divergence in Eq. (3.6) can be computed analytically, while the expectation can be straightforwardly approximated with a Monte Carlo estimator.

A common argument against the FEP is the so called *Dark Room Problem* [40]: if all an agent tries to achieve is to reduce its surprise, why does it not simply search a dark room and stays in it forever? Inside a dark room any observation can be predicted perfectly and thus the agent should never be surprised. Yet, this is obviously not how animals or humans behave, which raises the question whether the FEP is a feasible brain theory. According

to Friston [41], however, this argument neglects the fact that the agent could find it surprising to be in a dark room in the first place. It could have observation preferences encoded in a prior probability distribution $p(o)$ that might be violated in such situations. Even agents that are not surprised to be in dark places per se, like mice hiding in their burrows, might find other preferences violated which cause them to act. For example, an animal might expect a-priori not to be hungry, urging it to leave the dark room to search for food.

In a RL setting, controlling the observation preferences allows us to define the target behavior of the agent. While it is common in RL literature to use a reward function to give the agent a notion of "good" and "bad" behavior, in the AI framework, we define a prior distribution over target observations $p(o)$ that we would like the agent to make. Note that by making the reward part of the observation and setting the maximum reward as target observation [23], we can transform any reward-based task to fit into the AI framework.

Variational Free Energy for POMDPs

As we are dealing with discrete time series data, we adapt the VFE formalism to POMDPs, following Parr and Friston [39]. Hence, in the following, we assume a discrete time setting in a Partially Observable Markov Decision Process (POMDP), where we denote the hidden state at time τ as x_τ and the observation as o_τ. We further use the notation $v_{\tau_1:\tau_2}$ to refer to $(v_{\tau_1}, v_{\tau_1+1}, \dots, v_{\tau_2})$ for time series data v and $\tau_1 \leq \tau_2$. Hence, the generative model of our environment can be factored as

$$p(x_{1:t}, o_{1:t}, \pi) = \prod_{\tau=1}^{t} p(o_\tau \mid x_\tau)\, p(x_\tau \mid x_{\tau-1}, \pi)\, p(\pi) \tag{3.7}$$

where t is the current time step, x_τ and o_τ are the hidden state and the observation at time τ, and π is a policy. In order not to clutter the notation, we simply define $p(x_1 \mid x_0, \pi) = p(x_1)$.

We further assume that the observation distribution $p(o_\tau \mid x_\tau)$ and the dynamics $p(x_\tau \mid x_{\tau-1}, \pi)$ are given and known to the agent. While it might sound limiting to assume that the agent must know these two distributions, it is important to emphasize that the hidden state x can be arbitrarily complex and could, for example, contain the parameters of a universal function approximator used inside these distributions. Hence, in that case, inferring the hidden state is includes learning the parameters of those function approximators. Consequently, $p(o_\tau \mid x_\tau)$ and $p(x_\tau \mid x_{\tau-1}, \pi)$ can be arbitrarily powerful distributions

that are configured by parts of the hidden state. For example, in this work, we chose the dynamics to be a Gaussian distribution parameterized by a neural network:

$$p(\tilde{x}_\tau, \theta_\tau \mid \tilde{x}_{\tau-1}, \theta_{\tau-1}, \pi) = \mathcal{N}\left(\tilde{x}_\tau \mid \mu\left(\tilde{x}_{\tau-1}; \theta_{\tau-1}\right), \sigma\left(\tilde{x}_{\tau-1}; \theta_{\tau-1}\right)\right) \delta\left(\theta_\tau - \theta_{\tau-1}\right)$$

where $x_\tau = (\tilde{x}_\tau, \theta_\tau)$ and the delta distribution $\delta\left(\theta_\tau - \theta_{\tau-1}\right)$ simply ensures that the parameters stay constant between steps.

To ensure consistency with the Expected Free Energy formalism introduced later, we additionally introduce π to the variables inferred with the variational posterior q, resulting in the following definition:

$$\begin{aligned} F\left(o_{1:t}, \phi\right) &= -\mathbb{E}_{q_\phi(x_{1:t}, \pi)}\left[\ln \frac{p(o_{1:t}, x_{1:t}, \pi)}{q_\phi(x_{1:t}, \pi)}\right] \\ &= -\mathbb{E}_{q_\phi(x_{1:t}, \pi)}\left[\ln \frac{p(x_{1:t}, \pi \mid o_{1:t})\, p(o_{1:t})}{q_\phi(x_{1:t}, \pi)}\right] \\ &= D_{\mathrm{KL}}[q_\phi(x_{1:t}, \pi) \parallel p(x_{1:t}, \pi \mid o_{1:t})] - \ln p(o_{1:t}) \\ &\geq -\ln p(o_{1:t}) \end{aligned} \tag{3.8}$$

To make this objective more tractable, we choose the variational distribution $q_\phi(x_{1:t}, \pi)$ such that the hidden states of different time steps are independent:

$$q_\phi(x_{1:t}, \pi) = q_\phi(\pi) \prod_{\tau=1}^{t} q_\phi(x_\tau \mid \pi)$$

This assumption is commonly referred to as a mean-field approximation. While a mean-field approximation drastically reduces the expressive power of the variational distribution, it allows us to decompose the VFE into a sum over time steps:

$$\begin{aligned} F\left(o_{1:t}, \phi\right) &= \mathbb{E}_{q_\phi(\pi)}[F_\pi\left(o_{1:t}, \phi\right)] + D_{\mathrm{KL}}[q_\phi(\pi) \parallel p(\pi)] \\ F_\pi\left(o_{1:t}, \phi\right) &= -\mathbb{E}_{q_\phi(x_{1:t} \mid \pi)}\left[\ln p(o_{1:t}, x_{1:t} \mid \pi) - \sum_{\tau=1}^{t} \ln q_\phi(x_\tau \mid \pi)\right] \\ &= \sum_{\tau=1}^{t} F_{\pi,\tau}\left(o_\tau, \phi\right) \end{aligned} \tag{3.9}$$

where the free energy at time τ is given as

$$F_{\pi,\tau}\left(o_\tau, \phi\right) = -\mathbb{E}_{q_\phi(x_\tau \mid \pi) q_\phi(x_{\tau-1} \mid \pi)}[\ln p(o_\tau \mid x_\tau) + \ln p(x_\tau \mid x_{\tau-1}, \pi) - \ln q_\phi(x_\tau \mid \pi)]$$

Note that $F(o_{1:t}, \phi)$ depends only on the past, as we defined t to be the current time step, making o_t our most recent observation. Hence, optimizing the variational distribution over policies $q_\phi(\pi)$ is not about finding a policy that minimizes surprise, but rather retrospectively inferring a policy that explains the observations we made in the past.

The optimal variational policy distribution can now be computed by setting the derivative of F to 0:

$$\begin{aligned} 0 = \frac{\partial F(\phi)}{\partial q_\phi(\pi)} &= F_\pi(\phi) - \ln p(\pi) + \ln q_\phi(\pi) \\ \Leftrightarrow \; q^*_\phi(\pi) &= \sigma(\ln p(\pi) - F_\pi(\phi)) \end{aligned} \tag{3.10}$$

where σ is the softmax function, defined as

$$\sigma(f(\pi)) = \frac{e^{f(\pi)}}{\int e^{f(\hat{\pi})} d\hat{\pi}}$$

One issue of the VFE formulation is that it only considers the past up to the present and does not allow the agent to plan into the future. Practically, this means that the agent cannot use this formulation to perform action selection unless it knows the relation between action and observation a-priori. To tackle this shortcoming, multiple attempts have been made to extend the free energy formalism to facilitate action planning into the future, which we summarize below.

3.2 Expected Free Energy

For reference, we summarize the Expected Free Energy (EFE) as it is formulated in Parr and Friston [39]. To extend this formalism for planning into the future, Parr and Friston [39] propose to replace $F_\pi(\phi)$ in Eq. (3.10) by EFE $G_\pi(\phi)$:

$$q^*_\phi(\pi) = \sigma(\ln p(\pi) - G_\pi(\phi)) \tag{3.11}$$

The EFE is defined as

$$\begin{aligned} G_\pi(\phi) &= \sum_{\tau=t+1}^{T} G_{\pi,\tau}(\phi) \\ G_{\pi,\tau}(\phi) &= -\mathbb{E}_{q_\phi(o_\tau, x_\tau \mid \pi)}[\ln p(o_\tau, x_\tau) - \ln q_\phi(x_\tau \mid \pi)] \end{aligned} \tag{3.12}$$

where $q_\phi(o_\tau, x_\tau \mid \pi) = p(o_\tau \mid x_\tau)\, q_\phi(x_\tau \mid \pi)$, t is the current time step and T is the planning horizon. Note that unlike the VFE, the EFE does not depend on any observations, but rather uses the variational distribution to take an expectation over future observations.

Eq. (3.12) can now be decomposed into an epistemic and an extrinsic term:

$$G_{\pi,\tau}(\phi) = -\mathbb{E}_{q_\phi(o_\tau, x_\tau \mid \pi)}[\underbrace{\ln p(x_\tau \mid o_\tau) - \ln q_\phi(x_\tau \mid \pi)}_{\text{epistemic term}} + \underbrace{\ln p(o_\tau)}_{\text{extrinsic term}}] \tag{3.13}$$

The intuition behind the extrinsic term is fairly straightforward: it drives the agent towards realizing its observation preferences, which are encoded in the observation prior $p(o_\tau)$. As described above, observations that the agent favors have a higher probability in the observation prior than observations that the agent disfavors. The intuition behind the epistemic term, however, is less obvious. Parr and Friston [39] offer the following explanation: using the approximation

$$p(x_\tau \mid o_\tau) \approx q_\phi(x_\tau \mid o_\tau, \pi) \tag{3.14}$$

we can apply Bayes' rule and obtain

$$\frac{p(x_\tau \mid o_\tau)}{q_\phi(x_\tau \mid \pi)} \approx \frac{q_\phi(x_\tau \mid o_\tau, \pi)}{q_\phi(x_\tau \mid \pi)} = \frac{q_\phi(o_\tau \mid x_\tau, \pi)}{q_\phi(o_\tau \mid \pi)} \tag{3.15}$$

where we defined $q_\phi(o_\tau \mid \pi) \coloneqq \int p(o_\tau \mid \hat{x}_\tau)\, q_\phi(\hat{x}_\tau \mid \pi)\, d\hat{x}$ in the final step.

Inserting Eq. (3.15) back into Eq. (3.13), we obtain the EFE as a sum of the expected information gain (also known as Mutual Information) and the extrinsic term:

$$\begin{aligned} G_{\pi,\tau}(\phi) &\approx \underbrace{-\mathbb{E}_{q_\phi(x_\tau \mid \pi)}[D_{\mathrm{KL}}[q_\phi(o_\tau \mid x_\tau, \pi) \parallel q_\phi(o_\tau \mid \pi)]]}_{\text{expected information gain}} \underbrace{-\mathbb{E}_{q_\phi(o_\tau \mid \pi)}[\ln p(o_\tau)]}_{\text{extrinsic term}} \\ &=: \tilde{G}_{\pi,\tau}(\phi) \end{aligned} \tag{3.16}$$

where we defined the approximated EFE as $\tilde{G}$.

This decomposition is often highlighted as one of the key features of Active Inference, as it neatly combines exploration and exploitation into a single objective [6, 42]. Friston et al. [6] even claim that Eq. (3.16) shows that AI resolves the the exploration-exploitation dilemma. The reasoning behind this claim is that over time, the agent will become more and more certain about the hidden state x and hence the effect of the information gain is expected to vanish. Then, the only driving factor will be the extrinsic term, that causes the agent to achieve its objectives.

Combining the (approximated) EFE with the VFE yields a full control algorithm: In every step, the agent first minimizes the VFE to obtain a variational approximation of the hidden state posterior. Given this estimate of the hidden state, the agent then minimizes the EFE to obtain a variational distribution over policies to follow for the next time step. Here, the nature of the EFE ensures that the agent is both following extrinsic motives, as well as gathering information about its environment efficiently.

3.3 Free Energy of the Expected Future

An issue of the EFE is that its formal origin is unclear. While its formulation (Eq. (3.12)) looks similar to the definition of the VFE (Eq. (3.9)), it cannot be directly derived from it [43]. We discuss this issue in detail in Section 4.3.

In an attempt to find a more naturally grounded objective and thereby tackling this issue, Millidge, Tschantz, and Buckley [43] propose a novel objective function: the Free Energy of the Expected Future. They argue that an agent should always act to minimize the difference between the outcomes it expects and the outcomes it desires. Mathematically, they formulate this intuition as minimizing the KL divergence between the agent's model of the expected outcomes of a policy $q_\phi(o_{t+1:T}, x_{t+1:T} \mid \pi)$ and a model of the agent's desired outcomes $p(o_{t+1:T}, x_{t+1:T})$:

$$\pi^* = \arg\min_\pi \ \text{FEEF}(\pi)$$
$$\text{FEEF}(\pi) = D_{\text{KL}}[q_\phi(o_{t+1:T}, x_{t+1:T} \mid \pi) \parallel p(o_{t+1:T}, x_{t+1:T})]$$

One of the key properties of the FEEF is that it decomposes into an extrinsic and epistemic term, similar to the EFE. However, in order for this decomposition to work, Approximation (3.14) has to be made here as well.

$$p(x_{t+1:T} \mid o_{t+1:T}) \approx q_\phi(x_{t+1:T} \mid o_{t+1:T}, \pi)$$

Using the above assumptions and writing $\tilde{o} = o_{t+1:T}$ and $\tilde{x} = x_{t+1:T}$ for simplicity, the

FEEF can be decomposed as follows:

$$
\begin{aligned}
\text{FEEF}\,(\pi) &= D_{\text{KL}}[q_\phi(\tilde{o}, \tilde{x} \mid \pi) \parallel p(\tilde{o}, \tilde{x})] \\
&= \mathbb{E}_{q_\phi(\tilde{o}, \tilde{x} \mid \pi)}\left[\ln \frac{q_\phi(\tilde{o} \mid \tilde{x})\, q_\phi(\tilde{x} \mid \pi)}{p(\tilde{x} \mid \tilde{o})\, p(\tilde{o})}\right] \\
&= \mathbb{E}_{q_\phi(\tilde{o}, \tilde{x} \mid \pi)}[\ln q_\phi(\tilde{o} \mid \tilde{x}) - \ln p(\tilde{o}) - \ln p(\tilde{x} \mid \tilde{o}) + \ln q_\phi(\tilde{x} \mid \pi)] \\
&\approx \mathbb{E}_{q_\phi(\tilde{o}, \tilde{x} \mid \pi)}[\ln q_\phi(\tilde{o} \mid \tilde{x}) - \ln p(\tilde{o}) - \ln q_\phi(\tilde{x} \mid \tilde{o}, \pi) + \ln q_\phi(\tilde{x} \mid \pi)] \\
&= \mathbb{E}_{q_\phi(\tilde{x} \mid \pi)}\Big[\mathbb{E}_{q_\phi(\tilde{o} \mid \tilde{x})}[\ln q_\phi(\tilde{o} \mid \tilde{x}) - \ln p(\tilde{o})]\Big] \\
&\quad - \mathbb{E}_{q_\phi(\tilde{o} \mid \pi)}\Big[\mathbb{E}_{q_\phi(\tilde{x} \mid \tilde{o}, \pi)}[\ln q_\phi(\tilde{x} \mid \tilde{o}, \pi) - \ln q_\phi(\tilde{x} \mid \pi)]\Big] \\
&= \underbrace{\mathbb{E}_{q_\phi(\tilde{x} \mid \pi)}[D_{\text{KL}}[q_\phi(\tilde{o} \mid \tilde{x}) \parallel p(\tilde{o})]]}_{\text{extrinsic term}} \\
&\quad - \underbrace{\mathbb{E}_{q_\phi(\tilde{o} \mid \pi)}[D_{\text{KL}}[q_\phi(\tilde{x} \mid \tilde{o}, \pi) \parallel q_\phi(\tilde{x} \mid \pi)]]}_{\text{expected information gain}} \\
&=: \widetilde{\text{FEEF}}\,(\pi)
\end{aligned}
\tag{3.17}
$$

where $\widetilde{\text{FEEF}}\,(\pi)$ is the approximated FEEF.

This decomposition is very similar to the decomposition of the EFE we obtained in Eq. (3.16). Both share the expected information gain term[1], but have a different extrinsic term. The difference of the extrinsic terms is best highlighted by decomposing the KL divergence of the FEEF as follows:

$$
\mathbb{E}_{q_\phi(\tilde{x} \mid \pi)}[D_{\text{KL}}[q_\phi(\tilde{o} \mid \tilde{x}) \parallel p(\tilde{o})]] = \mathbb{E}_{q_\phi(\tilde{x} \mid \pi)}[H\left[q_\phi(\tilde{o} \mid \tilde{x})\right]] + \mathbb{E}_{q_\phi(\tilde{o} \mid \pi)}[\ln p(\tilde{o})]
$$

Hence, the FEEF uses the extrinsic term of the EFE plus the expected entropy of the observation distribution.

[1] To see that the terms are similar, one has to use the symmetry of the mutual information w.r.t. its variables o and x. The difference between the two terms is that Eq. (3.16) is factorized into time steps (using the mean-field assumption) while Eq. (3.17) is still defined over full series of o and x.

4 Formal Analysis of Active Inference

One of the central claims behind Active Inference is that it resolves the exploration-exploitation dilemma and explains curiosity mathematically [6]. In this chapter, we take a critical look at both of these claims and outline three central issues of the EFE.

First, in Section 4.1 we argue that the EFE as defined in Eq. (3.16) does not fully resolve the exploration-exploitation dilemma in practice. Second, we show that Approximation (3.14) is not feasible and changes the semantics of the EFE significantly in Section 4.2. We argue that the approximation alone causes the epistemic term to really be information seeking and show that the EFE without this approximation does not explicitly encourage exploration. Finally, in Section 4.3 we question the origin of the EFE and argue that there is a more natural extension of the VFE into the future that does not result in the information gain. To separate the arguments from Section 4.2 and Section 4.3, we assume that the approximation discussed in the former section is valid in the latter section.

Additionally, we show that Approximation (3.14) is also used in the definition of the FEEF and briefly investigate its impact.

4.1 Does the EFE resolve the exploration-exploitation dilemma?

While the EFE might provide an answer to the exploration-exploitation dilemma in theory, in practice some challenges remain. One of these challenges is that the weighting of exploration versus exploitation still has to be determined. Unlike in many RL approaches where this weighting is configured by specifying the intensity of noise on the agent's actions [36, 44–46], in the AI framework the specification of the weighting is pushed into the definition of the observation preference distribution $p(o)$. As an example, a common choice for $p(o)$ is a Gaussian distribution $\mathcal{N}(\mu, \sigma)$, which intuitively expresses that we

would like our agent to make observation μ, but allow some degree of error specified by σ. Thus, the extrinsic term becomes

$$\begin{aligned} -\mathbb{E}_{q_\phi(o_\tau \mid \pi)}[\ln p(o_\tau)] &= \mathbb{E}_{q_\phi(o_\tau \mid \pi)}\left[\frac{(o_\tau - \mu)^2}{2\sigma^2}\right] + const \\ &= \frac{1}{2\sigma^2}\mathbb{E}_{q_\phi(o_\tau \mid \pi)}\left[(o_\tau - \mu)^2\right] + const \end{aligned} \tag{4.1}$$

As visible in Eq. (4.1), the impact of the extrinsic term is directly controlled by the choice of σ. On one hand, if we choose σ too small, the agent will ignore exploration and fully focus on exploitation and, hence, might never find a good solution. On the other hand, if we choose σ too high, the agent will continue exploring even if it is already able to solve the task very well. In theory, the information gain should still eventually become zero and, thus, the extrinsic term should dominate the agent's policy selection. In practice, however, due to approximation errors it is unlikely that the information gain becomes exactly zero. Hence, choosing σ too high can cause the extrinsic term to remain insignificant and thus prevent the agent from converging towards a good solution. To summarize, the exploration-exploitation dilemma remains an issue in practice, but the EFE gives the weighting between these two objectives an interpretation.

4.2 Feasibility of Approximation (3.14)

The emergence of the information gain from the basic EFE formulation in Eq. (3.12) is often mentioned as one of its key properties [6, 42], as it offers a formal explanation for curiosity. Admittedly, there is some beauty in the theory that an agent, in order to survive, has to minimize its surprise, and, to do that, has to explore to understand the causes of its observations. This property is also one of the key contributing factors that makes AI interesting for the RL community, as it gives formal justification for using a sum of extrinsic reward and information gain as a reward term for agents. Hence, much hinges on the existence of the information gain as a direct consequence of the EFE formalism, making it reasonable to take a closer look at its formal origin.

Starting from Eq. (3.12), to arrive at the formulation that contains the information gain (Eq. (3.16)), Approximation (3.14) had to be made in Eq. (3.15). The argument behind this approximation is that the variational distribution q_ϕ approximates the original distribution p and assuming it does so effectively, we can interchange the two distributions. However, the devil lies in the detail, as the right side of the approximation ($q_\phi(x_\tau \mid o_\tau, \pi)$)

is conditioned on π while the right side ($p(x_\tau \mid o_\tau)$) is not. So intuitively, even if the agent did a perfect job in approximating p with the variational distribution, the two should only be (approximately) equal if the state x_τ is (approximately) independent of the policy π when o_τ is given.

To gain an intuition about the assumptions behind Approximation (3.14), we will now proceed as follows: First, we show that there is a tight coupling between Approximation (3.14) and the following approximation:

$$p(x_\tau) \approx q_\phi(x_\tau \mid \pi) \tag{4.2}$$

Then, assuming that the variational distribution q_ϕ approximates p well, we argue that $p(x_\tau)$ and $q_\phi(x_\tau \mid \pi)$ have substantially different intuitions and, hence, do not serve well as approximators for each other. This insight finally leads us to the conclusion that Approximation (3.14) is an infeasible approximation.

We start by applying Bayes' rule to both $p(\tilde{x} \mid \tilde{o})$ and $q_\phi(\tilde{x} \mid \tilde{o}, \pi)$.

$$p(\tilde{x} \mid \tilde{o}) = \frac{p(\tilde{o} \mid \tilde{x})\, p(\tilde{x})}{\int p(\tilde{o} \mid \tilde{x})\, p(\tilde{x})\, d\tilde{x}} \tag{4.3}$$

$$q_\phi(\tilde{x} \mid \tilde{o}, \pi) = \frac{q_\phi(\tilde{o} \mid \tilde{x})\, q_\phi(\tilde{x} \mid \pi)}{\int q_\phi(\tilde{o} \mid \tilde{x})\, q_\phi(\tilde{x} \mid \pi)\, d\tilde{x}} = \frac{p(\tilde{o} \mid \tilde{x})\, q_\phi(\tilde{x} \mid \pi)}{\int p(\tilde{o} \mid \tilde{x})\, q_\phi(\tilde{x} \mid \pi)\, d\tilde{x}} \tag{4.4}$$

where we used that the variational distribution follows the same observation model as the preference distribution: $q_\phi(\tilde{o} \mid \tilde{x}) = p(\tilde{o} \mid \tilde{x})$.

In Eq. (4.3), $p(\tilde{x})$ can be seen as the hidden state preferences of the agent. Usually, in AI preferences are defined over observations, by specifying a target observation model $p(\tilde{o})$. Due to the factorization of p, specifying $p(\tilde{o})$ implies the existence of a hidden state distribution $p(\tilde{x})$ that fulfills

$$p(\tilde{o}) = \int p(\tilde{o} \mid \tilde{x})\, p(\tilde{x})\, d\tilde{x} \tag{4.5}$$

Note that, depending on the observation likelihood $p(\tilde{o} \mid \tilde{x})$, there can be multiple or no $p(\tilde{x})$ that fulfil the above constraint. Hence, specifying $p(\tilde{o})$ does not fully determine the joint preference distribution $p(\tilde{o}, \tilde{x})$, while specifying $p(\tilde{x})$ does. In the following, we make no further assumptions on whether the preferences were defined over observations or hidden states and simply assume that $p(\tilde{o})$ and $p(\tilde{x})$ are defined in a manner consistent with Eq. (4.5).

From Eqs. (4.3) and (4.4) we can immediately infer that if Approximation (4.2) is exact, then Approximation (3.14) is exact:

$$(\forall \tilde{x} :\ p(x_\tau) = q_\phi(x_\tau \,|\, \pi)) \quad \Rightarrow \quad (\forall \tilde{x}, \tilde{o} :\ p(x_\tau \,|\, o_\tau) = q_\phi(x_\tau \,|\, o_\tau, \pi))$$

The other direction is not generally true, but holds under the condition that the preference posterior is nonzero everywhere, that is

$$(\forall \tilde{x}, \tilde{o} :\ p(\tilde{x} \,|\, \tilde{o}) \neq 0 \wedge p(x_\tau \,|\, o_\tau) = q_\phi(x_\tau \,|\, o_\tau, \pi)) \quad \Rightarrow \quad (\forall \tilde{x} :\ p(x_\tau) = q_\phi(x_\tau \,|\, \pi))$$

which we will prove in the following.

We start with the following equation that highlights the relation between the two distributions:

$$\begin{aligned} q_\phi(\tilde{x} \,|\, \tilde{o}, \pi) &= \frac{q_\phi(\tilde{o} \,|\, \tilde{x})\, q_\phi(\tilde{x} \,|\, \pi)}{q_\phi(\tilde{o} \,|\, \pi)} = \frac{p(\tilde{o} \,|\, \tilde{x})\, q_\phi(\tilde{x} \,|\, \pi)}{q_\phi(\tilde{o} \,|\, \pi)} \\ &= p(\tilde{x} \,|\, \tilde{o})\, \frac{q_\phi(\tilde{x} \,|\, \pi)}{p(\tilde{x})} \frac{p(\tilde{o})}{q_\phi(\tilde{o} \,|\, \pi)} \end{aligned} \tag{4.6}$$

where we again assumed that the variational distribution and the target distribution follow the same observation model: $q_\phi(\tilde{o} \,|\, \tilde{x}) = p(\tilde{o} \,|\, \tilde{x})$.

Based on Eq. (4.6) we can now derive under which circumstances Approximation (3.14) becomes exact. That is, for any pair $(\tilde{o}, \tilde{x})$ Approximation (3.14) is exact iff one of two conditions is fulfilled:

$$p(\tilde{x} \,|\, \tilde{o}) = 0 \qquad \text{or} \qquad \frac{q_\phi(\tilde{x} \,|\, \pi)}{p(\tilde{x})} \frac{p(\tilde{o})}{q_\phi(\tilde{o} \,|\, \pi)} = 1$$

or equivalently:

$$\forall \tilde{o}\ \forall \tilde{x} \quad p(\tilde{x} \,|\, \tilde{o}) \neq 0 \quad \Rightarrow \quad q_\phi(\tilde{x} \,|\, \pi) = \alpha(\tilde{o})\, p(\tilde{x}) \tag{4.7}$$

where we write $\alpha(\tilde{o}) \coloneqq \frac{q_\phi(\tilde{o} \,|\, \pi)}{p(\tilde{o})}$.

Intuitively, for a given observation, Eq. (4.7) can now be interpreted as follows: Within the set of hidden states that have nonzero probability under the posterior, the hidden state is independent of the policy. Or to phrase it differently: the hidden state is independent of the policy, given that it has nonzero probability under the posterior.

A common choice for variational distributions are Gaussians [23, 28, 47], which, among other distributions, fulfill the condition $p(\tilde{x} \mid \tilde{o}) \neq 0$ for all $\tilde{x}$ and $\tilde{o}$. As a result, for these kinds of distributions, Approximation (3.14) ($p(\tilde{x} \mid \tilde{o}) \approx q_\phi(\tilde{x} \mid \tilde{o}, \pi)$) is exact iff Approximation (4.2) is exact ($p(\tilde{x}) \approx q_\phi(\tilde{x} \mid \pi)$).

While this argument shows that there is a strong relation between the two approximations, it makes no statement about how the approximation error of one affects the other. To address this shortcoming, the following equation shows that the expected relative error of Approximation (3.14) is equal to the relative error of Approximation (4.2):

$$\mathbb{E}_{q_\phi(\tilde{o} \mid \pi)}\left[\frac{q_\phi(\tilde{x} \mid \tilde{o}, \pi)}{p(\tilde{x} \mid \tilde{o})}\right] = \mathbb{E}_{q_\phi(\tilde{o} \mid \pi)}\left[\frac{q_\phi(\tilde{x} \mid \pi)}{p(\tilde{x})}\frac{p(\tilde{o})}{q_\phi(\tilde{o} \mid \pi)}\right] = \frac{q_\phi(\tilde{x} \mid \pi)}{p(\tilde{x})}$$

where $\tilde{x}$ is arbitrary.

To summarize, we have shown that there is a tight coupling between Approximations (3.14) and (4.2). Any approximation error of Approximation (4.2) directly translates to approximation error of Approximation (3.14). Hence, the infeasibility of the latter can be directly concluded from the infeasibility of the former. It remains to show that Approximation (4.2) is indeed an infeasible approximation, independent of the quality of the variational posterior, which we will do in the following.

To separate the error stemming from Approximation (4.2) from the variational approximation error, we assume that the agent did a perfect job in approximating p with the variational distribution, transforming Approximation (4.2) into the following approximation:

$$p(x_\tau \mid \pi) \approx p(x_\tau) \tag{4.8}$$

This approximation is exact iff the state is independent of the policy, which means that the agent's actions have no effect on the environment.

Intuitively, the two distributions of Approximation (4.8) have entirely different meanings. To get an insight into these meanings, it is useful to first reiterate the semantics of the generative distribution p. For a single time step, p factorizes as follows:

$$p(o_\tau, x_\tau, \pi) = p(o_\tau \mid x_\tau)\, p(x_\tau \mid \pi)\, p(\pi)$$

where the dynamics $p(x_\tau \mid \pi)$ and the observation distribution $p(o_\tau \mid x_\tau)$ are pre-defined and fixed. Since all other distributions are fixed, it becomes apparent that $p(\pi)$ is not a

prior that we can freely choose, as it is restricted by our choice of $p(o_\tau)$:

$$\begin{aligned} p(o_\tau) &= \iint p(o_\tau, x_\tau, \pi)\, dx_\tau d\pi \\ &= \iint p(o_\tau \mid x_\tau)\, p(x_\tau \mid \pi)\, p(\pi)\, dx_\tau d\pi \end{aligned} \tag{4.9}$$

Hence, determining $p(\pi)$ is equivalent to finding a policy distribution that generates our target observation distribution exactly. Any $p(\pi)$ that fulfills Eq. (4.9) could hence be seen as the optimal distribution the agent should choose its policy from.

If we now write down the integral to compute $p(x_\tau)$,

$$\begin{aligned} p(x_\tau) &= \int p(x_\tau, \pi)\, d\pi \\ &= \int p(x_\tau \mid \pi)\, p(\pi)\, d\pi \end{aligned}$$

we can apply this insight to gain the following intuition: $p(x_\tau)$ is the marginal hidden state distribution we obtain if the agent behaves optimally in a sense that its expected observation distribution is exactly its preferred observation distribution.

The distribution we are using to approximate this marginal, however, is simply the dynamics distribution: $p(x_\tau \mid \pi)$ for some arbitrary policy π. This distribution has no notion of optimal behavior, as it is already conditioned on the policy. Hence, Approximation (4.8) can be precise for policies that have a very high probability in $p(\pi)$, but not in the general case.

Thus, we argue that Approximation (4.2) is not generally feasible. Due to their tight coupling, we further conclude that Approximation (3.14), which was used to transform the epistemic term into the information gain in Eq. (3.15), is also infeasible. Note that we did not make any assumptions about the quality of the variational posterior $q_\phi(\tilde{x} \mid \tilde{o}, \pi)$. Instead, even if we assume a perfect variational posterior, the infeasibility of Approximation (3.14) can be derived purely from the fact that one distribution is conditioned on the policy while the other one is not. While the conditioning on the policy might appear to be a minor detail at first glance, it eliminates the agent's preferences and, hence, changes the semantics of the distribution significantly.

The infeasibility of these approximations raises the question whether the epistemic term is really information seeking or rather fulfills another function. Another way to look at

this term is to directly write it as a difference of KL divergences:

$$
\begin{aligned}
&- \mathbb{E}_{q_\phi(o_\tau, x_\tau \mid \pi)}[\ln p(x_\tau \mid o_\tau) - \ln q_\phi(x_\tau \mid \pi)] \\
&= \mathbb{E}_{q_\phi(x_\tau \mid \pi)}\Big[\ln q_\phi(x_\tau \mid \pi) - \mathbb{E}_{q_\phi(o_\tau \mid x_\tau, \pi)}[\ln p(x_\tau \mid o_\tau)]\Big] \\
&= \mathbb{E}_{q_\phi(x_\tau \mid \pi)}\big[\ln q_\phi(x_\tau \mid \pi) - \mathbb{E}_{p(o_\tau \mid x_\tau)}[\ln p(x_\tau \mid o_\tau)]\big] \\
&= \mathbb{E}_{q_\phi(x_\tau \mid \pi)}\big[\ln q_\phi(x_\tau \mid \pi) - \mathbb{E}_{p(o_\tau \mid x_\tau)}[\ln p(o_\tau \mid x_\tau) - \ln p(o_\tau) + \ln p(x_\tau)]\big] \\
&= D_{\mathrm{KL}}[q_\phi(x_\tau \mid \pi) \parallel p(x_\tau)] - \mathbb{E}_{q_\phi(x_\tau \mid \pi)}[D_{\mathrm{KL}}[p(o_\tau \mid x_\tau) \parallel p(o_\tau)]]
\end{aligned}
\tag{4.10}
$$

The first KL divergence ensures that the agent's expected hidden state distribution under its policy $q_\phi(x_\tau \mid \pi)$ matches the target hidden state distribution $p(x_\tau)$. This is quite interesting, as the epistemic term is driving the agent towards realizing its extrinsic preferences, which is actually the purpose of the extrinsic term. The second (expected) KL divergence is in fact an information gain term, but one that seeks out hidden states that are informative about observations, not vice versa. Note that this term is not a mutual information, as $q_\phi(x_\tau \mid \pi) \neq p(x_\tau)$ and, thus, the order of x_τ and o_τ can not be switched. Interestingly, if we add the extrinsic term to Eq. (4.10), some parts cancel out and we obtain another decomposition of the EFE:

$$
\begin{aligned}
G_{\pi,\tau}(\phi) &= D_{\mathrm{KL}}[q_\phi(x_\tau \mid \pi) \parallel p(x_\tau)] - \mathbb{E}_{q_\phi(x_\tau \mid \pi)}[D_{\mathrm{KL}}[p(o_\tau \mid x_\tau) \parallel p(o_\tau)]] \\
&\quad - \mathbb{E}_{q_\phi(o_\tau, x_\tau \mid \pi)}[\ln p(o_\tau)] \\
&= D_{\mathrm{KL}}[q_\phi(x_\tau \mid \pi) \parallel p(x_\tau)] \\
&\quad - \mathbb{E}_{q_\phi(o_\tau, x_\tau \mid \pi)}[\ln p(o_\tau \mid x_\tau) - \ln p(o_\tau) + \ln p(o_\tau)] \\
&= D_{\mathrm{KL}}[q_\phi(x_\tau \mid \pi) \parallel p(x_\tau)] + \mathbb{E}_{q_\phi(x_\tau \mid \pi)}[H[p(o_\tau \mid x_\tau)]]
\end{aligned}
\tag{4.11}
$$

While this decomposition cannot be computed as $p(x_\tau)$ is usually not explicitly defined, it consists of parts that all have a clear meaning. The first term ensures the realization of latent preferences and the second term ensures that the expected observation entropy is as low as possible. Wanting a low observation entropy is reasonable as it allows the agent to better predict observations and hence reduces surprise. Yet, none of these terms seem to promote exploration, which gives reason to believe that the sole cause for the expected information gain term in Eq. (3.16) to appear is Approximation (3.14).

In order to fully understand the impact of Approximation (3.14) on the EFE, we can follow

Da Costa et al. [48] and rewrite its exact formulation (Eq. (3.12)) as

$$\begin{aligned} G_{\pi,\tau}(\phi) &= -\mathbb{E}_{q_\phi(o_\tau, x_\tau \mid \pi)}[\ln p(o_\tau, x_\tau) - \ln q_\phi(x_\tau \mid \pi)] \\ &= -\mathbb{E}_{q_\phi(o_\tau, x_\tau \mid \pi)}[\ln q_\phi(x_\tau \mid o_\tau, \pi) - \ln q_\phi(x_\tau \mid \pi) + \ln p(o_\tau) \\ &\quad - \ln q_\phi(x_\tau \mid o_\tau, \pi) + \ln p(x_\tau \mid o_\tau)] \\ &= -\mathbb{E}_{q_\phi(o_\tau \mid \pi)}[D_{\mathrm{KL}}[q_\phi(o_\tau \mid x_\tau, \pi) \parallel q_\phi(o_\tau \mid \pi)]] - \mathbb{E}_{q_\phi(o_\tau \mid \pi)}[\ln p(o_\tau)] \\ &\quad + \mathbb{E}_{q_\phi(o_\tau \mid \pi)}[D_{\mathrm{KL}}[q_\phi(x_\tau \mid o_\tau, \pi) \parallel p(x_\tau \mid o_\tau)]] \\ &\geq -\mathbb{E}_{q_\phi(x_\tau \mid \pi)}[D_{\mathrm{KL}}[q_\phi(o_\tau \mid x_\tau, \pi) \parallel q_\phi(o_\tau \mid \pi)]] - \mathbb{E}_{q_\phi(o_\tau \mid \pi)}[\ln p(o_\tau)] \\ &= \tilde{G}_{\pi,\tau}(\phi) \end{aligned} \tag{4.12}$$

where $\tilde{G}_{\pi,\tau}(\phi)$ is the approximated EFE as defined in Eq. (3.16).

From Eq. (4.12) we can immediately see that by minimizing $\tilde{G}_{\pi,\tau}(\phi)$ we are actually minimizing a lower bound of the EFE. The difference between $\tilde{G}_{\pi,\tau}(\phi)$ and $G_{\pi,\tau}(\phi)$ is given by the KL divergence between $q_\phi(x_\tau \mid o_\tau, \pi)$ and $p(x_\tau \mid o_\tau)$. By our argument above, this KL divergence could be arbitrarily high, especially if $p(\pi)$ is low. Hence, it is not clear whether minimizing $\tilde{G}_{\pi,\tau}(\phi)$ results in any minimization of the actual objective $G_{\pi,\tau}(\phi)$.

This insight leads us to conclude that the minimization of the information gain does not generally contribute to the minimization of the exact EFE. Rather it seems as if Approximation (3.14) is the sole reason for the EFE neatly decomposing into an extrinsic term and an explicitly information seeking term. The question that naturally arises is whether the EFE in its exact formulation even exhibits information seeking behavior at all. Definitively answering this question is not easily possible, as it is hard to quantify which behavior is explicitly information seeking and which is not. Instead, we provide a comparison of the behaviors induced by the exact EFE and the approximated EFE on a discrete example system in Fig. 4.1. Although no general statement is possible from this singular example, it shows that it is straightforward to find cases in which the exact EFE exhibits no information seeking behavior while the exact EFE does.

Comparison of exact versus approximated EFE on a discrete system

In the following we present a simple discrete environment in which the approximated EFE induces substantially different behavior than the exact EFE. Specifically, we will see that the agent following the approximated EFE exhibits curious behavior, while the agent following the exact EFE does not.

In this example, which is visualized in Fig. 4.1, the agent is a mouse that can visit two possible locations, denoted s_1 and s_2. At the beginning, the mouse is situated on s_1 and can decide whether it wants to *stay* on s_1 or *move* to s_2. If the mouse decides to move to s_2, it might or might not encounter a mouse trap, which has a 20% probability of killing it. Since the mouse is short-sighted, the only way of knowing whether there is a mouse trap on s_2 or not is by moving to s_2 and checking while risking death in the process. To simplify this example, we assume that the mouse can only take a single step and has full knowledge of the system dynamics and observation distribution. The only extrinsic objective the mouse has is to stay alive.

This example system is specifically designed to give the agent one option that has a high extrinsic reward (staying on s_1 ensures the survival of the mouse) and one option that is informative (moving to s_2 allows the mouse to reduce uncertainty about the presence of the trap). Note that for an agent trying to maximize purely the extrinsic reward, there is no reason to ever move to s_2 as knowing about the trap does not facilitate a better strategy. Simply staying on s_1 is always the (extrinsically) optimal strategy regardless of the presence of a trap. Hence, having this setup allows us to distinguish curious behavior from non-curious behavior in a clear manner, as the only reason for following the *move* policy is information gathering and the only reason for the *stay* policy is maximization of extrinsic reward.

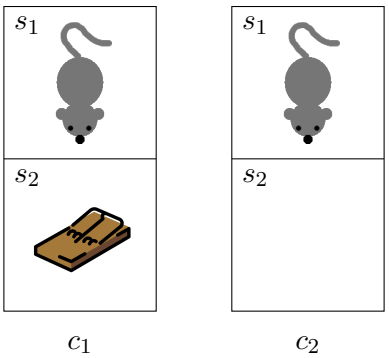


Figure 4.1: Visualization of the example system. The mouse is situated in one of two contexts, denoted c_1 and c_2, but has no prior knowledge which one it is currently in. Both contexts are equal with the exception of c_1 having a trap on location s_2, while s_2 in context c_2 is empty. The mouse starts on location s_1 and has to make a decision whether to move to s_2 or stay where it is. Staying on s_1 is the safer but less informative option, as it does not reveal any information about the context. Moving to s_2 allows the mouse to check for the trap and, hence, determine the context but might result in its death.

For the formal modelling of this system, we require the following symbols:

$$\begin{aligned}
&\Pi \coloneqq \{\textit{stay}, \textit{move}\} \\
&X \coloneqq X_C \times X_S, && X_C \coloneqq \{c_1, c_2\}, && X_S \coloneqq \{s_1, s_2\} \\
&O \coloneqq O_T \times X_A, && O_T \coloneqq \{\textit{see trap}, \textit{no trap}\}, && O_A \coloneqq \{\textit{alive}, \textit{dead}\}
\end{aligned}$$

where Π denotes the set of actions the agent can take, X is the set of latent states the system can assume and O is the set of observations the agent can make. As shown in Fig. 4.1, c_1 denotes the context in which there is a trap in front of the mouse and c_2 is the context without a trap.

The generative model of this system is now given as

$$\begin{aligned}
&p((o_t, o_a), (x_c, x_s) \mid \pi) = p(o_t \mid x_c, x_s)\, p(o_a \mid x_c, x_s)\, p(x_c)\, p(x_s \mid \pi) \\
&\quad \forall (o_t, o_a) \in O, (x_c, x_s) \in X, \pi \in \Pi
\end{aligned}$$

where we set the probability of a mouse trap being in front of the mouse to 50% and define the individual distributions as

$p(x_c)$	c_1	c_2
	0.5	0.5

$p(x_s \mid \pi)$	s_1	s_2
stay	1	0
move	0	1

$p(o_t \mid x_c, x_s)$	*see trap*	*no trap*
(c_1, s_1)	0.5	0.5
(c_1, s_2)	0.99	0.01
(c_2, s_1)	0.5	0.5
(c_2, s_2)	0.01	0.99

$p(o_a \mid x_c, x_s)$	*alive*	*dead*
(c_1, s_1)	0.99	0.01
(c_1, s_2)	0.8	0.2
(c_2, s_1)	0.99	0.01
(c_2, s_2)	0.99	0.01

Finally, we define the extrinsic preferences of the mouse such that it avoids dying and is indifferent to seeing a trap by setting

$$p(o_t, o_a) = p(o_t)\, p(o_a)$$

$p(o_t)$	*see trap*	*no trap*
	0.5	0.5

$p(o_a)$	*alive*	*dead*
	0.99	0.01

Two aspects of these definitions might appear curious on the first sight, but actually have a very specific purpose. First, in the observation distributions we use 0.01 and 0.99 instead of 0 and 1 in many places. The reason for this is that the information gain is only defined if every possible observation has non-zero probability under every possible hidden state, as otherwise we would have to compute the logarithm of 0. Hence, we give every possible observation some probability under the observation distributions. Intuitively, this means that the mouse has some probability of seeing the trap if it is not there and vice versa, and there is also some probability that the mouse dies despite staying on s_1.

Second, the probability of seeing a trap on s_1 is 0.5, independent of the context. The reason for setting this probability to 0.5 is again of a technical nature. In order to properly compare the approximated EFE to the exact EFE, the environment must be defined in such a way that a hidden preference distribution $p(x)$ exists which is consistent with the observation preference distribution $p(o)$. Thus, the following constraint must be fulfilled by the environment:

$$\exists p(x) \quad p(o) = \sum_x p(o \mid x)\, p(x) \tag{4.13}$$

If we simply set $p(o_t = \textit{see trap} \mid x_c = c_i, x_s = s_1) = 0.01$, this constraint is not fulfilled, meaning that there is no hidden preference distribution consistent with the observation preference distribution. By setting $p(o_t = \textit{see trap} \mid x_c = c_i, x_s = s_1) = 0.5$ we ensure that the hidden preference distribution exists. Note that although it might seem strange that the mouse can sense a mouse trap at s_1, it cannot gather any information from that sensation as it is completely random.

Finally, we set the preference of staying alive to only 0.99 instead of 1. The reason for this is a combination of the two points we discussed before. Since we set the probability of surviving to 0.99 and not 1 in s_1, there is no state in which the mouse is certain to survive. Hence, if we set $p(\textit{alive}) = 1$, then there would be no hidden preference distribution that fulfills Constraint (4.13).

With the environment definition complete, we can now compute the unique solution of the equation system induced by Constraint (4.13) to obtain the hidden preference distribution:

$p(x_c, x_s)$	s_1	s_2
c_1	0.5	0.0
c_2	0.5	0.0

Using Eq. (3.16) and Eq. (4.11) we can compute both the approximated EFE $\tilde{G}_\pi(\phi)$ and the exact EFE $G_\pi(\phi)$ for each policy.

	$\pi = \textit{stay}$	$\pi = \textit{move}$
$\tilde{G}_\pi(\phi)$	1.50	1.29
$G_\pi(\phi)$	1.53	23.88

where we are again assuming that variational distributions are exact, to exclude this source of approximation error.

Plugging the free energy values into Eq. (3.11) and assuming a uniform policy prior, we obtain the following policy distributions:

$q^*_\phi(\pi)$	*stay*	*move*
approx. EFE	0.45	0.55
exact EFE	1.00	0.00

We can see that the approximated EFE slightly prefers the more informative *move* policy, which is reasonable as it is composed of both an extrinsic term and the information gain. The exact EFE, on the other hand, almost[1] exclusively chooses the *stay* option and does not exhibit any information seeking behavior. Hence, in this example, the curious behavior was not naturally induced by the EFE but rather arose from Approximation (3.14). Given that the reason for the information gain to appear in the EFE formulation is Approximation (3.14), this observation is not surprising and further confirms our previous arguments about the infeasibility of Approximation (3.14).

4.3 Origin of the EFE

Setting aside the arguments made in the previous section, the question about the origin of the EFE as it is defined in Eq. (3.12) remains. At first glance it looks like G_π was obtained from the definition of F_π (Eq. (3.9)) by simply taking the expectation over unknown future observations,

$$\begin{aligned}&\mathbb{E}_{q_\phi(o_{t+1:T} \mid \pi)}[F_\pi(o_{t+1:T}, \phi)] \\ &= -\mathbb{E}_{q_\phi(o_{t+1:T} \mid \pi) q_\phi(x_{t+1:T} \mid \pi)}\left[\ln p(o_{t+1:T}, x_{t+1:T} \mid \pi) - \sum_{\tau=t}^{T} \ln q_\phi(x_\tau \mid \pi)\right]\end{aligned}$$

However, there are some obvious differences between the above term and the definition of G_π. One difference is that we are taking the expectation w.r.t. $q_\phi(o_{t+1:T} \mid \pi)\, q_\phi(x_{t+1:T} \mid \pi)$ instead of $q_\phi(o_{t+1:T}, x_{t+1:T} \mid \pi)$. Here one could argue as follows: during the minimization of $F(o_{1:t}, \phi)$ w.r.t. ϕ, the variational posterior $q_\phi(x_{1:t} \mid \pi)$ is actually trained to minimize the KL divergence to the posterior $p(x_{1:t} \mid o_{1:t}, \pi)$ (see Eq. (3.8)). While $q_\phi(x_{1:t} \mid \pi)$ is not explicitly conditioned on $o_{1:t}$, if we change $o_{1:t}$, the optimization of $F(o_{1:t}, \phi)$ w.r.t. ϕ will yield different distributions $q_\phi(x_{1:t} \mid \pi)$ as result. Hence, although $q_\phi(x_{1:t} \mid \pi)$ is not conditioned on $o_{1:t}$, the optimal variational posterior $q^*(x_{1:t} \mid o_{1:t}, \pi)$ is

$$\begin{aligned} q^*(x_{1:t} \mid o_{1:t}, \pi) &\coloneqq q_{\phi^*(o_{1:t})}(x_{1:t} \mid \pi) \\ \text{where} \qquad \phi^*(o_{1:t}) &= \arg\min_\phi F(o_{1:t}, \phi) \end{aligned} \tag{4.14}$$

The crucial difference between past and future is that observations we made in the past already happened and are thus fixed. So instead of finding a (conditional) variational

[1] Note that these numbers are rounded.

posterior that approximates the real posterior for all possible observations, we can resort to finding a (marginal) variational posterior that is a good approximation for just one observation. In the future, however, we need a variational posterior for a variety of possible observations we could make. One way to view this issue is as a nested optimization problem, where we find the optimal variational parameters for each possible observation:

$$\mathbb{E}_{q_\phi(o_{t+1:T} \mid \pi)} \left[\min_{\hat{\phi}} F_\pi \left(o_{t+1:T}, \hat{\phi} \right) \right]$$

This view neatly visualizes the idea that the agent, while planning ahead, not only thinks about what observations it is going to encounter, but also how its belief is going to change. Using Eq. (4.14), we obtain

$$\begin{aligned}
&\mathbb{E}_{q_\phi(o_{t+1:T} \mid \pi)} \left[\min_{\hat{\phi}} F_\pi \left(o_{t+1:T}, \hat{\phi} \right) \right] \\
&= \mathbb{E}_{q_\phi(o_{t+1:T} \mid \pi)} [F_\pi \left(o_{t+1:T}, \phi^* \left(o_{t+1:T} \right) \right)] \\
&= -\mathbb{E}_{q_\phi(o_{t+1:T} \mid \pi) q_{\phi^* \left(o_{t+1:T} \right)}(x_{t+1:T} \mid \pi)} \left[\ln p(o_{t+1:T}, x_{t+1:T} \mid \pi) - \sum_{\tau=t}^{T} \ln q_{\phi^*(o_{t+1:T})}(x_\tau \mid \pi) \right]
\end{aligned}$$

Instead of finding the optimal variational parameters for each possible observation, we can also view this problem as finding the optimal parameters $\tilde{\phi}$ of a variational distribution conditioned on $o_{t+1:T}$:

$$q_{\tilde{\phi}}(x_{t+1:T} \mid o_{t+1:T}, \pi) \coloneqq q_{\phi^*(o_{t+1:T})}(x_{t+1:T} \mid \pi)$$

To avoid cluttering the notation, we assume that $\tilde{\phi}$ is part of ϕ and write

$$\begin{aligned}
&\mathbb{E}_{q_\phi(o_{t+1:T} \mid \pi)} \left[\min_{\hat{\phi}} F_\pi \left(o_{t+1:T}, \hat{\phi} \right) \right] \\
&= -\mathbb{E}_{q_\phi(o_{t+1:T} \mid \pi) q_\phi(x_{t+1:T} \mid o_{t+1:T}, \pi)} \left[\ln p(o_{t+1:T}, x_{t+1:T} \mid \pi) - \sum_{\tau=t}^{T} \ln q_\phi(x_\tau \mid o_{t+1:T}, \pi) \right] \\
&= -\mathbb{E}_{q_\phi(o_{t+1:T}, x_{t+1:T} \mid \pi)} \left[\ln p(o_{t+1:T}, x_{t+1:T} \mid \pi) - \sum_{\tau=t}^{T} \ln q_\phi(x_\tau \mid o_\tau, \pi) \right] \qquad (4.15)
\end{aligned}$$

where we made use of the mean-field assumption in the last step.

While similar to Eq. (3.12), we are now taking the expectation over the variational joint instead of two marginals, another difference arose: the final term in Eq. (4.15) is now conditioned on the observations. We will get back to this issue once we discussed the decomposition of Eq. (4.15) into individual time steps.

Taking the mean-field assumption we made about the variational distribution into account, we can decompose Eq. (3.12) into individual time steps:

$$\begin{aligned}\hat{G}_\pi(\phi) &= \sum_{\tau=1}^{T} \hat{G}_{\pi,\tau}(\phi) \\ \hat{G}_{\pi,\tau}(\phi) &= -\mathbb{E}_{\hat{q}(o_\tau, x_\tau \mid \pi)\hat{q}(x_{\tau-1} \mid \pi)}[\ln p(o_\tau, x_\tau \mid x_{\tau-1}, \pi) - \ln \hat{q}(x_\tau \mid o_\tau, \pi)]\end{aligned}$$

This decomposition shows that the difference between $\hat{G}_\pi(\phi)$ and $G_\pi(\phi)$ is now twofold. To get from $\hat{G}_\pi(\phi)$ to $G_\pi(\phi)$, we first have to assume that each state is independent of the previous state and the policy, that is $p(x_\tau \mid x_{\tau-1}, \pi) = p(x_\tau)$. With this assumption, we obtain

$$\hat{G}_{\pi,\tau}(\phi) = -\mathbb{E}_{\hat{q}(o_\tau, x_\tau \mid \pi)}[\ln p(o_\tau, x_\tau) - \ln \hat{q}(x_\tau \mid o_\tau, \pi)] \tag{4.16}$$

In order to get an intuition about the effect of this assumption, it is useful to take a close look at the two distributions being interchanged with one another. The first distribution, $p(o_\tau, x_\tau \mid x_{\tau-1}, \pi)$, is simply the forward distribution of state and observation for a fixed policy π. As the policy is fixed, there is no preference encoded in this distribution, and it is fully defined by merely the dynamics and observation distribution:

$$p(o_\tau, x_\tau \mid x_{\tau-1}, \pi) = p(o_\tau \mid x_\tau)\, p(x_\tau \mid x_{\tau-1}, \pi)$$

The second distribution, $p(o_\tau, x_\tau)$ is conditioned neither on the policy, nor on the previous state and is hence fully defined by the agent's observation preferences and the posterior:

$$p(o_\tau, x_\tau) = p(x_\tau \mid o_\tau)\, p(o_\tau)$$

It is important to note that this distribution is completely oblivious of the agent's current state. That means, it will give high probability to desired states that the agent might be completely unable to reach from its current state. Contrary to that, if we took into account the previous state (that is $p(o_\tau, x_\tau \mid x_{\tau-1})$), the distribution would still encode the agent's preferences, but would only give high probability to states that can be reached from the previous state $x_{\tau-1}$. To summarize, the first distribution is an unbiased model over future observations and states given some policy, and the second distribution encodes the agent's preferences but has no concept of the dynamics.

Apart from this independence assumption, the second difference between $\hat{G}_\pi$ and G_π is that the right term of Eq. (4.16) is conditioned on o_τ, while its equivalent in Eq. (3.12) is not. This difference has been discussed by Millidge, Tschantz, and Buckley [43], where they call $\hat{G}_\pi$ the Free Energy of the Future (FEF). Instead of taking the path over the expectation of the future free energy, they derive $\hat{G}_\pi$ as a variational bound on the expected surprise [2]:

$$\begin{aligned}
-\mathbb{E}_{\hat{q}(o_\tau \mid \pi)}[\ln p(o_\tau)] &= -\mathbb{E}_{\hat{q}(o_\tau \mid \pi)}\left[\ln \int p(o_\tau, x_\tau)\, dx_\tau\right] \\
&= -\mathbb{E}_{\hat{q}(o_\tau \mid \pi)}\left[\ln \int p(o_\tau, x_\tau)\, \frac{\hat{q}(x_\tau \mid o_\tau, \pi)}{\hat{q}(x_\tau \mid o_\tau, \pi)} dx_\tau\right] \\
&\leq -\mathbb{E}_{\hat{q}(o_\tau \mid \pi)}\left[\mathbb{E}_{\hat{q}(x_\tau \mid o_\tau, \pi)}\left[\ln \frac{p(o_\tau, x_\tau)}{\hat{q}(x_\tau \mid o_\tau, \pi)}\right]\right] \\
&= -\mathbb{E}_{\hat{q}(o_\tau, x_\tau \mid \pi)}\left[\ln \frac{p(o_\tau, x_\tau)}{\hat{q}(x_\tau \mid o_\tau, \pi)}\right] \\
&= \hat{G}_{\pi,\tau}\left(\phi\right)
\end{aligned}$$

Millidge, Tschantz, and Buckley [43] show that the EFE is a lower bound on the expected surprise if the variational posterior approximates the real posterior well:

$$\begin{aligned}
G_\pi\left(\phi\right) &= \underbrace{-\mathbb{E}_{q_\phi(o_\tau \mid \pi)}[\ln p(o_\tau)]}_{\text{expected surprise}} + \underbrace{\mathbb{E}_{q_\phi(o_\tau \mid \pi)}[D_{\mathrm{KL}}[q_\phi(x_\tau \mid o_\tau, \pi) \parallel p(x_\tau \mid o_\tau)]]}_{\text{posterior approximation error}} \\
&\quad - \underbrace{\mathbb{E}_{q_\phi(o_\tau \mid \pi)}[D_{\mathrm{KL}}[q_\phi(x_\tau \mid o_\tau, \pi) \parallel q_\phi(x_\tau \mid \pi)]]}_{\text{expected information gain}} \\
&\approx -\mathbb{E}_{q_\phi(o_\tau \mid \pi)}[\ln p(o_\tau)] - \mathbb{E}_{q_\phi(o_\tau \mid \pi)}[D_{\mathrm{KL}}[q_\phi(x_\tau \mid o_\tau, \pi) \parallel q_\phi(x_\tau \mid \pi)]] \\
&\leq -\mathbb{E}_{q_\phi(o_\tau \mid \pi)}[\ln p(o_\tau)]
\end{aligned}$$

where we used Approximation (3.14) in the second step.

As a result, maximizing the expected information gain actually makes the EFE a looser bound on the expected surprise.

[2] Note, that in their version, the variational posterior is not conditioned on the policy π, but they later assume $\hat{q}(x_\tau \mid o_\tau, \pi) \approx \hat{q}(x_\tau \mid o_\tau)$, which makes these formulations basically identical.

Furthermore, they note that the Free Energy of the Future (FEF) can be rewritten as

$$\begin{aligned}\hat{G}_\pi(\phi) &= -\mathbb{E}_{\hat{q}(o_\tau, x_\tau \mid \pi)}[\ln p(o_\tau, x_\tau) - \ln \hat{q}(x_\tau \mid o_\tau, \pi)] \\ &= -\mathbb{E}_{\hat{q}(o_\tau, x_\tau \mid \pi)}[\ln p(o_\tau, x_\tau) - \ln \hat{q}(x_\tau \mid \pi)] \\ &\quad + \mathbb{E}_{q_\phi(o_\tau \mid \pi)}[D_{\mathrm{KL}}[q_\phi(x_\tau \mid o_\tau, \pi) \parallel q_\phi(x_\tau \mid \pi)]] \\ &= G_\pi(\phi) + \underbrace{\mathbb{E}_{q_\phi(o_\tau \mid \pi)}[D_{\mathrm{KL}}[q_\phi(x_\tau \mid o_\tau, \pi) \parallel q_\phi(x_\tau \mid \pi)]]}_{\text{expected information gain}}\end{aligned}$$

Hence, the EFE is approximately the FEF plus expected information gain. It is worth noting that the FEF has no information seeking term and instead is purely purely striving to reduce expected surprise and expected variational approximation error.

To conclude, we have shown two ways in which the FEF, not the EFE, can be seen as an extension of the VFE into the future. Following the example of the EFE, in both of these derivations we consider each time step individually and ignore the system dynamics in the preference distribution. Although the FEF ($\hat{G}_\pi$) is very similar to the EFE (G_π) they are not equal, as the following comparison highlights:

$$\begin{aligned}\hat{G}_{\pi,\tau}(\phi) &= -\mathbb{E}_{\hat{q}(o_\tau, x_\tau \mid \pi)}[\ln p(o_\tau, x_\tau) - \ln \hat{q}(x_\tau \mid o_\tau, \pi)] \\ G_{\pi,\tau}(\phi) &= -\mathbb{E}_{\hat{q}(o_\tau, x_\tau \mid \pi)}[\ln p(o_\tau, x_\tau) - \ln \hat{q}(x_\tau \mid \pi)]\end{aligned}$$

Hence, even though the EFE is often treated as a natural extension of the VFE into the future [6, 39], its mathematical origins remain unclear. In fact, the EFE occurs when we add the information gain to the FEF, an objective that we showed is a fairly natural extension of the VFE into the future. This relation between EFE and FEF gives reason to question the mathematical soundness of the EFE as a direct extension of the VFE [43]. Thus, it can be concluded that the EFE might not have been derived in a mathematically principled manner, but rather specifically chosen to contain the information gain.

4.4 Conclusion

In this section, we provided three separate arguments against the claims that the EFE would explain curiosity and resolve the exploration-exploitation dilemma. First, we showed that the EFE, while arguably providing a meaning for the balancing factor between exploration and exploitation, still requires this factor to be carefully calibrated in practice. We went further by providing arguments that the information seeking term of the EFE might not

arise in a mathematically principled way, but rather by the combination of an infeasible approximation and the choice of the definition of the EFE. These results indicate that the current definition of the EFE does not sufficiently justify how exploratory behavior can be derived from the FEP. Finally, we argue that the EFE is not the only extension of the VFE into the future, and that there exists another objective which arises more naturally. Curiously, this objective is exactly the EFE without the expected information gain. We want to emphasize that we are not questioning the FEP from a cognitive science perspective, but rather contest the way the EFE explains intrinsic motivation from a mathematical point of view.

4.5 Impact of Approximation (3.14) on the FEEF

While the origin of the FEEF could be perceived as more intuitive and natural than that of the EFE, it is making use of Approximation (3.14) similar to the EFE. Since we argued in Section 4.2 that this approximation is infeasible, it remains to analyze what impact it has on the FEEF formulation. To gain an intuition on this impact, we start by rewriting the approximated FEEF.

$$\begin{aligned}
\widetilde{\text{FEEF}}(\pi) &= \mathbb{E}_{q_\phi(\tilde{x}\,|\,\pi)}[D_{\text{KL}}[q_\phi(\tilde{o}\,|\,\tilde{x}) \parallel p(\tilde{o})]] \\
&\quad - \mathbb{E}_{q_\phi(\tilde{o}\,|\,\pi)}[D_{\text{KL}}[q_\phi(\tilde{x}\,|\,\tilde{o},\pi) \parallel q_\phi(\tilde{x}\,|\,\pi)]] \\
&= \mathbb{E}_{q_\phi(\tilde{o},\tilde{x}\,|\,\pi)}[\ln q_\phi(\tilde{o}\,|\,\tilde{x}) - \ln q_\phi(\tilde{x}\,|\,\tilde{o},\pi) + \ln q_\phi(\tilde{x}\,|\,\pi) - \ln p(\tilde{o})] \\
&= \mathbb{E}_{q_\phi(\tilde{o}\,|\,\pi)}[\ln q_\phi(\tilde{o}\,|\,\pi) - \ln p(\tilde{o})] \\
&= D_{\text{KL}}[q_\phi(\tilde{o}\,|\,\pi) \parallel p(\tilde{o})]
\end{aligned} \tag{4.17}$$

The above equation shows that Approximation (3.14) eliminates the hidden state $\tilde{x}$ from the original objective. This insight begs the question why the FEEF was originally defined as the KL divergence between both $\tilde{x}$ and $\tilde{o}$ if it could simply have been defined as in Eq. (4.17) to avoid Approximation (3.14). Furthermore, Eq. (4.17) gives reason to believe that the approximated FEEF might not be explicitly information seeking, but rather purely extrinsic in its nature.

5 Methodology: Reinforcement Learning via Active Inference

In Chapter 3, we summarized the overall AI framework. However, the framework leaves many questions open, including the choice of the generative model and the variational distributions, as well as the methods used to optimize them. In this chapter, we propose answers to these questions and show how classical Reinforcement Learning problems can be tackled with Active Inference in practice.

This chapter is structured as follows: In Section 5.1, we show how classical RL problems can be reformulated to fit into the AI framework and derive their perception and planning objectives. We explain our choice of model and the learning procedure in Section 5.2. In Section 5.3, we illustrate our choice of planning algorithm used to perform Model Predictive Control. Crucial to the efficiency of our method is that the EFE can be evaluated swiftly during planning. Since it does not have a closed-form solution, we describe how it can be approximated efficiently in Section 5.4. Furthermore, we discuss the use of Lautum Information as an alternative to the EFE's information term in Section 5.5. Finally, we provide a summary of the resulting AI algorithm in Section 5.6.

5.1 Setup

In this work, we assume that the generative model of the environment factorizes into a MDP conditioned on some parameters θ. More specifically, we assume that the generative model factorizes as

$$p(x_{0:T}, a_{1:T}, r_{1:T}, \theta) = p(x_0)\, p(a_{1:T})\, p(\theta) \prod_{\tau=1}^{T} p(r_\tau \mid x_\tau, a_\tau, \theta)\, p(x_\tau \mid x_{\tau-1}, a_\tau, \theta) \tag{5.1}$$

where $x_\tau \in \mathbb{R}^{N_x}$ is the environment state at time τ, $r_\tau \in \mathbb{R}$ is the reward the agent receives at state x_τ and $a_\tau \in \mathbb{R}^{N_a}$ is the action the agent took leading into step τ.

Formally, a couple of issues have to be addressed to ensure that the model definition in Eq. (5.1) fits into the framework of AI, which assumes a POMDP as per Eq. (3.7). First, we assume that the policy π is modelled as a static sequence of actions $a_{1:T}$. Intuitively, this choice of policy means that during planning the agent does not consider how information it is gathering in the future will change the course of actions it will take. While there is some recent work in the AI domain on taking future information into account during planning [49], it makes planning substantially harder. Thus, it has only been shown to work on very small discrete problems.

Second, there is no explict notion of static parameters in AI. This issue can be easily fixed by assuming that θ is part of the hidden state with a transition function that keeps it constant. Formally, we set $\theta_0 \coloneqq \theta$ and define the state transition function such that

$$p(\theta_\tau \,|\, \theta_{\tau-1}) = \delta(\theta_\tau - \theta_{\tau-1}) \quad \forall \tau \in \{1, \dots, T\}$$

where δ is the Dirac delta function.

Finally, the AI framework does not allow to maximize rewards directly, but rather expects a target observation distribution. While it is possible to provide the agent with target observations instead of rewards for each environment, it would break compatibility to other RL algorithms that are typically designed to maximize a reward signal. Instead we chose to make the rewards part of the observation and set the agent's desire in such a way that it prefers observing high rewards. Consequently, our observation model is a combination of the reward model and a Dirac distribution to make the state fully observable:

$$p(o^r, o^x \,|\, x, a, \theta) = p(r = o^r \,|\, x, a, \theta)\, \delta(o^x - x)$$

where we denote the reward component of the observation vector by o^r and the state component by o^x. Additionally, we dropped the τ subscript as the observation preference model is agnostic of the time step.

This observation model now allows us to define the target observation distribution such that the agent is driven towards high rewards:

$$p(o^r, o^x) = \mathcal{L}\left(o^r \mid \mu_r^*, \sigma_r^*\right) \prod_{i=1}^{N_x} \mathcal{U}\left(o_i^x \mid l_i^x, u_i^x\right)$$

where o_i^x is the i-th component of the state observation and $\mathcal{U}$ and $\mathcal{L}$ are the uniform and Laplace distributions defined as

$$\mathcal{U}(x \mid l, u) = \begin{cases} \frac{1}{u-l} & x \in [l, u] \\ 0 & \text{else} \end{cases}$$
$$\mathcal{L}(x \mid \mu, \sigma) = \frac{1}{2\sigma} e^{-\frac{|x-\mu|}{\sigma}}$$

By setting $l_i^x = -a$ and $u_i^x = a$, where a is a very large but finite real number, we ensure that the model has no a-priori preferences for states. μ_r^* can be seen as the target reward our agent is trying to achieve. If this value is set lower than the maximum achievable reward, the agent will try to avoid rewards that lie significantly above μ_r^*, which is typically not wanted in RL tasks. However, we found that simply setting μ_r^* to the maximum achievable reward does not work well in practice, as learned reward models might overestimate rewards and prevent the agent from visiting high reward regions. As there is no disadvantage in setting μ_r^* really high, we simply assume that it is a large real number. σ_r^*, on the other hand, has to be carefully calibrated as it directly controls the influence of the extrinsic term on the overall objective.

5.1.1 Modelling the transition and reward distributions

With the definition of the general structure of the generative model out of the way, it remains to specify the exact form of the transition model $p(x_\tau \mid x_{\tau-1}, a_\tau, \theta)$, the reward model $p(r_\tau \mid x_\tau, a_\tau, \theta)$, as well as the variational posterior $q_\phi(\theta)$. To ensure that the transition model and the reward model are powerful enough for complex RL tasks, we model both as Gaussian distributions conditioned on the output of a neural network:

$$p(x_\tau \mid x_{\tau-1}, a_\tau, \theta) \coloneqq \mathcal{N}\left(x_\tau \mid \mu_\theta^x(x_{\tau-1}, a_\tau), \sigma^x I\right) \tag{5.2}$$
$$p(r_\tau \mid x_\tau, a_\tau, \theta) \coloneqq \mathcal{N}\left(r_\tau \mid \mu_\theta^r(x_\tau, a_\tau), \sigma^r I\right) \tag{5.3}$$

where $\mu_\theta^x \colon \mathbb{R}^{N_x} \times \mathbb{R}^{N_a} \to \mathbb{R}^{N_x}$ and $\mu_\theta^r \colon \mathbb{R}^{N_x} \times \mathbb{R}^{N_a} \to \mathbb{R}$ are neural networks computing the means of the distributions and $\sigma^x \in \mathbb{R}$ and $\sigma^r \in \mathbb{R}$ are constant standard deviations. While using neural networks to compute the standard deviations as well makes the model more powerful, we found it to require too much data to converge during training in practice.

5.1.2 Perception objective

Having completed the definition of the generative model, we can now write down the VFE, which will be minimized during perception. Let therefore $o^x_{0:t} \in \mathbb{R}^{N_x}$ and $o^r_{1:t} \in \mathbb{R}$ be the states and rewards the agent observed, and $\hat{a}_{1:t} \in \mathbb{R}^{N_a}$ the actions the agent took in the past. We define the variational distribution as

$$q_\phi(x_{0:t}, \theta, \pi) = q_\phi(\pi)\, q_\phi(\theta) \prod_{\tau=1}^{t} q_\phi(x_{0:t})$$

where we refrain from conditioning the state on the policy and instead fix the policy to the actual actions taken by setting

$$p(\pi) \coloneqq p(a_{1:t} \mid \hat{a}_{1:t}) \coloneqq \prod_{\tau=1}^{t} \delta(a_\tau - \hat{a}_\tau)$$

The VFE is now given as

$$\begin{aligned}
\mathcal{F}\left(o^x_{0:t}, o^r_{1:t}, \hat{a}_{1:t}, \phi\right) &= \mathbb{E}_{q_\phi(x_{0:t}, a_{1:t}, \theta)}[\ln q_\phi(x_{0:t}, \theta) - \ln p(o^x_{0:t}, o^r_{1:t}, x_{0:t}, \theta \mid a_{1:t})] \\
&\quad + D_{\mathrm{KL}}[q_\phi(a_{1:t}) \parallel p(a_{1:t} \mid \hat{a}_{1:t})] \\
&= \mathbb{E}_{q_\phi(x_{0:t}, a_{1:t}, \theta)}[\ln q_\phi(\theta) - \ln p(o^r_{1:t}, x_{0:t}, \theta \mid a_{1:t})] \\
&\quad + \mathbb{E}_{q_\phi(x_{0:t})}[\ln q_\phi(x_{0:t}) - p(o^x_{0:t} \mid x_{0:t})] \\
&\quad + D_{\mathrm{KL}}[q_\phi(a_{1:t}) \parallel p(a_{1:t} \mid \hat{a}_{1:t})] \\
&= \mathbb{E}_{q_\phi(x_{0:t}, a_{1:t}, \theta)}[\ln q_\phi(\theta) - \ln p(o^r_{1:t}, x_{0:t}, \theta \mid a_{1:t})] \\
&\quad + D_{\mathrm{KL}}[q_\phi(x_{0:t}) \parallel p(x_{0:t} \mid o^x_{0:t})] \\
&\quad + D_{\mathrm{KL}}[q_\phi(a_{1:t}) \parallel p(a_{1:t} \mid \hat{a}_{1:t})]
\end{aligned} \tag{5.4}$$

where we made use of

$$\begin{aligned}
p(o^x_{0:t} \mid x_{0:t}) &= p(x_{0:t} \mid o^x_{0:t})\, \frac{p(o^x_{0:t})}{p(x_{0:t})} = \delta(o^x_{0:t} - x_{0:t})\, \frac{p(o^x_{0:t})}{p(x_{0:t})} \\
&= \delta(o^x_{0:t} - x_{0:t}) = p(x_{0:t} \mid o^x_{0:t})
\end{aligned}$$

in the second step.

From Eq. (5.4) we can trivially derive that the optimal variational distributions for the policy and the state are given by Dirac delta distributions around the observed values:

$$\begin{aligned}
q^*_\phi(x_{0:t}) &= p(x_{0:t} \mid o^x_{0:t}) = \delta(x_{0:t} - o^x_{0:t}) \\
q^*_\phi(a_{1:t}) &= p(a_{1:t} \mid \hat{a}_{1:t}) = \delta(a_{1:t} - \hat{a}_{1:t})
\end{aligned}$$

Inserting the optimal distributions back into Eq. (5.4) gives

$$\begin{aligned}
\mathcal{F}\left(o^x_{0:t}, o^r_{1:t}, \hat{a}_{1:t}, \phi\right) &= \mathbb{E}_{q_\phi(\theta) q^*_\phi(x_{0:t}) q^*_\phi(a_{1:t})}[\ln q_\phi(\theta) - \ln p(o^r_{1:t}, x_{0:t}, \theta \mid a_{1:t})] \\
&= \mathbb{E}_{q_\phi(\theta) q^*_\phi(x_{0:t}) q^*_\phi(a_{1:t})}[\ln q_\phi(\theta) - \ln p(o^r_{1:t}, x_{0:t}, \theta \mid a_{1:t})] \\
&= \mathbb{E}_{q_\phi(\theta) q^*_\phi(x_{0:t}) q^*_\phi(a_{1:t})}[\ln q_\phi(\theta) - \ln p(\theta) \\
&\quad - \ln p(x_{0:t} \mid a_{1:t}, \theta) - \ln p(o^r_{1:t} \mid x_{0:t})] \\
&= -\mathbb{E}_{q_\phi(\theta) q^*_\phi(x_{0:t}) q^*_\phi(a_{1:t})}\left[\sum_{\tau=1}^{t} \ln p(x_\tau \mid x_{\tau-1}, a_\tau, \theta) + \ln p(o^r_\tau \mid x_\tau, \theta)\right] \\
&\quad + D_{\mathrm{KL}}[q_\phi(\theta) \,\|\, p(\theta)] \\
&= -\mathbb{E}_{q_\phi(\theta)}\left[\sum_{\tau=1}^{t} \ln p\big(o^x_\tau \,\big|\, o^x_{\tau-1}, \hat{a}_\tau, \theta\big) + \ln p(o^r_\tau \mid o^x_\tau, \theta)\right] \\
&\quad + D_{\mathrm{KL}}[q_\phi(\theta) \,\|\, p(\theta)]
\end{aligned}$$

Plugging in Definitions (5.2) and (5.3) and simplifying the notation by writing $x_{0:t} \coloneqq o^x_{0:t}$, $r_{1:t} \coloneqq o^r_{1:t}$, and $a_{1:t} \coloneqq \hat{a}_{1:t}$ we arrive at the learning objective for the perception step:

$$\min_{\phi} \quad \mathcal{F}\left(x_{0:t}, r_{1:t}, a_{1:t}, \phi\right)$$

where

$$\begin{aligned}
\mathcal{F}\left(x_{0:t}, r_{1:t}, a_{1:t}, \phi\right) \coloneqq \frac{1}{2}\mathbb{E}_{q_\phi(\theta)}\left[\sum_{\tau=1}^{t} \left(x_\tau - \mu^x_\theta\left(x_{\tau-1}, a_\tau\right)\right)^2 + \left(r_\tau - \mu^r_\theta\left(x_\tau, a_\tau\right)\right)^2\right] \\
+ D_{\mathrm{KL}}[q_\phi(\theta) \,\|\, p(\theta)] + \ln \sigma^x + \ln \sigma^r + \ln 2\pi
\end{aligned}$$

Dropping constant terms, we obtain a regular expected Mean Squared Error (MSE) loss with a KL regularization on the parameters. We discuss the optimization of this objective in Section 5.2.

5.1.3 Planning objective

During planning, the parameters ϕ of the policy distribution $q^*_\phi(a_{t+1:T})$ are optimized to minimize the EFE[1] $\tilde{G}_\pi\left(\phi\right)$. A crucial component of the EFE is the predictive distribution

[1] By "EFE" we are referring to the approximated EFE that is commonly used in the AI literature.

$q_\phi(x_{t+1:T}, \theta \mid a_{t+1:T})$ that allows us to evaluate the expectation of both the intrinsic and extrinsic terms. While the formal appearance of the predictive distribution suggests that it is equal to the variational distribution used in the VFE, it is important to note that it might not be possible to evaluate the latter for unseen states into the future. In our case, the variational distribution over states $q_\phi(x_{0:t})$ is a Dirac delta distribution around the observed states. Hence, it holds no information about how deviation in one state affects the following states. Thus, it cannot be used to generate predictions of future states.

One solution would be to train a separate model, that maps the current state x_t and a sequence of actions $a_{t+1:T}$ to a distribution over expected states that factorizes in accordance with the mean-field assumption:

$$\tilde{q}_\phi(x_{t+1:t} \mid a_{t+1:T}; x_t) = \prod_{\tau=t+1}^{T} \tilde{q}_\phi(x_\tau \mid a_{t+1:T}; x_t)$$

where T is the planning horizon.

However, learning such a distribution would cause significant training overhead as an entire new model has to be trained in addition to the transition and reward models. Additionally, especially in contact-rich scenarios as those that are considered in this work, it is likely that the mean-field assumption is violated substantially in many cases.

Hence, we deviate from prior work [5, 23, 39] and refrain from making a mean-field assumption over states. Dropping the mean-field assumption makes the evaluation of the EFE more challenging but allows us to use the transition model directly to generate state predictions:

$$q_\phi(x_{t+1:t} \mid a_{t+1:T}, \theta) \coloneqq p(x_{t+1:T} \mid x_t, a_{t+1:T}, \theta) = \prod_{\tau=t+1}^{T} p(x_\tau \mid x_{\tau-1}, a_\tau, \theta)$$

where x_t is defined as the current observed state.

The EFE for a given action sequence is now given as

$$\tilde{G}(\phi, a) = \underbrace{-\mathbb{E}_{q_\phi(\theta, x \mid a)}[D_{\mathrm{KL}}[q_\phi(o^x, o^r \mid x, a, \theta) \parallel q_\phi(o^x, o^r \mid a)]]}_{\text{intrinsic term}} \underbrace{-\mathbb{E}_{q_\phi(o^x, o^r \mid a)}[\ln p(o^x, o^r)]}_{\text{extrinsic term}}$$

where we dropped subscripts for notational convenience.

Since the preference distribution over states $p(o^x)$ is uniform and o^x is guaranteed to lie within its support, the extrinsic term can be rewritten as

$$\begin{aligned}
-\mathbb{E}_{q_\phi(o^x, o^r \mid a)}[\ln p(o^x, o^r)] &= -\mathbb{E}_{p(o^x, o^r \mid x, a, \theta) q_\phi(x \mid a)}[\ln p(o^x) + \ln p(o^r)] \\
&= -\mathbb{E}_{p(o^x \mid x) q_\phi(x \mid a)}[\ln p(o^x)] - \mathbb{E}_{p(o^r \mid x, a, \theta) q_\phi(x \mid a)}[\ln p(o^r)] \\
&\propto -\mathbb{E}_{p(o^r \mid x, a, \theta) q_\phi(x \mid a)}[\ln p(o^r)]
\end{aligned}$$

Furthermore, we can rewrite the intrinsic term as

$$\begin{aligned}
&- \mathbb{E}_{q_\phi(\theta, x \mid a)}[D_{\mathrm{KL}}[q_\phi(o^x, o^r \mid x, a, \theta) \parallel q_\phi(o^x, o^r \mid a)]] \\
&= -\mathbb{E}_{q_\phi(\theta, x \mid a)}[D_{\mathrm{KL}}[p(o^x, o^r \mid x, a, \theta) \parallel q_\phi(o^x, o^r \mid a)]] \\
&= -\mathbb{E}_{q_\phi(\theta, x \mid a) p(o^r \mid x, a, \theta)}\left[\mathbb{E}_{p(o^x \mid x)}[\ln p(o^x, o^r \mid x, a, \theta) - \ln q_\phi(o^x, o^r \mid a)]\right] \\
&= -\mathbb{E}_{q_\phi(\theta, x \mid a) p(o^r \mid x, a, \theta)}[\ln p(o^r \mid x, a, \theta) - \ln q_\phi(x, o^r \mid a)] \\
&= -\mathbb{E}_{q_\phi(\theta) p(x \mid a, \theta) p(o^r \mid x, a, \theta)}[\ln p(x, o^r \mid a, \theta) - \ln q_\phi(x, o^r \mid a) - \ln p(x \mid a, \theta)] \\
&= \underbrace{-\mathbb{E}_{q_\phi(\theta)}[D_{\mathrm{KL}}[p(x, o^r \mid a, \theta) \parallel q_\phi(x, o^r \mid a)]]}_{\text{information gain}} + \underbrace{\mathbb{E}_{q_\phi(\theta)}[H\left[p(x \mid a, \theta)\right]]}_{\text{state entropy}}
\end{aligned}$$

where we used that the predictive distribution q_ϕ follows the same observation model as the generative model p in the first step.

Combining the intrinsic and extrinsic terms again, we arrive at the following joint term for the action-conditioned EFE:

$$\begin{aligned}
\tilde{G}\left(\phi, a\right) = &- \mathbb{E}_{q_\phi(\theta)}[D_{\mathrm{KL}}[p(x, o^r \mid a, \theta) \parallel q_\phi(x, o^r \mid a)]] \\
&+ \mathbb{E}_{q_\phi(\theta)}[H\left[p(x \mid a, \theta)\right]] - \mathbb{E}_{p(o^r \mid x, a, \theta) q_\phi(x \mid a) q_\phi(\theta)}[\ln p(o^r)]
\end{aligned} \tag{5.5}$$

Plugging Definition (5.2) in, we see that the entropy term is constant w.r.t. ϕ:

$$
\begin{aligned}
&\mathbb{E}_{q_\phi(\theta)}[H\left[p(x_{t+1:T} \mid a_{t+1:T}, \theta)\right]] \\
&= -\mathbb{E}_{q_\phi(\theta) p(x_{t+1:T} \mid a_{t+1:T}, \theta)}\left[\sum_{\tau=t+1}^{T} \ln p(x_\tau \mid x_{\tau-1}, a_\tau, \theta)\right] \\
&= -\mathbb{E}_{q_\phi(\theta)}\left[\sum_{\tau=t+1}^{T} \mathbb{E}_{p(x_\tau, x_{\tau-1} \mid a_{t+1:T}, \theta)}[\ln p(x_\tau \mid x_{\tau-1}, a_\tau, \theta)]\right] \\
&= -\mathbb{E}_{q_\phi(\theta)}\left[\sum_{\tau=t+1}^{T} \mathbb{E}_{p(x_{\tau-1} \mid a_{t+1:T}, \theta)}[H\left[p(x_\tau \mid x_{\tau-1}, a_\tau, \theta)\right]]\right] \\
&= -\mathbb{E}_{q_\phi(\theta)}\left[\sum_{\tau=t+1}^{T} \mathbb{E}_{p(x_{\tau-1} \mid a_{t+1:T}, \theta)}\left[\frac{N_x}{2} \ln\left(2\pi e \left(\sigma^x\right)^2\right)\right]\right] \\
&= -\frac{N_x}{2} \ln\left(2\pi e \left(\sigma^x\right)^2\right)
\end{aligned}
\tag{5.6}
$$

Furthermore, using Definition (5.3), the extrinsic term can be rewritten as

$$
\begin{aligned}
&- \mathbb{E}_{q_\phi(x_{t+1:T} \mid a_{t+1:T}) p\left(o^r_{t+1:T} \,\middle|\, x_{t+1:T}, a_{t+1:T}, \theta\right) q_\phi(\theta)}\left[\ln p\left(o^r_{t+1:T}\right)\right] \\
&= -\mathbb{E}_{q_\phi(x_{t+1:T} \mid a_{t+1:T}) q_\phi(\theta)}\left[\sum_{\tau=t+1}^{T} \mathbb{E}_{p(o^r_\tau \mid x_\tau, a_\tau, \theta)}[\ln p(o^r_\tau)]\right] \\
&= -\mathbb{E}_{q_\phi(x_{t+1:T} \mid a_{t+1:T}) q_\phi(\theta)}\left[\sum_{\tau=t+1}^{T} \mathbb{E}_{p(o^r_\tau \mid x_\tau, a_\tau, \theta)}\left[-\frac{|o^r_\tau - \mu^*_r|}{\sigma^*_r} + \ln 2\sigma^*_r\right]\right] \\
&\approx -\mathbb{E}_{q_\phi(x_{t+1:T} \mid a_{t+1:T}) q_\phi(\theta)}\left[\sum_{\tau=t+1}^{T} \mathbb{E}_{p(o^r_\tau \mid x_\tau, a_\tau, \theta)}\left[\frac{o^r_\tau - \mu^*_r}{\sigma^*_r} - \ln 2\sigma^*_r\right]\right] \\
&= -\frac{1}{\sigma^*_r} \mathbb{E}_{q_\phi(x_{t+1:T} \mid a_{t+1:T}) q_\phi(\theta)}\left[\sum_{\tau=t+1}^{T} \mu^r_\theta\left(x_\tau, a_\tau\right)\right] + \frac{\mu^*_r}{\sigma^*_r} + \ln 2\sigma^*_r
\end{aligned}
\tag{5.7}
$$

where we used in step three that $o^r_\tau < \mu^*_r$ holds for all o^r_τ with non-neglectable probability density, since μ^*_r was defined to be very large.

Plugging Eqs. (5.6) and (5.7) back into Eq. (5.5) and dropping constant terms, we obtain

the optimization problem that must be solved during planning:

$$\min_{\phi} \quad \tilde{G}(\phi) \coloneqq \mathbb{E}_{q_\phi(a_{t+1:T})}\left[\tilde{G}(\phi, a_{t+1:T})\right] + D_{\mathrm{KL}}[q_\phi(a_{t+1:T}) \parallel p(a_{t+1:T})] \tag{5.8}$$

where

$$\tilde{G}(\phi, a_{t+1:T}) \propto \underbrace{-\mathbb{E}_{q_\phi(\theta)}\left[D_{\mathrm{KL}}[p\left(x_{t+1:T}, o^r_{t+1:T} \,\middle|\, a_{t+1:T}, \theta\right) \parallel q_\phi\left(x_{t+1:T}, o^r_{t+1:T} \,\middle|\, a_{t+1:T}\right)]\right]}_{\text{information gain}}$$
$$\underbrace{-\frac{1}{\sigma_r^*}\mathbb{E}_{q_\phi(x_{t+1:T} \mid a_{t+1:T}, \theta) q_\phi(\theta)}\left[\sum_{\tau=t+1}^{T} \mu_\theta^r(x_\tau, a_\tau)\right]}_{\text{expected mean reward}}$$

Hence, minimization of the EFE is equivalent to maximization of both the expected mean reward and the expected information gain. Intuitively, this objective compels our agent to not only try to solve some task, but also to explore the environment by making observations that are informative w.r.t. the model parameters. The balance between these two objectives is controlled by the hyperparameter σ_r^*: Increasing it shifts the agent's focus towards exploration, while decreasing it causes the agent to collect task reward more greedily.

So far, we have not discussed the exact choice of the parametric form of the variational distributions $q_\phi(a_{t+1:T})$ and $q_\phi(\theta)$. Since the choice of $q_\phi(a_{t+1:T})$ is tightly coupled with the choice of the planning algorithm, we will discuss it in Section 5.3, where we describe the planning algorithm we use in this work. Furthermore, $q_\phi(\theta)$ should be chosen in a way that facilitates learning and an efficient approximation of the planning objective given in Eq. (5.8). We will elaborate on our choice of $q_\phi(\theta)$ in Section 5.2.

5.2 Learning the model

As derived in Section 5.1.2, for observed states $x_{0:t}$, actions $a_{1:t}$, and rewards $r_{1:t}$, the learning objective is given as

$$\min_{\phi} \quad \mathbb{E}_{q_\phi(\theta)}[f(x_{0:t}, a_{1:t}, r_{1:t}, \theta)]$$

where

$$f\left(x_{0:t}, a_{1:t}, r_{1:t}\right) \coloneqq \frac{1}{2} \sum_{\tau=1}^{t} \left(x_\tau - \mu_\theta^x\left(x_{\tau-1}, a_\tau\right)\right)^2 + \left(r_\tau - \mu_\theta^r\left(x_\tau, a_\tau\right)\right)^2$$

and we set $p(\theta) \coloneqq q_\phi(\theta)$ and thereby expressed that we have no prior preferences over parameters.

There are multiple options to represent the variational posterior $q_\phi(\theta)$. Common choices for representing a distribution over model parameters are particle-based representations [23, 50], Gaussian distributions with diagonal covariance matrix [51] or a combination of both [52]. As we further outline in Section 5.4, we approximate the EFE with a Nested Monte Carlo (NMC) estimator during planning. Since particle based representations are sufficient for Nested Monte Carlo (NMC) estimators, we choose the first option and represent $q_\phi(\theta)$ by a set of P particles $\theta^1, \ldots, \theta^P$. This choice of representation is also known as an ensemble model.

While one could train each particle individually with Stochastic Gradient Descent (SGD), it would neglect the idea that they jointly represent a full distribution. For example, if the training objective had only a single mode, all particles would collapse at the center of that mode and, thus, no longer represent a meaningful posterior. To alleviate this issue, we utilize Stein Variational Gradient Descent (SVGD) [53], which uses a Radial Basis Function (RBF)-kernel term to ensure that the particles stay spread out and represent a meaningful distribution. It is worth noting that due to our small particle size of $P = 5$ and the complexity of the problem we are learning, Stein Variational Gradient Descent (SVGD) seems to have no notable influence on the performance of our method. Instead, it was added in to give more formal justification to our learning method.

5.2.1 Multi-step prediction loss

As we further outline in Section 5.4, during planning model roll-outs have to be performed to predict the results of different choices of action sequences $a_{t+1:T}$. Typically, we set the planning horizon $T = t + 20$, which means that the transition model has to be stacked 20 times to produce a full state sequence $x_{t+1:T}$:

$$\left.\begin{aligned} x_\tau &\sim \mathcal{N}(\mu_\theta^x\left(x_{\tau-1}, a_\tau\right)) \\ r_\tau &\sim \mathcal{N}(\mu_\theta^r\left(x_\tau, a_\tau\right)) \end{aligned}\right\} \quad \forall \tau \in \{t+1, \ldots, T\}$$

One issue we faced in our experiments was that these model roll-outs would start to get extremely inaccurate already after a few steps. The reason for this behavior is that the models are trained on real states, but executed on predicted states during the roll-outs. While this might not seem to be a huge source of error, it is important to keep in mind that in most environments, there are areas of the state space that the agent cannot visit. As an example, in the classical Mountain Car environment (see Section 6.1 for reference), the car has no way of achieving maximum velocity on top of the hill. If due to model error, one of the predicted states falls into such an area, then the model is evaluated on a state it has never seen and also will never see in the dataset. Hence, the following prediction will likely also be inaccurate, making it unlikely that the model will recover onto a reasonable trajectory again. Since these erroneous trajectory predictions can result in large objective values, already a few of them are sufficient to divert the planner away from a good plan.

Hence, we follow prior work [28] and tackle this issue by augmenting the training data of the transition model with predicted states. Formally, we define a set of new generative models, called m-step predictive models, which are defined as

$$p_m(x_{0:t}, r_{1:t} \mid a_{1:t}, \theta) = \left(\prod_{\tau=m}^{t} p(x_\tau \mid x_{\tau-m}, a_{\tau-m+1:\tau}, \theta)\right)\left(\prod_{\tau=1}^{m-1} \mathcal{U}(x_\tau, l_x, u_x)\right) \left(\prod_{\tau=1}^{t} p(r_\tau \mid x_\tau, a_\tau, \theta)\right) \tag{5.9}$$

where the m-step transition model is given as

$$p(x_\tau \mid x_{\tau-m}, a_{\tau-m+1:\tau}, \theta) = \int \prod_{\tau'=\tau-m+1}^{\tau} p(x_{\tau'} \mid x_{\tau'-1}, a_{\tau'}, \theta)\, dx_{\tau-m+1:\tau-1}$$

and $\mathcal{U}$ is again the uniform distribution with very loose limits as defined in Section 5.1.

Under this distribution, the first $m-1$ states are uniformly distributed and from the m-th state on, the $\tau-m$-th state is used to predict the τ-th state. Intuitively, when predicting the next state x_τ, the agent does not use information from the previous $m-1$ states, but rather entirely relies on the action trajectory $a_{\tau-m:\tau}$ and the state $x_{\tau-m}$ it observed m steps ago. The advantage of such a generative model is that when we use it to train the transition model $p(x_{\tau'} \mid x_{\tau'-1}, a_{\tau'}, \theta)$, the agent learns to make accurate predictions m steps into the future, while relying solely on its internal model to generate the intermediate states.

Replacing the generative model defined in Eq. (5.1) with p_m yields the m-step VFE:

$$\mathcal{F}_m(x_{0:t}, r_{1:t}, a_{1:t}, \phi) \propto \mathbb{E}_{q_\phi(\theta)}[f_m(x_{0:t}, r_{1:t}, a_{1:t}, \theta)]$$

where

$$f_m\left(x_{0:t}, r_{1:t}, a_{1:t}, \theta\right) \coloneqq \frac{1}{2} \sum_{\tau=m}^{t} \mathbb{E}_{p(\hat{x}_{\tau-1} \mid x_{\tau-m}, a_{\tau-m+1:\tau}, \theta)} \Big[\left(x_\tau - \mu_\theta^x\left(\hat{x}_{\tau-1}, a_\tau\right)\right)^2 \Big] + \left(r_\tau - \mu_\theta^r\left(x_\tau, a_\tau\right)\right)^2$$

To obtain a stochastic gradient of this objective, we apply MC and obtain

$$f_m\left(x_{0:t}, r_{1:t}, a_{1:t}, \theta\right) \approx \frac{1}{2} \sum_{\tau=m}^{t} \left(x_\tau - \mu_\theta^x\left(\hat{x}_{\tau-1}^m, a_\tau\right)\right)^2 + \left(r_\tau - \mu_\theta^r\left(x_\tau, a_\tau\right)\right)^2$$

where $\hat{x}_{\tau-1}^m$ is a result of sampling m steps from the transition model:

$$\begin{aligned} \hat{x}_\tau^1 &\coloneqq x_\tau \\ \hat{x}_\tau^i &\sim \mathcal{N}\left(\mu_\theta^x\left(\hat{x}_{\tau-1}^{i-1}, a_\tau\right), \sigma_x I\right) \end{aligned} \tag{5.10}$$

Similar to Hafner et al. [28], we stop gradients from flowing through the transition model more than once per sample, as we want model evaluations at later time steps to correct errors made at earlier steps and not vice versa.

To ensure that the model is trained for all step distances it encounters during planning, we define the final model loss function as a weighted mean over all step distances within the planning horizon:

$$f^H\left(x_{0:t}, r_{1:t}, a_{1:t}, \theta\right) = \sum_{m=1}^{H} \beta_m f_m\left(x_{0:t}, r_{1:t}, a_{1:t}, \theta\right)$$

where H is the relative planning horizon and β allows to weight the m-step VFEs. In our implementation, we choose $\beta_1 \coloneqq 0.5$ and $\beta_i \coloneqq \frac{1}{2(H-1)}$ for all $i \neq 1$.

5.2.2 Hardening the reward model via multi-step prediction loss

One issue of the multi-step prediction loss as proposed by Hafner et al. [28] is that it does not allow the reward model to be trained on predicted data. While, unlike the transition model, the accuracy of the reward model does not have a direct impact on the future course of the predicted trajectory, a lack thereof can still cause the planner to misjudge

trajectories. Hence, in this section, we take the idea one step further and extend the multi-step prediction loss to the reward model.

We start by rewriting Eq. (5.9) to condition the reward model on predicted states instead of observed states:

$$p_m(x_{0:t}, r_{1:t} \mid a_{1:t}, \theta) = \left(\prod_{\tau=m}^{t} p(x_\tau \mid x_{\tau-m}, a_{\tau-m+1:\tau}, \theta) \right) \left(\prod_{\tau=1}^{m-1} \mathcal{U}(x_\tau, l_x, u_x) \right) \left(\prod_{\tau=m}^{t} p(r_\tau \mid x_{\tau-m+1}, a_{\tau-m+1:\tau}, \theta) \right) \left(\prod_{\tau=1}^{m-1} \mathcal{U}(r_\tau, l_r, u_r) \right)$$

where l_r and u_r are again set to be very loose bounds, such that we can neglect the impact of the uniform distribution on the loss, and

$$p(r_\tau \mid x_{\tau-m+1}, a_{\tau-m+1:\tau}, \theta) = \int p(r_\tau \mid x_\tau, a_\tau, \theta) \prod_{\tau'=\tau-m+2}^{\tau} p(x_{\tau'} \mid x_{\tau'-1}, a_{\tau'}, \theta)\, dx_{\tau-m+2:\tau-1}$$

Analogously to Section 5.2.1, we arrive at the following m-step VFE:

$$f_m(x_{0:t}, r_{1:t}, a_{1:t}, \theta) \approx \frac{1}{2} \sum_{\tau=m}^{t} \left(x_\tau - \mu_\theta^x\left(\hat{x}_{\tau-1}^m, a_\tau\right)\right)^2 + \left(r_\tau - \mu_\theta^r\left(\hat{x}_\tau^m, a_\tau\right)\right)^2$$

where $\hat{x}_{\tau-1}^m$ is again sampled according to Eq. (5.10).

We evaluate this choice of loss function against the one derived in Section 5.2.1 in Chapter 6.

5.3 Model Predictive Control in the Active Inference setting

In Section 5.1.3, we derived the policy selection objective, but did not elaborate on how it can be optimized. Noting that the objective function $\tilde{G}(\phi)$ can be expressed as an expectation w.r.t. the policy:

$$\begin{aligned} \tilde{G}(\phi) &= \mathbb{E}_{q_\phi(a_{t+1:T})}[g(\phi, a_{t+1:T})] \\ g(\phi, a_{t+1:T}) &= \tilde{G}(\phi, a_{t+1:T}) + \ln q_\phi(a_{t+1:T}) - \ln p(a_{t+1:T}) \end{aligned}$$

we follow prior work [23, 28] and optimize it with the Cross Entropy Method (CEM) [54].

CEM is a method for optimizing problems of the form

$$\min_{\phi} \quad \mathbb{E}_{q_\phi(\nu)}[f(\nu)]$$

where $f\colon \mathbb{R}^{N_\nu} \to \mathbb{R}$ is an objective function depending on some variable $\nu \in \mathbb{R}^{N_\nu}$.

The core idea behind CEM is to start with an initial guess of the parameters ϕ_0 and then iteratively shift $q_\phi(\nu)$ towards low-cost regions. Specifically, in each iteration i, M samples $\nu_1, \ldots, \nu_M$ are drawn from $q_\phi(\nu)$. Then, the cost $c_j \coloneqq f(\nu_j)$ of each sample is evaluated to determine the ρ-quantile c_p for some $\rho \in (0, 1)$. Now, a new distribution $q_{\phi_i}^{c_p}$ is defined, which has zero probability at points where the cost is more than c_p and is otherwise proportional to q_{ϕ_i}:

$$q_{\phi_i}^{c_p}(\nu) \propto \begin{cases} q_{\phi_i}(\nu) & \text{if } f(\nu) \leq c_p \\ 0 & \text{else} \end{cases}$$

Intuitively, we would ideally carry this distribution into the next iteration, as it only assigns non-zero probability to values of ν that have low cost under the objective function. However, usually there exists no ϕ, such that $q_\phi(\nu) = q_{\phi_i}^{c_p}(\nu)\ \forall \nu$. Hence, in the final step we minimize the cross entropy between $q_{\phi_i}^{c_p}$ and $q_{\phi_{i+1}}$ to ensure that $q_{\phi_{i+1}}$ is as close to $q_{\phi_i}^{c_p}$ as possible. As the cross entropy between those two distributions is usually intractable to compute, the samples drawn at the beginning of the episode are reused to obtain a MC approximation.

Similar to prior work [23, 28], we choose the parametric form of $q_\phi(a_{t+1:T})$ to be Gaussian with diagonal variance as it provides a closed-form solution for the cross entropy minimization step:

$$\arg\min_{\mu_\phi, \sigma_\phi} -\frac{1}{p} \sum_{\substack{j=1 \\ c_j \leq c_p}}^{M} \ln \mathcal{N}(\nu_j \mid \mu_\phi, \sigma_\phi) = \left(\mu_\phi^*, \sigma_\phi^*\right)$$

where

$$\left.\begin{aligned} \mu^*_{\phi,k} &= \frac{1}{p}\sum_{\substack{j=1 \\ c_j \leq c_p}}^{M} \nu_{j,k} \\ \sigma^*_{\phi} &= \frac{1}{p}\sum_{\substack{j=1 \\ c_j \leq c_p}}^{M} \left(\nu_{j,k} - \mu^*_{\phi,k}\right)^2 \end{aligned}\right\} \quad \forall k \in N_\nu$$

and $\nu_{j,k}$ denotes the k-th component of sample ν_j.

The full CEM procedure for action selection is depicted in Algorithm 1.

Algorithm 1 Vanilla Cross Entropy Method for planning

Input: Variational parameters ϕ

Initialize $\mu_\tau \leftarrow 0, \sigma_\tau \leftarrow \mathbb{1} \quad \forall \tau \in \{t+1, \ldots, T\}$
for $i = 1, \ldots, n$ **do**
  Sample action sequences $a_{1,\tau}, \ldots, 1_{M,\tau} \sim \mathcal{N}(\mu_\tau, \sigma_\tau I) \quad \forall \tau$
  Evaluate costs $g_m \leftarrow g(\phi, a_m) \quad \forall m \in \{1, \ldots, M\}$
  Collect all samples below the ρ-quantile: $J = \{j \colon a_j \leq g_p\}$
  Compute new means and standard deviations for all τ:
  $\mu_\tau \leftarrow \frac{1}{p}\sum_{j \in J} a_{j,\tau}$
  $\sigma_\tau \leftarrow \sqrt{\frac{1}{p}\sum_{j \in J} (\mu_\tau - a_{j,\tau})^T (\mu_\tau - a_{j,\tau})}$
end for
return μ, σ

5.3.1 Improving the initialization of the CEM planner

One issue we noticed during our experiments is that in environments with sparse rewards, the agent's performance would start to decay after the environment had been explored to a certain degree. Interestingly, we found that with increasing accuracy of the model, the planner's ability to find good policies gradually diminished. The reason for this behavior is rather intricate and best illustrated in an example.

Consider a sparse environment, like the classical Mountain Car environment (see Section 6.1 for reference), where the agent only gets any reward if it reaches a specific,

hard-to-reach target state. At the beginning, the agent does not have a good understanding of its environment. Hence, the intrinsic reward will lead the planner away from its initial state, and it will eventually find the target state. However, if at some point the state space surrounding the initial state has been fully explored, the intrinsic term will become close to zero and, thus, not guide the planner away from the initial state anymore. In that case, the only driving factor will be the extrinsic reward, which is only obtained at the target state. Due to the way CEM works, if none of the initial M action sequences lead the agent to the target state, it will collapse onto some local minimum, as it has no way of learning that there is any higher reward at all. In case of Mountain Car and many other environments, this local minimum is usually not to move at all, as any action will lead to some minor costs. Hence, it is crucial that under the initial action sequences, there are some that lead CEM close enough to the target state, such that the resulting reward causes it to search in the direction of the target state.

Unfortunately, in the vanilla CEM implementation we depict in Algorithm 1, the initial action sequences often do not fulfill this requirement. Even if the target state is comfortably reachable within the planning horizon, we found the probability of randomly drawing an action sequence from the initial normal distribution that reaches that state often to be too low. While simply increasing M and thereby drawing more action sequences per iteration tackles this issue to some degree in simple environments, the curse of dimensionality makes this attempt infeasible for complex, high-dimensional environments.

Instead, in this work, we propose to reuse optimized action sequences from previous episodes to initialize CEM. Specifically, we modify Algorithm 1 such that it stores all optimized action means μ_I and standard deviations σ_I together with the current state x_t in a buffer. Then, at the start of each optimization, we extract the means and standard deviations belonging to the k nearest neighbors of the current state x_t from the buffer. From these means and standard deviations we create samples, which we use as initial samples for CEM. To ensure that our planner is still capable of producing novel trajectories, we still sample from the initial normal distribution and simply use the union of both sets of samples in the first iteration. The resulting method is depicted in Algorithm 2.

Algorithm 2 Cross Entropy Method with policy proposals

Input: Variational parameters ϕ, current state x_t

Initialize $\mu_\tau \leftarrow 0, \sigma_\tau \leftarrow \mathbb{1} \quad \forall \tau \in \{t+1, \dots, T\}$
for $i = 1, \dots, n$ **do**
  Sample action sequences $a_{1,\tau}, \dots, 1_{M,\tau} \sim \mathcal{N}(\mu_\tau, \sigma_\tau I) \quad \forall \tau$
  if i = 0 **then**
    Fetch $\left(\hat{x}^1, \hat{\mu}^1, \hat{\sigma}^1\right), \dots, \left(\hat{x}^k, \hat{\mu}^k, \hat{\sigma}^k\right)$ from the policy proposal buffer, where $\hat{x}^1, \dots, \hat{x}^k$ are the k nearest neighbors of x_t in the buffer
    Sample policy proposals $a_{M+l,\tau} \sim \mathcal{N}\left(\hat{\mu}_\tau^l, \hat{\sigma}_\tau^l\right) \quad \forall \tau \forall l \in \{1, \dots, k\}$
  end if
  Evaluate EFE $g_m \leftarrow g(\phi, a_m) \quad \forall m \in \{1, \dots, M\}$
  Collect all samples below the ρ-quantile: $J = \{j \colon a_j \leq g_p\}$
  Compute new means and standard deviations for all τ:
  $\mu_\tau \leftarrow \frac{1}{p} \sum_{j \in J} a_{j,\tau}$
  $\sigma_\tau \leftarrow \sqrt{\frac{1}{p} \sum_{j \in J} (\mu_\tau - a_{j,\tau})^T (\mu_\tau - a_{j,\tau})}$
end for
Store (x_t, μ, σ) in the policy proposal buffer
return μ, σ

5.4 Approximation of the EFE

In Section 5.3 we have illustrated how CEM can be used to find a policy distribution $q_\phi(a_{t+1:T})$ that minimizes the action-conditioned EFE $g(\phi, a_{t+1:T})$. However, we have not touched upon how the EFE can be evaluated for given variational parameters ϕ and actions $a_{t+1:T}$. Unfortunately, due to the complex nature of the transition and reward model, it cannot be computed analytically. Hence, in this section, we discuss how the EFE can be approximated. Our objective is not to obtain the most accurate approximation possible, but rather one that can be evaluated efficiently while still leading to high overall task performance.

As derived in Section 5.1.3, the action-conditioned EFE decomposes into three parts:

$$g(\phi, a_{t+1:T}) \propto \underbrace{-\mathbb{E}_{q_\phi(\theta)}\left[D_{\mathrm{KL}}[p\big(x_{t+1:T}, o^r_{t+1:T} \,\big|\, a_{t+1:T}, \theta\big) \parallel q_\phi\big(x_{t+1:T}, o^r_{t+1:T} \,\big|\, a_{t+1:T}\big)]\right]}_{\text{information gain}}$$
$$\underbrace{-\frac{1}{\sigma_r^*}\,\mathbb{E}_{q_\phi(x_{t+1:T} \,|\, a_{t+1:T}, \theta) q_\phi(\theta)}\left[\sum_{\tau=t+1}^{T} \mu_\theta^r\left(x_\tau, a_\tau\right)\right]}_{\text{expected mean reward}}$$
$$\underbrace{+\ln q_\phi(a_{t+1:T}) - \ln p(a_{t+1:T})}_{\text{log difference to prior}}$$

The logarithmic difference to the policy prior is trivial to compute as long as the prior $p(a_{t+1:T})$ is chosen to be a distribution that can be evaluated analytically. A reasonable choice for $p(a_{t+1:T})$ might be a zero-mean Gaussian distribution, which would express that the agent prefers actions closer to zero. However, most environments already provide the agent with an incentive to choose small actions by punishing the L2 norm of the action vector in their reward functions. Consequently, we choose $p(a_{t+1:T}) \coloneqq q_\phi(a_{t+1:T})$ to effectively eliminate the impact of the prior on the planner.

Since both the transition model and the reward model are nonlinear functions, neither the expected mean reward, nor the information gain can be computed analytically. While the former can be approximated with sufficient accuracy via MC, the latter is known to be notoriously difficult to compute [55]. In the past, a variety of research has been dedicated towards finding computationally tractable methods of maximizing expected information gain, which is also known as Mutual Information (MI) in the literature. We elaborate on those methods briefly in the following Section 5.4.1.

5.4.1 Variational approximations of Mutual Information

Instead of maximizing Mutual Information (MI) directly, many methods maximize a variational lower bound of it [56–59].

A popular lower bound is the Barber-Agakov bound [56], which is given as

$$\mathrm{MI}(o, \theta) = D_{\mathrm{KL}}[P(o, \theta) \parallel P(o)\, P(\theta)] \geq H\left[P(\theta)\right] + \mathbb{E}_{P(o,\theta)}[Q(\theta \,|\, o)]$$

where P is a generative model over observations o and parameters θ, and Q is an arbitrary variational distribution.

An estimate of MI can then be obtained from this bound by maximizing it with an EM-style algorithm at every evaluation [56] or via learning an amortized approximation [57]. The issue with the first approach is that it is too expensive to be used in our scenario. During one execution of our planner, which we require at every step our agent takes, the EFE is evaluated $n \cdot M$ times, where n are the CEM iterations and M are the number of samples drawn per iteration. In our experiments we use $n = 10$ and $M = 2000$, which gives a total of 20000 evaluations per time step. Consequently, for each step the agent takes, it has to solve 20000 large-scale optimization problems, which is completely infeasible.

The second approach suffers from a different problem. Here, the issue of the large amount of optimization problems is alleviated by training an amortized approximation (e.g. a neural network), that outputs an estimate of the variational distribution $Q(\theta \mid o)$ given the observations o. The issue here is that in our case, θ contains the parameters of two neural networks and is, hence, very large. Due to the high-dimensionality of θ, training such an amortized approximation would likely be challenging and require a huge amount of data. Additionally, running 20000 forward passes through a general approximator with such high output dimensions is also going to be very resource intensive. Hence, this approach is also not feasible in our scenario.

Another popular approach of estimating a lower bound is *Mutual Information Neural Estimation (MINE)* [58]. In this approach, a neural network is utilized to maximize the Donsker-Varadhan [60] or f-divergence [61, 62] bound. The bounds are given as

$$\begin{aligned}
\text{MI}(o, \theta) &\geq \text{DV}(\psi) \coloneqq \mathbb{E}_{P(o)}[T_\psi(o, \theta)] - \ln \mathbb{E}_{P(o)P(\theta)}\left[e^{T_\psi(o,\theta)}\right] \\
\text{MI}(o, \theta) &\geq \text{f-div}(\psi) \coloneqq \mathbb{E}_{P(o)}[T_\psi(o, \theta)] - \mathbb{E}_{P(o)P(\theta)}\left[e^{T_\psi(o,\theta)-1}\right]
\end{aligned}$$

where $T_\psi \colon \mathbb{R}^{N_o} \times \mathbb{R}^{N_\theta} \to \mathbb{R}$ is an arbitrary function parameterized by ψ. The issue with this approach is of a similar nature as the previous issue we discussed: The function T_ψ, which is implemented by a neural network, has to map from the huge parameter space of θ to $\mathbb{R}$. Thus, training it will likely be difficult and require a large amount of data, and its execution will be expensive. Hence, in this work, we estimate the information term with a Nested Monte Carlo (NMC) estimator, which we will elaborate on in the following.

5.4.2 NMC approximation of MI

The NMC estimator of MI can be formulated as

$$\begin{aligned}\mathrm{MI}(o,\theta) &= \mathbb{E}_{P(o,\theta)}\big[\ln P(o\,|\,\theta) - \ln \mathbb{E}_{P(\theta)}[P(o\,|\,\theta)]\big] \\ &\approx \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ln P(o_i\,|\,\theta_{i,0}) - \ln \underbrace{\frac{1}{m}\sum_{k=1}^{m} P(o_i\,|\,\theta_{i,k})}_{\text{inner estimator}}}_{\text{outer estimator}}\end{aligned} \tag{5.11}$$

where each $\theta_{i,k}$ are drawn i.i.d. from $P(\theta)$ and o_i is a sample drawn from the observation model induced by $\theta_{i,0}$:

$$\left.\begin{aligned}\theta_{i,k} &\sim P(\theta) \\ o_i &\sim P(o\,|\,\theta_{i,0})\end{aligned}\right\} \quad \forall i \in \{1,\dots,n\}, k \in \{1,\dots,m\}$$

Due to the inner estimator which we require to approximate $P(o)$, the overall estimator is biased towards overestimating the exact MI:

$$\begin{aligned}&\mathbb{E}_{P(\theta_{1:n,0:m},o_{1:n,1:n_o})}\left[\frac{1}{n}\sum_{i=1}^{n} \ln P(o_i\,|\,\theta_{i,0}) - \ln \frac{1}{m}\sum_{k=1}^{m} P(o_i\,|\,\theta_{i,k})\right] \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{P(\theta,o)}[\ln P(o\,|\,\theta)] - \mathbb{E}_{P(\theta_{i,1:m},o_i)}\left[\ln \frac{1}{m}\sum_{k=1}^{m} P(o_i\,|\,\theta_{i,k})\right] \\ &\geq \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{P(\theta,o)}[\ln P(o\,|\,\theta)] - \mathbb{E}_{P(o)}\left[\ln \frac{1}{m}\sum_{k=1}^{m} \mathbb{E}_{P(\theta)}[P(o\,|\,\theta)]\right] \\ &= \mathbb{E}_{P(\theta,o)}[\ln P(o\,|\,\theta) - \ln P(o)] = \mathrm{MI}(o,\theta)\end{aligned}$$

where we used Jensen's inequality and the independence of o_i and $\theta_{i,1:m}$ in the second step. Note that, assuming $P(\theta)$ is non-degenerate, the inequality is non-strict iff o is independent of θ in P. Hence, except in this corner case, the estimator exhibits a non-zero bias.

Yet, as shown by Rainforth et al. [63], this estimator will converge towards the real MI at a rate of $\mathcal{O}(\frac{1}{n} + \frac{1}{m^2})$. Consequently, if the total number of samples is fixed, $n \propto m^2$ should be chosen for optimal convergence.

Reusing samples

In our scenario, two separate factors are limiting the maximum number of samples we can use in the NMC estimator. First, since we chose a particle-based representation of the parameter distribution, for every additional sample drawn from $P(\theta)$, two full neural networks have to be trained during the perception step. Second, when evaluating the estimator, for every sample $\theta_{i,k}$, the generative model $P(o \mid \theta)$ must be evaluated once if $k \neq 0$ and twice otherwise (once inside Eq. (5.11) and once to generate o_i). To limit both the number of θ samples and the number of evaluations of the generative model as much as possible, we propose to reuse the samples from the outer estimator inside the inner estimator.

Hence, we propose the approximation

$$\mathrm{MI}(o, \theta) \approx \frac{1}{n} \sum_{i=1}^{n} \ln P(o_i \mid \theta_i) - \ln \frac{1}{n} \sum_{\substack{k=1 \\ k \neq i}}^{n} P(o_i \mid \theta_k) \tag{5.12}$$

where

$$\left.\begin{aligned} \theta_i &\sim P(\theta) \\ o_i &\sim P(o \mid \theta_i) \end{aligned}\right\} \quad \forall i \in \{1, \ldots, n\}$$

Note that we exclude the sample θ_i that generated o_i from being used in the inner estimator. The reason for this exclusion is twofold: First, using the generating sample in the inner estimator introduces additional bias, as o_i and θ_i are dependent:

$$\begin{aligned} & - \mathbb{E}_{P(\theta_{1:n}, o_i)} \left[\ln \frac{1}{n} \sum_{k=1}^{n} P(o_i \mid \theta_k) \right] \\ & \geq -\mathbb{E}_{P(o_i)} \left[\ln \frac{1}{n} \left(\mathbb{E}_{P(\theta_i \mid o_i)}[P(o_i \mid \theta_i)] + \sum_{\substack{k=1 \\ k \neq i}}^{n} \mathbb{E}_{P(\theta_k)}[P(o_i \mid \theta_k)] \right) \right] \\ & = -\mathbb{E}_{P(o_i)} \left[\ln \left(\frac{1}{n} \left(\mathbb{E}_{P(\theta_i \mid o_i)}[P(o_i \mid \theta_i)] + (n-1) P(o_i) \right) \right) \right] \\ & \geq -\mathbb{E}_{P(o_i)} \left[\ln \left(\frac{1}{n} \left(P(o_i) + (n-1) P(o_i) \right) \right) \right] \\ & = -\mathbb{E}_{P(o_i)}[\ln P(o_i)] \end{aligned}$$

where we used the inequality $\mathbb{E}_{P(\theta_i \mid o_i)}[P(o_i \mid \theta_i)] \geq P(o_i)$ in the third step. This inequality is again a consequence of Jensen's inequality and non-strict only if o_i is independent of θ_i:

$$\begin{aligned}\mathbb{E}_{P(\theta_i \mid o_i)}[P(o_i \mid \theta_i)] &= \int p(o_i \mid \theta_i)\, p(\theta_i \mid o_i)\, d\theta_i = \int p(o_i \mid \theta_i)^2 \, \frac{p(\theta_i)}{p(o_i)} d\theta_i \\ &= \frac{1}{p(o_i)} \mathbb{E}_{p(\theta_i)}\Big[p(o_i \mid \theta_i)^2\Big] \geq \frac{1}{p(o_i)} \mathbb{E}_{p(\theta_i)}[p(o_i \mid \theta_i)]^2 = p(o_i)\end{aligned}$$

The second reason we exclude the generating sample from the inner estimator is of a practical nature, although it is related to the first reason. When evaluating the inner estimator during our experiments, the generating sample θ_i usually assigns o_i a far higher probability than all other samples θ_k. This should not come as a surprise, as o_i was sampled from $P(o_i \mid \theta_i)$ and is now evaluated against the distribution that it originated from. The issue with this difference in scale of $P(o_i \mid \theta_k)$ is that it makes the log-sum operation of the inner estimator numerically unstable. Hence, due to rounding error, we regularly observed $\sum_{k=1}^{m} P(o_i \mid \theta_k)$ being evaluated to $P(o_i \mid \theta_i)$, which results in an overall MI estimate of $\ln m$, which is not informative.

If the generating sample is excluded from the inner estimator, it is trivial to show that the estimator in Eq. (5.12) has the same bias as the vanilla estimator in Eq. (5.11) if we set $m := n-1$. However, by reusing samples, we violate the i.i.d. estimation which guarantees us a convergence rate of $\mathcal{O}\big(\frac{1}{n} + \frac{1}{m^2}\big)$. While a formal investigation into the convergence rate of this estimator is beyond the scope of this thesis, we show some empirical results that suggest a much higher sample efficiency than the vanilla estimator in Fig. 5.1, especially if the number of samples is low.

5.4.3 Restraining the information term via likelihood clipping

A crucial hyperparameter of our method is the weight of the extrinsic term $\frac{1}{\sigma_r^*}$. If it is set too high, the agent will ignore any intrinsic reward and greedily move towards the closest local optimum. On the other hand, if it is set too low, the agent will not exhibit any exploitative behavior and, hence, not converge towards an optimal strategy.

One factor that makes calibrating this weight particularly challenging is that both the exact MI, as well as our estimator are unbounded above. In practice, using this unbounded NMC estimator regularly results in large spikes of the intrinsic term that overrule any influence of the extrinsic term and effectively prevent exploitative behavior. A common technique in such situations is to clip values at a certain threshold. However, we found

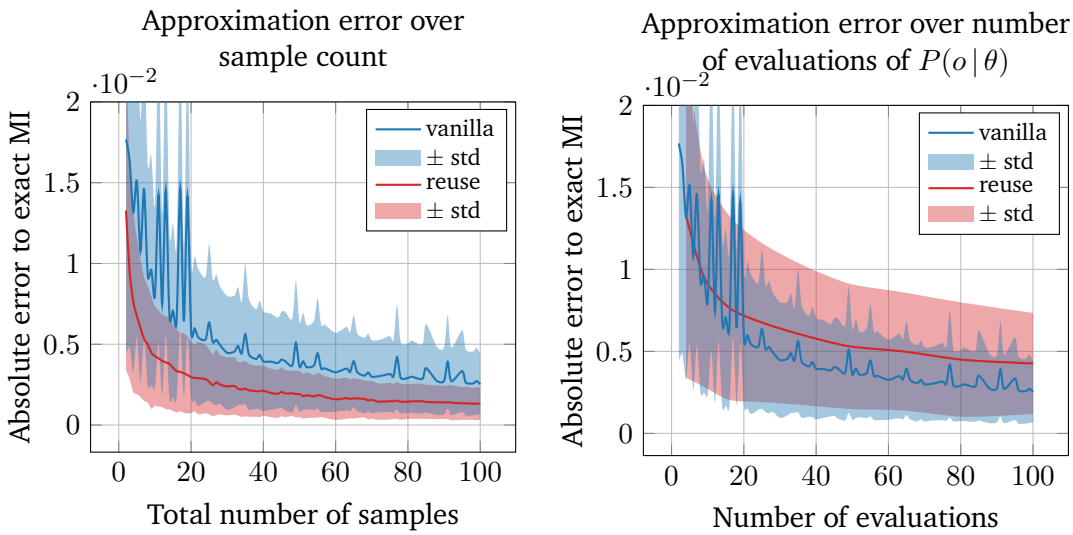


Figure 5.1: Empirical comparison of the approximation errors of the vanilla MI estimator (Eq. (5.11)) and the sample-reusing MI estimator (Eq. (5.12)). Both estimators were tested on 1000 randomly generated discrete distributions $P(o \mid \theta)$. The parameter θ as well as the observation o can take 100 different values under P. "Total number of samples" refers to the total number of samples drawn from $P(\theta)$, which is $T = nm$ in case of the vanilla estimator and $T = n$ in case of the sample-reusing estimator. Since in case of the vanilla estimator we do not know how to choose n and m for a fixed T, we try all pairs of (n, m) such that $T = nm$ and report the best result. On the left, the accuracy, measured in absolute error to the real MI, is plotted over the total number of samples of θ. It is apparent, that reusing samples in the inner estimator yields a significant advantage in sample efficiency, especially when the number of samples is low. On the right, we compare the accuracy over the number of evaluations of the likelihood model $P(o \mid \theta)$. Here, the vanilla estimator is in the advantage as the number of evaluations rise, which is not surprising as it has a larger and more diverse set of θ samples. However, the advantage seems to be insignificant for a low number of evaluations.

that simply clipping the estimated intrinsic term does not yield good exploratory behavior in practice.

To understand why clipping of the intrinsic term does not yield good results, it is useful to write down the NMC estimator for our generative model:

$$
\begin{aligned}
&\mathrm{MI}((x_{t+1:T}, r_{t+1:T}), \theta \mid a_{t+1:T}, x_t) \\
&\approx \frac{1}{n} \sum_{i=1}^{n} \ln p\big(x^i_{t+1:T}, r^i_{t+1:T} \,\big|\, \theta_i\big) - \ln \frac{1}{n} \sum_{\substack{k=1 \\ k \neq i}}^{n} p\big(x^i_{t+1:T}, r^i_{t+1:T} \,\big|\, \theta_k\big) \\
&= \frac{1}{n} \sum_{i=1}^{n} \sum_{\tau=t+1}^{T} \ln p\big(x^i_\tau, r^i_\tau \,\big|\, x^i_{\tau-1}, a_\tau, \theta_i\big) - \ln \frac{1}{n} \sum_{\substack{k=1 \\ k \neq i}}^{n} \prod_{\tau=t+1}^{T} p\big(x^i_\tau, r^i_\tau \,\big|\, x^i_{\tau-1}, a_\tau, \theta_k\big) \\
&= \frac{1}{n} \sum_{i=1}^{n} \sum_{\tau=t+1}^{T} \ln p\big(x^i_\tau, r^i_\tau \,\big|\, x^i_{\tau-1}, a_\tau, \theta_i\big) \\
&\quad - \ln \frac{1}{n} \sum_{\substack{k=1 \\ k \neq i}}^{n} \exp\left(\sum_{\tau=t+1}^{T} \ln p\big(x^i_\tau, r^i_\tau \,\big|\, x^i_{\tau-1}, a_\tau, \theta_k\big) \right)
\end{aligned}
\tag{5.13}
$$

where

$$
\left.
\begin{aligned}
\theta_i &\sim q_\phi(\theta) \\
x^i_{t+1:T} &\sim p\big(x^i_{t+1:T} \,\big|\, a_{t+1:T}, x_t, \theta_i\big) \\
r^i_{t+1:T} &\sim p\big(r^i_{t+1:T} \,\big|\, x^i_{t+1:T}, a_{t+1:T}, \theta_i\big)
\end{aligned}
\right\} \quad \forall i \in \{1, \ldots, n\}
$$

From Eq. (5.13), it becomes apparent that in order to maximize the estimated MI, the first logarithmic term $\ln p\big(x^i_\tau, r^i_\tau \,\big|\, x^i_{\tau-1}, a_\tau, \theta_i\big)$ has to be maximized and the second term has to be minimized. Since (x^i_τ, r^i_τ) is drawn from $p\big(x^i_\tau, r^i_\tau \,\big|\, x^i_{\tau-1}, a_\tau, \theta_i\big)$ and we fixed the variances of the underlying Gaussian distributions, there is not much variance to expect in the first term. In the second term, however, the generated states and rewards are evaluated under distributions they were not sampled from. Practically, each state-reward trajectory pair is generated by some neural network and then evaluated against all other neural networks in the ensemble. Intuitively, our estimator computes the "informativeness" of action trajectories $a_{t+1:T}$ based on an estimate of how much the models disagree on the trajectory the actions would produce.

Since the neural networks are not perfect, they will occasionally make substantially different predictions, leading to very low likelihoods in the second term. As all likelihoods

in the second term are multiplied, a single low likelihood per trajectory will cause the total trajectory $p\big(x_\tau^i, r_\tau^i \,\big|\, x_{\tau-1}^i, a_\tau, \theta_k\big)$ likelihood to be close to zero. Consequently, if the predictions of the generating model and the evaluating models each differ substantially in just a single time step, the overall estimated MI becomes arbitrarily large.

While it is certainly reasonable to explore trajectories in which the models do not agree, it is questionable whether it is worth doing so if the disagreement is limited on a single step of the predicted trajectory only. We argue that it is better to explore trajectories for which the disagreement is large in every step, instead of massive in a single step, as the models will have more unseen data to train on. However, if we just clip the overall estimated MI, the clipping limit might be already be reached by even such trajectories that cause large disagreement in a singular step. Hence, there would be no way of differentiating those trajectories that contain more than one interesting transition.

To tackle this issue, we propose to clip the per-step likelihoods below instead of the full estimated MI above. That is, we define

$$p_c\big(x_\tau^i, r_\tau^i \,\big|\, x_{\tau-1}^i, a_\tau, \theta_i\big) = \max\left(p_c\big(x_\tau^i, r_\tau^i \,\big|\, x_{\tau-1}^i, a_\tau, \theta_i\big), c\right)$$

where $c \in \mathbb{R}$ is the clipping limit and use p_c in the NMC estimator instead of p.

By doing so, we essentially express that there is a limit after which we consider the discrepancy between models to stop gaining "informativeness". In other words, if for some transition the evaluating model predicts a vastly different mean successor state than the generating model, then we do not care how far away it is, as soon as the distance reached a certain threshold. Practically, this change to the NMC estimator causes the planner to prefer action sequences that result in many transitions with large model discrepancies, instead of few transitions with huge discrepancy.

5.5 Lautum Information

Another idea we investigate in this thesis is to replace the KL-divergence inside MI with the reverse KL-divergence. Hence, instead of $D_{\mathrm{KL}}[P(o) \parallel P(o)\,P(\theta)]$, we obtain

$$\mathrm{LI}(o, \theta) \coloneqq D_{\mathrm{KL}}[P(o)\,P(\theta) \parallel P(o, \theta)]$$

where P is again a generative model over observations o and model parameters θ.

The resulting information term is known as Lautum Information (LI) [7] and, similar to MI, measures information between o and θ. However, contrary to MI, Lautum Information (LI) has not seen much attention in the Bayesian Optimal Experimental Design (BOED) literature so far. Hence, in this section, we highlight some of its properties and compare it to MI. We will argue both theoretically and empirically that LI is a feasible alternative to MI. Furthermore, we will present a sample-efficient NMC approximation of LI that outperforms MI-NMC estimator in simple discrete test cases.

5.5.1 Sample efficient NMC estimation of LI

Analogously to MI, the vanilla NMC estimator is given as

$$\mathrm{LI}(o,\theta) = \mathbb{E}_{P(o)P(\theta)}\left[\ln \mathbb{E}_{P(\theta)}[P(o \mid \theta)] - \ln P(o \mid \theta)\right] \tag{5.14}$$

$$\approx \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ln \underbrace{\left(\frac{1}{m}\sum_{k=1}^{m} P(o_i \mid \theta_{i,k})\right)}_{\text{inner estimator}} - \ln P(o_i \mid \theta_{i,0})}_{\text{outer estimator}} \tag{5.15}$$

where

$$\left.\begin{aligned} \theta_{i,k}, \hat{\theta}_i &\sim P(\theta) \\ o_i &\sim P\left(o \,\middle|\, \hat{\theta}_i\right) \end{aligned}\right\} \quad \forall i \in \{1,\dots,n\}, k \in \{1,\dots,m\}$$

Note that in case of LI, the generating sample $\hat{\theta}_i$ of o_i does not appear in the estimator again. The reason for this is that the expectation is taken over the two independent marginal distributions $P(o)$ and $P(\theta)$ rather than the joint $P(o,\theta)$.

Analogously to the vanilla MI estimator, one can show that this estimator is also biased, but underestimates the exact value instead of overestimating it. Since the bias of both estimators stems from the same term only with the sign switched, they actually exhibit the exact same bias for fixed P, only with different signs.

Similar to the sample-reusing estimator we derived for MI, we can also reuse samples in this estimator. However, the fact that the variables o and θ in the outer expectation of Eq. (5.14) are drawn independently allows us to take this idea one step further. Instead of drawing samples $\theta_i \sim P(\theta)$, $o_i \sim P(o)$ and only using them in a single pair (θ_i, o_i), we can form arbitrary combinations of samples (θ_i, o_j). Hence, we receive quadratically many samples at the cost of violating the i.i.d. assumption of the sample points.

The resulting sample-reusing NMC estimator is given as

$$\mathrm{LI}(o,\theta) \approx \frac{1}{n}\sum_{i=1}^{n} \ln\left(\frac{1}{n}\sum_{\substack{k=1\\k\neq i}}^{n} P(o_i \,|\, \theta_k)\right) - \frac{1}{n}\sum_{\substack{j=1\\j\neq i}}^{n} \ln P(o_i \,|\, \theta_j) \tag{5.16}$$

where

$$\left.\begin{aligned}\theta_i &\sim P(\theta)\\ o_i &\sim P(o \,|\, \theta_i)\end{aligned}\right\} \quad \forall i \in \{1,\ldots,n\}\,,$$

Note that we again explicitly avoid that any observation sample is evaluated against its generating θ sample to prevent introducing additional bias. Although it is also worth mentioning that, in practice, we did not experience any numerical issues or other disadvantages when not excluding the generating θ from the inner sums.

To evaluate the effect of reusing samples on the estimator's accuracy, we conduct the same empirical analysis we did for the MI estimator in Fig. 5.2. Although a thorough formal investigation would be required to make a definitive statement, the empirical results suggest that the advantage of reusing samples in the LI estimator is even more significant than it was in the MI estimator. Note that due to the difference in scale, the error values of the LI and MI estimators cannot be compared directly. However, we conduct an empirical comparison of the information gathering efficiency of the MI and LI in Section 5.5.3.

5.5.2 Relation between LI and MI

In the context of this thesis, an interesting question is how the information seeking behaviors induced by MI and LI differ. To answer this question, we consider a scenario in which the agent has no extrinsic goal, but rather purely relies on the information term for choosing policies. Hence, we want to gain an understanding of the characteristics of policies that maximize MI versus LI.

In the following we will take a close look at the definitions of MI and LI and argue that they induce fundamentally different exploration strategies. Let therefore

$$P(o,\theta \,|\, \pi) = P(o \,|\, \theta,\pi)\, P(\theta)$$

be a generative model over observations o and model parameters θ, which is conditioned on some policy π.

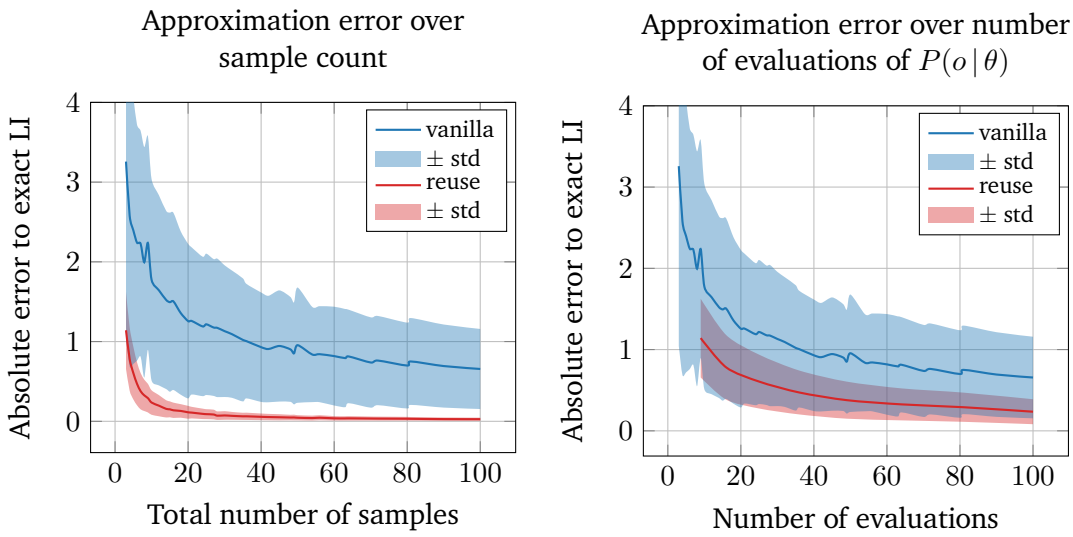


Figure 5.2: Empirical comparison of the approximation errors of the vanilla LI estimator (Eq. (5.15)) and the sample-reusing LI estimator (Eq. (5.16)). The test conditions are exactly the same as described in Fig. 5.1. On the left, the accuracy, measured in absolute error to the real MI, is plotted over the total number of samples of θ. Since we reuse θ samples not only in the inner estimator but also combine them with multiple o samples, the difference in sample efficiency is even more significant than for the MI estimators. On the right, we compare the accuracy over the number of evaluations of the likelihood model $P(o \mid \theta)$. Here, the sample-reusing estimator also holds the advantage, as it can reuse evaluations of $P(o \mid \theta)$ from the outer estimator in the inner estimator.

The definition of MI can be rewritten as

$$\mathrm{MI}(o, \theta \mid \pi) = \mathbb{E}_{P(o \mid \pi)}\big[\mathbb{E}_{P(\theta \mid o, \pi)}[\ln P(\theta \mid o, \pi) - \ln P(\theta)]\big] \tag{5.17}$$

Upon close inspection, it becomes apparent that for a given observation o, the inner difference of logarithms becomes maximal for any θ that has a high probability under the posterior and a low probability under the prior. Since the expectation under the posterior $P(\theta \mid o, \pi)$ is taken over this difference, it is essentially reweighted in favor of θ with high posterior probability. Roughly speaking, the inner difference of logarithms is only considered for those θ that have a substantial probability under the posterior. Hence, an agent maximizing MI is seeking to make observations that for some θ maximize their posterior while minimizing their prior.

Intuitively, this behavior could be understood as follows. For simplicity, consider a discrete set of model parameters $\theta_1, \ldots, \theta_K$. We can think of these parameters as different theories the agent has over the causes of the observations it is making. In its prior $P(\theta)$, the agent assigns each of these theories some probability, and in the posterior $P(\theta \mid o, \pi)$, the theories get reweighted to incorporate the a new observation o. For example, the agent could have theories of the weather being either rainy or sunny. Without checking, it might assign both theories an a-priori probability of 50%. As soon as it goes outside, however, it might observe dark clouds and subsequently increase the probability of its theory that the weather is rainy.

In this setting, theories are proven by making observations that give them a high probability under the posterior and disproven by observations that give them a low probability under the posterior. Thus, the behavior of an agent maximizing MI could be understood as trying to make observations that prove theories which were a-priori considered to be unlikely. Of course this scenario is idealized, as the agent will rarely have the choice to make observations that conclusively prove theories in the real world. However, it gives a good impression of what an MI maximizing agent is striving to achieve.

An agent following LI is acting upon a different exploration strategy, which we see by rewriting the definition of LI to

$$\mathrm{LI}(o, \theta \mid \pi) = \mathbb{E}_{P(o \mid \pi)}\big[\mathbb{E}_{P(\theta)}[\ln P(\theta) - \ln P(\theta \mid o, \pi)]\big]$$

The difference to Eq. (5.17) is that the difference of logarithms is negated and that the expectation is now taken over the prior $P(\theta)$ and not the posterior. Hence, roughly speaking, the inner approximation considers only those θ that have non-neglectable probability under the prior. Consequently, LI can be maximized by making observations

such that for some θ with high a-priori probability, their posterior becomes minimal. If we apply the same thought process we applied to MI, we can conclude that LI drives agents towards disproving theories they deemed a-priori likely.

Thus, the difference of behavior induced by MI and LI can be summarized as follows: Mutual Information drives agents towards *proving* a-priori *unlikely* theories, while Lautum Information drives agents towards *disproving* a-priori *likely* theories.

Exemplary comparison of behaviors induced by MI and LI

To see that these two strategies can lead to substantially different behavior, consider the following discrete example: The agent has three theories about the hidden state of the world θ_1, θ_2, and θ_3, which a-priori it assumes to be equally likely:

$P(\theta)$	θ_1	θ_2	θ_3
	0.33	0.33	0.33

It can interact with the world by choosing one of two policies π_1 and π_2, and make one of two observations o_1 and o_2. The corresponding likelihood models for each policy are given as

$P(o \mid \theta, \pi_1)$	θ_1	θ_2	θ_3
o_1	0.5	0.01	0.5
o_2	0.5	0.99	0.5

$P(o \mid \theta, \pi_2)$	θ_1	θ_2	θ_3
o_1	0.01	0.5	0.01
o_2	0.99	0.5	0.99

To gain an in-depth understanding of the composition of the value of MI, we can analyze the weighted values of the inner term before they are summed up in the expectation. Hence, for

$$\begin{aligned}
\mathrm{MI}(o, \theta \mid \pi) &= \sum_{i=1}^{2} \sum_{j=1}^{3} f_{\mathrm{MI}}\left(o_i, \theta_j, \pi\right) \\
f_{\mathrm{MI}}\left(o_i, \theta_j, \pi\right) &= P(\theta_j)\, P(o_i \mid \theta_i, \pi) \left(\ln P(\theta_j \mid o_i, \pi) - \ln P(\theta_j)\right)
\end{aligned}$$

we obtain

$f_{\text{MI}}(o, \theta, \pi_1)$	θ_1	θ_2	θ_3
o_1	0.066	-0.012	0.066
o_2	-0.047	0.132	-0.047
$\sum$		0.158	

$f_{\text{MI}}(o, \theta, \pi_2)$	θ_1	θ_2	θ_3
o_1	-0.010	0.177	-0.010
o_2	0.060	-0.084	0.060
$\sum$		0.193	

We can see that MI prefers policy π_2. Upon inspection of the values composing MI for π_2, it becomes apparent that the pair (o_1, θ_2) has the strongest positive influence on the overall value. The strong influence of this pair tells us something about the reasoning behind MI's preference of π_2: After choosing π_2, MI is hoping to observe o_1, because it would allow the agent to nearly conclusively prove theory θ_2, since it is very unlikely that θ_1 or θ_3 produced o_1. For π_1 the strongest positive influence comes from (o_2, θ_2) with a similar argument: Observing o_2 makes θ_2 more likely in the posterior. However, in case of π_1, θ_1 and θ_3 also each have a fairly high probability of producing o_2, meaning that θ_2 is not proven after observing o_2.

The same procedure can be applied to LI with

$$\text{LI}(o, \theta \mid \pi) = \sum_{i=1}^{2} \sum_{j=1}^{3} f_{\text{LI}}(o_i, \theta_j, \pi)$$

$$f_{\text{LI}}(o_i, \theta_j, \pi) = P(\theta_j)\, P(o_i \mid \pi) \left(\ln P(\theta_j) - \ln P(\theta_j \mid o_i, \pi)\right)$$

yielding

$f_{\text{LI}}(o, \theta, \pi_1)$	θ_1	θ_2	θ_3
o_1	-0.044	0.395	-0.044
o_2	-0.063	-0.089	-0.063
$\sum$		0.342	

$f_{\text{LI}}(o, \theta, \pi_2)$	θ_1	θ_2	θ_3
o_1	0.165	-0.061	0.165
o_2	-0.050	0.139	-0.050
$\sum$		0.308	

As we can see, unlike MI, LI prefers policy π_1. The largest positive influence on the LI for π_1 is taken by the pair (o_1, θ_2). Hence, the reason why LI prefers π_1 is that it hopes to make observation o_1, which would allow it to almost conclusively disprove θ_2. In case of π_2, the strongest influence comes from (o_2, θ_2), because observing o_2 reduces the probability of θ_2. Interestingly, if we compare the value compositions, we see that exactly those pairs with a high positive influence in MI have a negative influence in LI and vice versa.

This example shows the fundamentally different approaches MI and LI take to define which action is informative and which is not. Which of these approaches results in a more effective exploration is hard analyze from a theoretical point of view, but we will conduct empirical comparisons in Section 5.5.3 and throughout our experiments in Chapter 6.

5.5.3 Empirical comparison of LI and MI

In this section, we conduct an empirical study on the exploration capabilities of the MI and LI NMC approximations. The essential question we aim to answer is how effective the approximations are in identifying informative policies. To answer this question, we will conduct two experiments on randomly generated generative models $P(o, \theta)$. In the first experiment, we test how well the relative "informativeness" of policies estimated by the approximators match these of their exact counterparts. In the second experiment, we investigate how well the approximated information measures are suited to optimize the identification of model parameters.

For both experiments, we assume discrete sets of $n_o = 100$ different observations o, $n_\pi = 100$ policies π and $n_\theta = 50$ model parameter realizations θ. The generative model again factors as

$$P(o, \theta \,|\, \pi) = P(o \,|\, \theta, \pi)\, P(\theta)$$

We set $P(\theta) = \frac{1}{n_\theta}$ for all θ, expressing that there is no prior knowledge about the model parameters. The likelihood distribution $P(o \,|\, \theta, \pi)$ is created randomly for each repetition of each experiment.

One approach to obtain random feasible likelihood distributions is by drawing random values for the probabilities of each triple (o, θ, π) and normalizing the distribution afterwards. However, we found this approach to yield distributions in which all policies are equally informative, which defeats the purpose of using an information measure. Hence, we purposefully make some policies more informative than others by defining for given θ_i and π_j:

$$P(o_k \,|\, \theta_i, \pi_j) \coloneqq \frac{e^{T_{ij} c_k}}{\sum_{l=1}^{n_o} e^{T_{ij} c_l}}$$

where

$$c_k \sim \mathcal{U}(0, 1)$$
$$T_{ij} \sim \mathcal{U}(0, 15)$$

where $\mathcal{U}$ is a uniform distribution.

Here, the variable T_{ij} can be understood as a temperature, which we draw individually for each pair (θ_i, π_j). The temperature controls how "spiky" the distribution $P(o_k \,|\, \theta_i, \pi_j)$ is for θ_i and π_j. If T_{ij} has a low value, it will be more uniform and less informative, and if the value is high will likely be more informative.

Experiment 1: Relative deviation from the exact measure

In this work, we use the information term to make decisions over which policies to follow. Hence, we are not interested in the exact absolute value of this term, but rather want to know which policies are more informative than others. Thus, an interesting question is how well the approximations resemble the policy preferences of their exact counterparts. Ideally, we want every policy with a high exact information value also to have a high approximated information value and vice versa.

Formally, we can think of the information term to produce a vector v of information values, one for each policy it is evaluated for:

$$v_{\text{IM}} \coloneqq \begin{pmatrix} \text{IM}(o, \theta \mid \pi_1) \\ \vdots \\ \text{IM}(o, \theta \mid \pi_{n_\pi}) \end{pmatrix}$$

where IM is an information measure (LI, MI or their approximated counterparts).

As argued before, we are not interested in the scale of this vector, but rather its direction, which tells us which policies to prefer. A standard method of comparing the directions of vectors is the cosine similarity, which is defined as

$$\text{cos-sim}\,(v, w) = \frac{v \cdot w}{\|v\|\,\|w\|}$$

for vectors $v, w \in \mathbb{R}^n$.

The cosine-similarity allows us to compare the approximations of both MI and LI with their exact counterparts on the same scale. We do so in an experiment on 1000 randomly generated generative models and report the results in Fig. 5.3. In this experiment, the LI approximator exhibits a clear advantage in sample efficiency, especially for low numbers of samples. Although we did not conduct a formal study of the convergence properties of the LI approximator, this experiment suggests that the exhaustive combination of θ and o samples we describe in Section 5.5.1 gives it an advantage over the MI approximator.

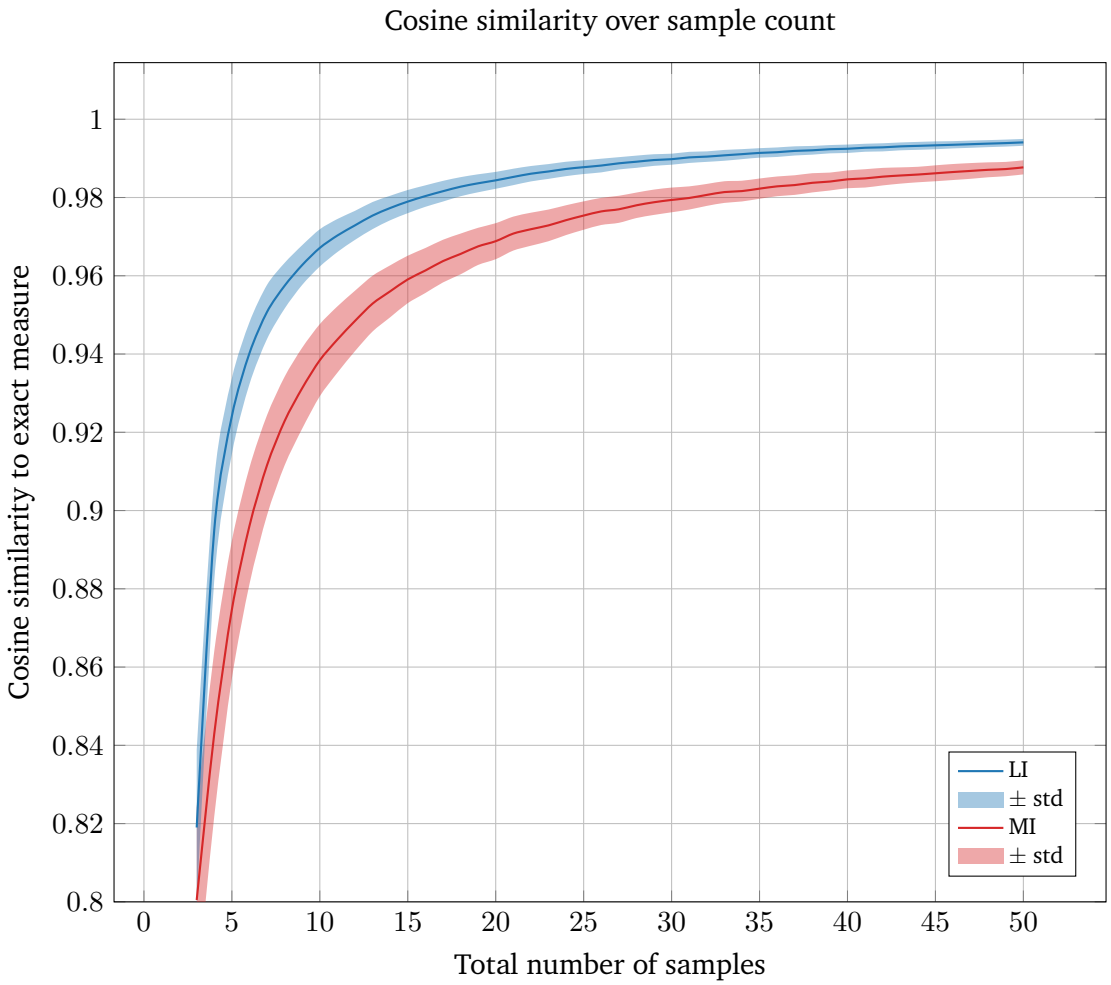


Figure 5.3: Experiment 1: Comparison of the cosine similarity between the (sample-reusing) NMC approximations of MI and LI to their respective exact counterparts. For each number of samples, we conducted 1000 experiments on different, randomly generated generative models. It is apparent that the LI approximator has a clear advantage in sample efficiency compared to MI. Note that the optimal value possible under cosine similarity is 1.

Experiment 2: Parameter identification

The goal of the second experiment is to evaluate the exploratory capabilities of the information measures and their approximations. In this experiment, we randomly draw a model parameter realization $\theta^* \sim P(\theta)$. The agent's objective is to determine θ^* based on observations o. To obtain these observations, the agent can query the generative model $P(o \mid \theta^*, \pi)$ three times with different policies π, obtaining a sample o each time. We assume that the agent can perform exact inference to update its belief over model parameters after each observation. After the third observation is obtained, we measure whether the θ^* is the parameter realization with the highest probability in the posterior. That is, we check whether

$$\theta^* = \arg\max_{\theta} P(\theta \mid o_{1:3}, \pi_{1:3})$$

where o_i is the observation the agent made in the i-th step and π_i is the policy of the i-th step.

We report the success probabilities for agents following the different information measures in their policy selection in Fig. 5.4. It seems as if the agent following the approximated LI for policy selection has a slight advantage over the agent following the approximated MI. However, due to the fairly large standard deviations, the difference is not significant.

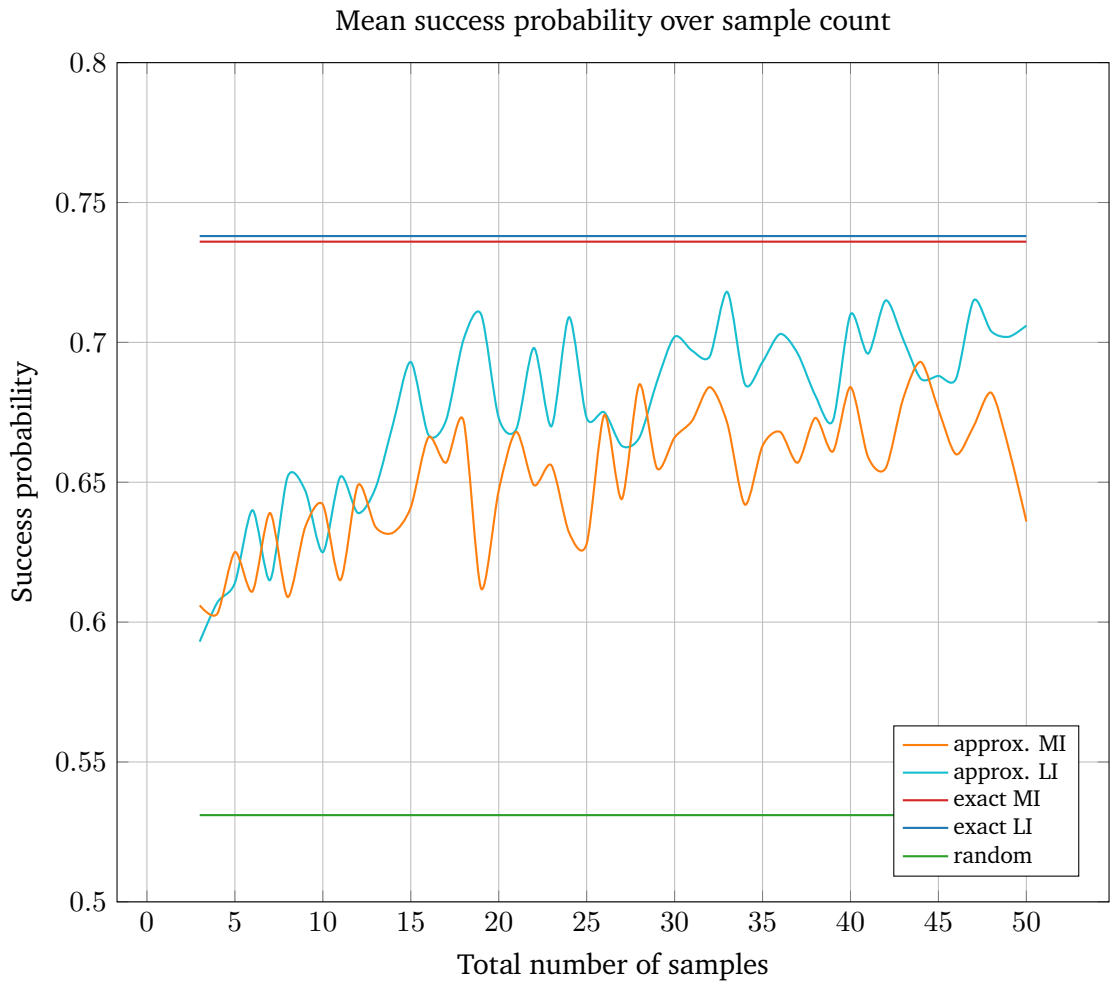


Figure 5.4: Experiment 2: Comparison of the success probabilities in identifying the correct model parameter realization θ^* for different numbers of samples used in the NMC approximations. For each number of samples we conducted 1000 experiments on different, randomly generated generative models. The baselines *exact MI*, *exact LI* and *random* are sample independent and only displayed for reference. We do not plot the standard deviations as they are too large ($\sim$ 0.45 for all estimators). Although the approximated LI seems to be slightly ahead of the approximated MI, the advantage is not significant due to the large standard deviations. However, both outperform the random baseline significantly.

5.5.4 Lautum Information in Bayesian Optimal Experimental Design

As mentioned before, unlike Mutual Information, Lautum Information has not seen much attention in the BOED-literature so far. In this section, we conducted multiple investigations that suggest that LI is indeed a viable alternative to MI. We showed empirically that Lautum Information allows a more sample efficient Monte Carlo approximation than MI and that it can keep up with, if not outperform, the information seeking capabilities of Mutual Information. Although preliminary, we believe that these results justify further research into the usage of Lautum Information in BOED. To further strengthen this point, we conduct experiments with LI on real RL problems in Chapter 6.

5.6 Complete learning algorithm

To conclude this chapter, we outline the resulting procedure in Algorithm 3. Note that for performance reasons, we chose not to update the variational parameter distribution $q_\phi(\theta)$ every step, but rather after every episode. To facilitate these delayed updates and also to prevent catastrophic forgetting, we deploy a replay buffer.

Algorithm 3 Reinforcement Learning through Active Inference

Initialize replay buffer
Initialize ensemble parameters $\theta^1, \ldots, \theta^P$
for $e = 1, \ldots,$ MAX_EPISODES **do**
    Reset agent to initial state x_0
    for $t = 1, \ldots, T_{\text{max}}$ **do**
        Execute CEM planner (Algorithm 1) to obtain next action a_t
        Execute action a_t on environment
        Obtain new state x_t and reward r_t from environment
        Store (x_t, r_t, a_t) in replay buffer
    end for
    Optimize variational parameters ϕ for M steps using the replay buffer
end for

6 Experimental Evaluation

A central feature that sets our method apart from other purely model-based approaches [26–28, 64–68] is the intrinsic term, that explicitly drives the agent to explore its environment. Hence, in the following four experiments, we put the exploration capabilities of our method to test. In each experiment, the agent is faced with an environment that is designed to be particularly sparse and, thus, hard to explore. We will show that our method results in systematic, directed exploration, and manages to solve environments in which state-of-the-art methods that do not explicitly encourage directed exploration fail.

In all of the following environments, the agent has to reach some goal state, that can only be reached by precisely executing a sequence of actions. It does not receive any reward signal until it has solved the respective task, except for a small quadratic penalty on all actions. We terminate each episode after $T_{\text{max}} = 50$ steps, independently of if or when the agent reached the goal state. The planning horizon is set to $T = t + 20$, where t is the current time step, in all experiments, and the environments are designed in such a way that the goal state is reachable within the planning horizon. Since all environments are fully deterministic, we set $\sigma^x = \sigma^r = 10^{-4}$, to reduce unwanted noise during planning as much as possible.

In each experiment except for the first, we compare multiple versions of our method to a Soft Actor Critic (SAC) [69] baseline. The reason we chose SAC as baseline is that its entropy regularization should be advantageous for exploring sparse environments, compared to other model-free algorithms like TD3 [45] or PPO [70]. To ensure proper implementation and tuning of the SAC baseline, we use the Stable Baselines3 [71] implementation. Also, since the execution of SAC is computationally comparatively cheap, we repeat each SAC experiment 5 times and report the best result.

Furthermore, we attempted to compare against MBPO [25]. However, we found that neither the original implementation by Janner et al. [25], nor the newer implementation of MBRL-Lib [72] were stable enough to endure longer experiments without crashing.

Abbreviation	Description
MI	Standard configuration where we use Mutual Information as intrinsic term.
MI no RMSP	Similar to the MI configuration, with the exception that we do not use the multi-step prediction loss for the reward model (see Section 5.2.2).
LI	In this configuration we use Lautum Information instead of Mutual Information as intrinsic term.
LI no RMSP	Similar to the LI configuration, with the exception that we do not use the multi-step prediction loss for the reward model (see Section 5.2.2).
non-intrinsic	In this configuration we remove the intrinsic term entirely and plan purely based on the expected reward.
non-intrinsic AN	Similar to the *non-intrinsic* configuration, with the exception that we apply Gaussian noise to the actions before we execute them. Specifically, we take the action distribution produced by the CEM planner and sample from it instead of taking the mean.

Table 6.1: Descriptions of different configurations of our method with abbreviations.

Hence, we were only able to obtain a MBPO baseline for the *Ridge Ball* environment (see Section 6.2).

For notational convenience, we will use abbreviations for different configurations of our method. A description of these abbreviations can be found in Table 6.1.

6.1 Mountain Car

In the *Mountain Car* environment [20], the agent's objective is to reach the top of the hill, marked with a yellow flag in Fig. 6.1. To achieve this objective, the agent can control the force the car is exerting in either direction. Apart from a small action penalty, the agent receives a reward of 1 for every time step in which the car is located at or right of the flag.

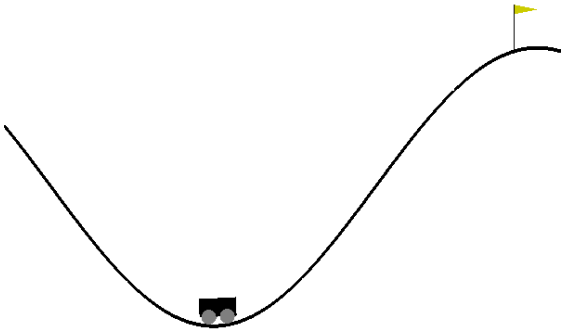


Figure 6.1: Visualization of the *Mountain Car* environment generated by OpenAI gym [20].

There are two things that make this task challenging. First, at the beginning, the agent does not know that its objective is to reach the top of the hill. The only way it can find out that it will receive reward when reaching the flag is by moving up the hill and discovering it. Second, the maximum force the car can exert is too small to move it up the hill from a stationary condition. Hence, it has to move up the left hill first to gain enough momentum to reach the hilltop. Both of these properties make *Mountain Car* a fairly challenging environment to solve, despite its low dimensional state space.

As visible in Fig. 6.2, all of our configurations manage to solve the environment within 50 episodes. Although the intrinsic configurations (*MI* and *LI*) reach the hilltop earlier than the non-intrinsic agents, their performance equalizes quickly after a couple of episodes. From Fig. 6.3, it can be seen that the reason for the intrinsic configurations being faster is a better coverage of the state space within the first 10 episodes.

We conclude that this environment is not challenging enough to require information seeking behavior to be solved. A reason for the rather small challenge this environment poses is most likely its low dimensional state space, that can rather quickly be searched even by undirected exploratory behavior. Hence, in the next experiment, we test our method on a higher-dimensional and harder-to-explore environment.

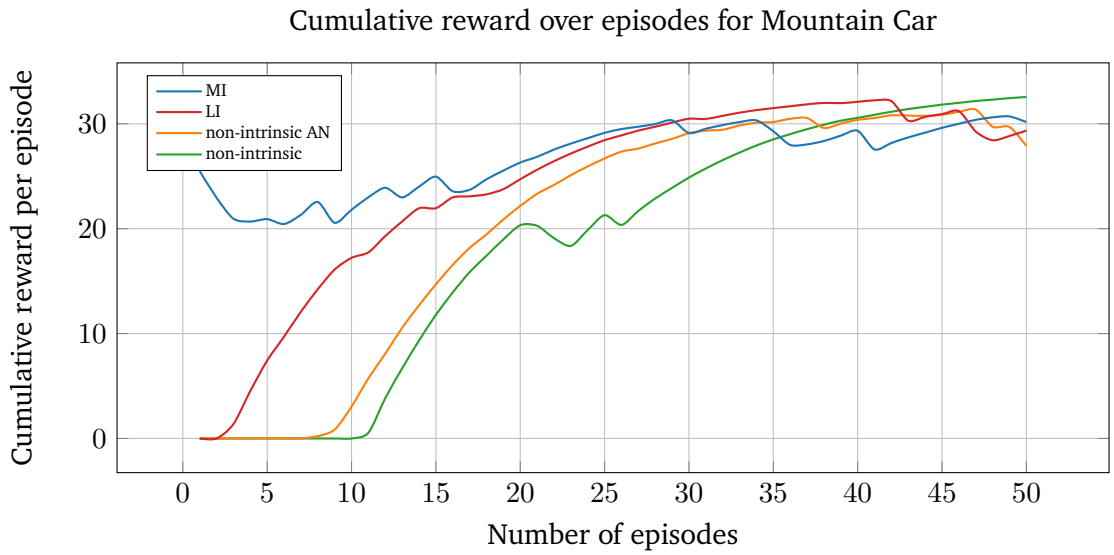


Figure 6.2: Cumulative per-episode reward on *Mountain Car* over the course of the training. Displayed in this graph is the evaluation reward, which is obtained by rolling out the learned model without considering the intrinsic reward. We apply exponential moving average smoothing with a discount of 0.9 to make the graph more readable. Each configuration was run once. For a description of the configurations, refer to Table 6.1.

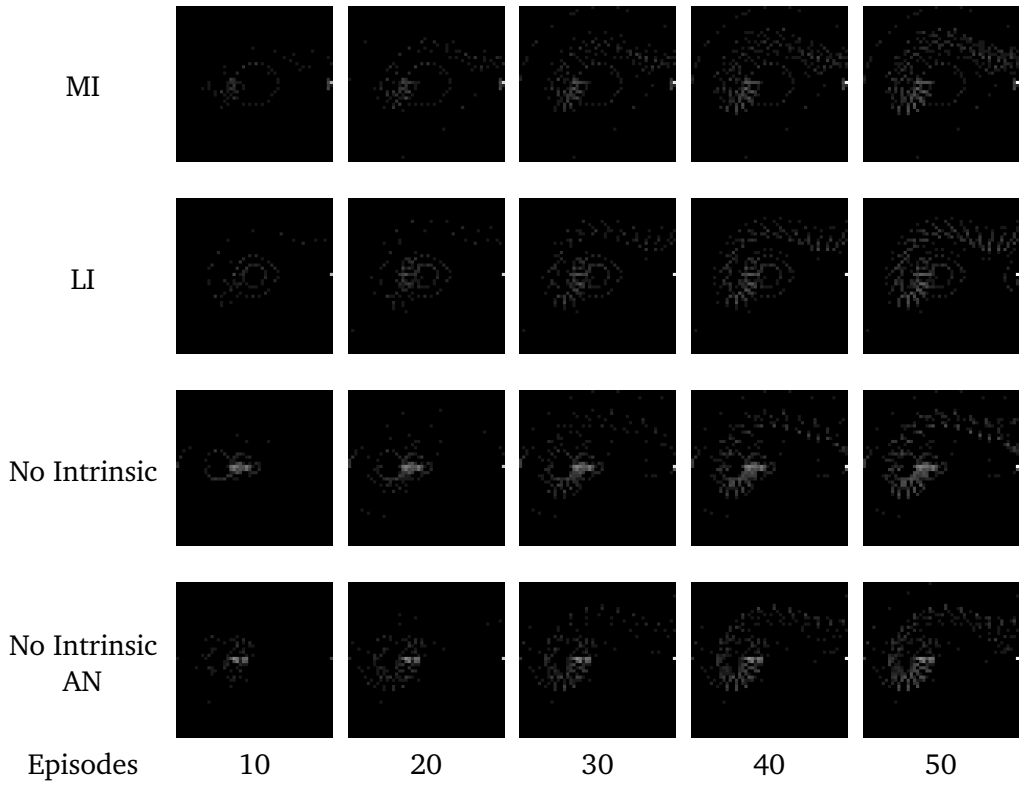
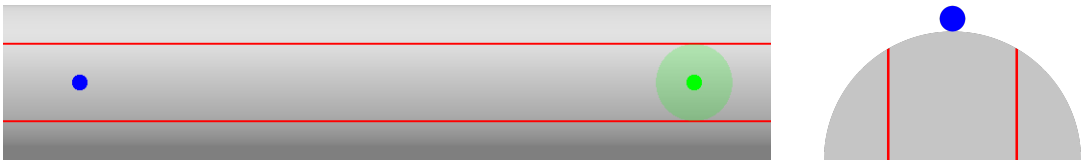


Figure 6.3: Cumulative state visitation histograms for the *Mountain Car* environment. Position of the car is on the X-axis and velocity on the Y-axis. The brightness of each pixel indicates how often the respective histogram bin has been visited. Each configuration was run once. For a description of the configurations, refer to Table 6.1.

6.2 Ridge Ball

The objective in this environment is to control the blue ball (see Fig. 6.4) in such a way that it reaches the green target zone. Unless the agent has reached the target zone, the only reward signal it is getting is a quadratic penalty on its actions. To achieve this objective, the agent can exert horizontal force in an arbitrary direction on the ball. The ball's 2D position and velocity serve as input to the agent

The main challenge in this environment is to control the ball in such a way that it stays on top of the ridge. Since the control force that the agent can exert is limited, there exists a point of no return on either side of the ridge, marked by red lines. As soon as the ball leaves the area between the red lines, the agent has no way of recovering it and has to wait for the episode to terminate before it can continue exploring.



(a) Top-down view (b) Side view

Figure 6.4: Visualization of the *Ridge Ball* environment.

Situations in which an agent cannot recover are a common occurrence in manipulation tasks, since manipulated objects might be accidentally dropped or moved into positions where they become unreachable. These dead-ends pose a significant challenge to RL, as the agent first has to learn to avoid them before it can make progress on its task. This challenge is only exponentiated if combined with a sparse reward, as it forces the agent to learn to avoid dead-ends without getting any reward as guidance. Since this environment provides both, sparse rewards and dead-ends, we believe it is a good intermediate step towards complex manipulation tasks.

The results of this experiment can be seen in Fig. 6.5, where we visualize the cumulative per-episode reward obtained by each configuration over the course of the training. As visible in this graph, three out of four intrinsic configurations found the target zone within the first 400 episodes. One out of five SAC agents we ran on this task managed to find the target zone around the 2200th episode and converged towards a near optimal policy. MBPO, both of the non-intrinsic configurations, as well as the LI configuration failed to find the target and converged to a local optimum. Out of the three successful intrinsic configurations, the *MI no RMSP* configuration outperformed the other two significantly. From the state visitation histograms in Fig. 6.6 it seems that *MI no RMSP* managed to better stabilize the ball on the target zone, compared to the other configurations.

One of the reasons for the performance discrepancy of the four intrinsic configurations is

most likely the choice of the extrinsic weight parameter σ_r^*. To achieve optimal performance, σ_r^* has to be carefully chosen for each configuration individually. Judging from Fig. 6.6, it seems as if the *LI* configuration stopped exploring right at the start, which is a sign for σ_r^* being too low. While restarting this experiment with a different value of σ_r^* could yield better results, we decided that the benefit would be little, as the *Ridge Ball* environment is merely a toy task.

In conclusion, this experiment has shown that our method is able to systematically explore a challenging environment with sparse rewards and many dead-ends. The stark contrast in exploration efficiency compared to the non-intrinsic configurations and the baselines shows the importance of an information seeking term for environments of this kind. However, due to the environment's relatively simple dynamics and the low dimensional state space, the question arises whether our method is able to scale to more complex, higher dimensional tasks. We will approach this question in the following two experiments, in which we apply our method to a robotic manipulation problem.

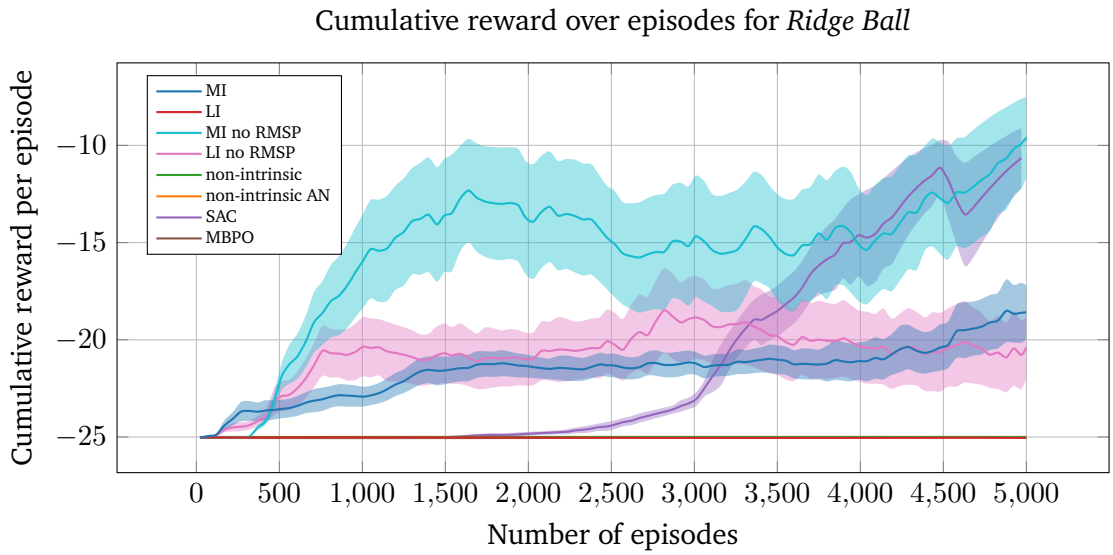


Figure 6.5: Cumulative per-episode reward on *Ridge Ball* over the course of the training. The shaded area around each line is a half a standard deviation. MBPO, both non-intrinsic configurations, as well as the LI configuration failed to find the objective and converged to local minima. Displayed in this graph is the evaluation reward, which is obtained by rolling out the learned model without considering the intrinsic reward. We apply exponential moving average smoothing with a discount of 0.9 to make the graph more readable. Each configuration was run once, except for SAC, where we show the only successful run out of 5. For a description of the configurations, refer to Table 6.1.

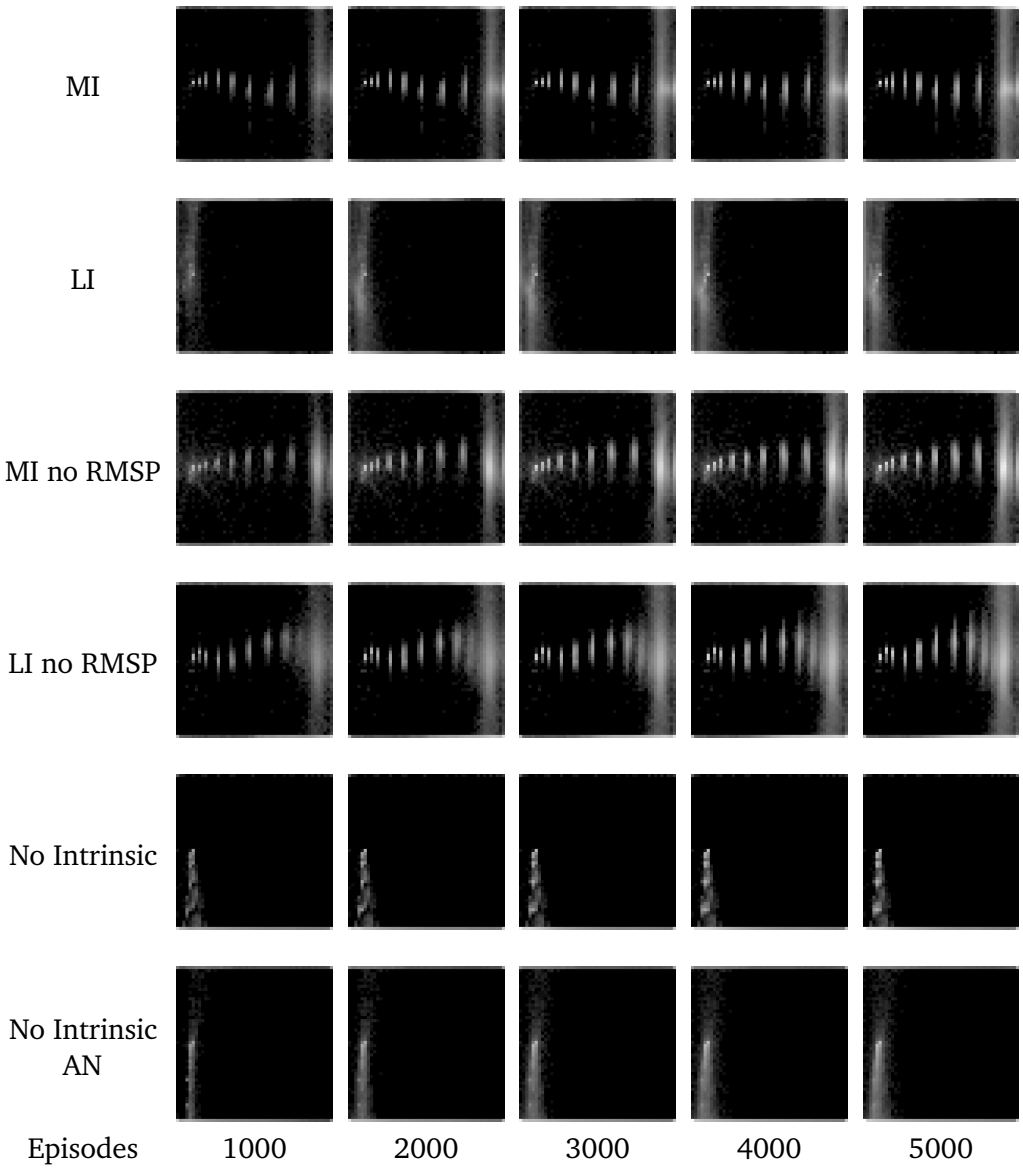
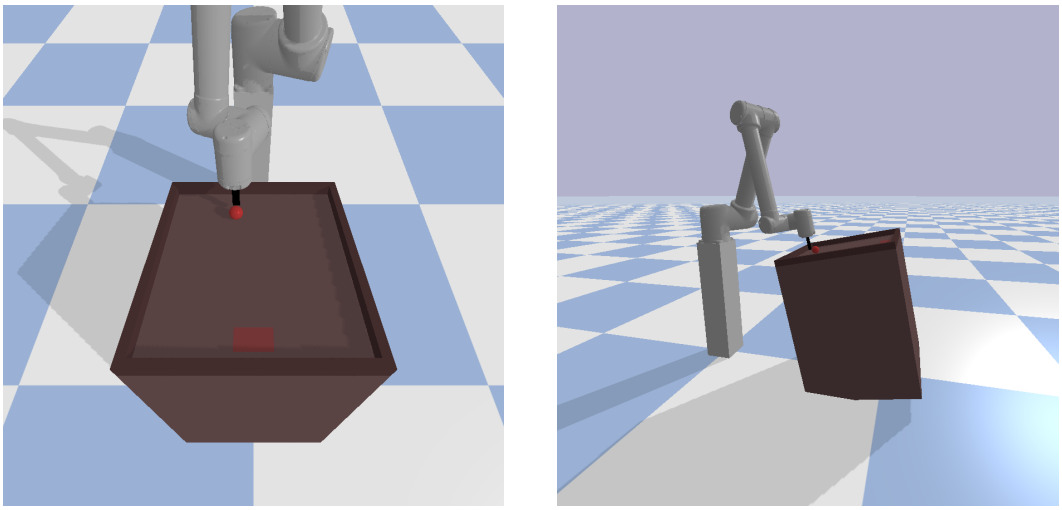


Figure 6.6: Cumulative state visitation histograms for the *Ridge Ball* environment. We visualize the 2D position of the ball. Note that X and Y axes are scaled to an aspect ratio of 1:1. The brightness of each pixel indicates how often the respective histogram bin has been visited. Each configuration was run once. For a description of the configurations, refer to Table 6.1.

6.3 Tilted Pushing

While the previous experiment showed that our method is capable of efficiently exploring sparse environments, we have yet to show that this property is maintained if we increase the complexity of the environment. Hence, in this experiment, we apply our method to a sparse and particularly hard to explore robotic manipulation task.

In this task, which we call *Tilted Pushing* and visualize in Fig. 6.7, the agent has to push a ball into a target zone on a tilted table. Similarly to *Ridge Ball*, the agent does not receive any reward except for a quadratic action penalty, as long as the ball is outside of the target zone. The gripper always starts on the same Y coordinate (Y-axis being the axis that points from the robot towards the goal) while the X coordinate is randomized. At the start of each episode, the ball is placed directly in front of the gripper.



(a) Top-down view (b) Side view

Figure 6.7: Visualization of the *Tilted Pushing* environment. The target zone is marked in red. For reference, in the top-down view, the X-axis points left, the Y-axis in direction of the viewer and the Z-axis upwards.

The agent can move the gripper in a plane parallel to the table and rotate the black end-effector around the Z-axis (Z-axis being the axis that is orthogonal to the brown table and points up). Specifically, the agent controls the horizontal 2D linear acceleration of

the gripper as well as the angular velocity of the end-effector. As input, the agent receives the 2D positions and velocities of both the gripper and the ball, as well as the angular position and velocity of the end-effector.

What makes this task particularly challenging is that due to the tilted table, the ball has to be balanced constantly during exploration. If the ball drops, it cannot be recovered, since the gripper cannot move close enough to the boundary to retrieve it again. Hence, similar to *Ridge Ball*, the agent has to wait for the episode to terminate before it can continue exploring. However, in contrast to *Ridge Ball*, the environment is much higher dimensional and its complex, contact-rich dynamics make learning the model challenging.

In addition to the standard configuration, we run this experiment in another configuration, where we decreased the friction of the ball by 40%. The results for both configurations are shown in Fig. 6.8. Furthermore, we visualize the state coverage for both configurations in Figs. 6.9 and 6.10, respectively.

From Fig. 6.8 we can see that only the intrinsic agents are able to solve this environment within the given number of episodes. It seems that the configurations that utilize reward multi-step prediction loss are faster to receive the first reward and also exhibit a better asymptotic convergence. However, since we ran only a single repetition per agent and environment configuration, no definitive statement can be made.

From the state coverage histograms in Figs. 6.9 and 6.10 it becomes apparent that the intrinsic configurations achieve a much better state coverage than the non-intrinsic configurations. With the exception of the *No Intrinsic AN* run in the standard configuration, none of the non-intrinsic configurations managed to push the ball into the upper half of the table in any of the episodes. Thus, the broad state coverage we observe for the intrinsic agents must be a direct consequence of the information seeking behavior mandated by the intrinsic term.

This experiment shows 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 systematically move it around the environment until the target zone was found. The sole reason for this behavior to occur in the first place is that our agents understood they could only explore the full state space if they kept balancing the ball and move it to unseen locations.

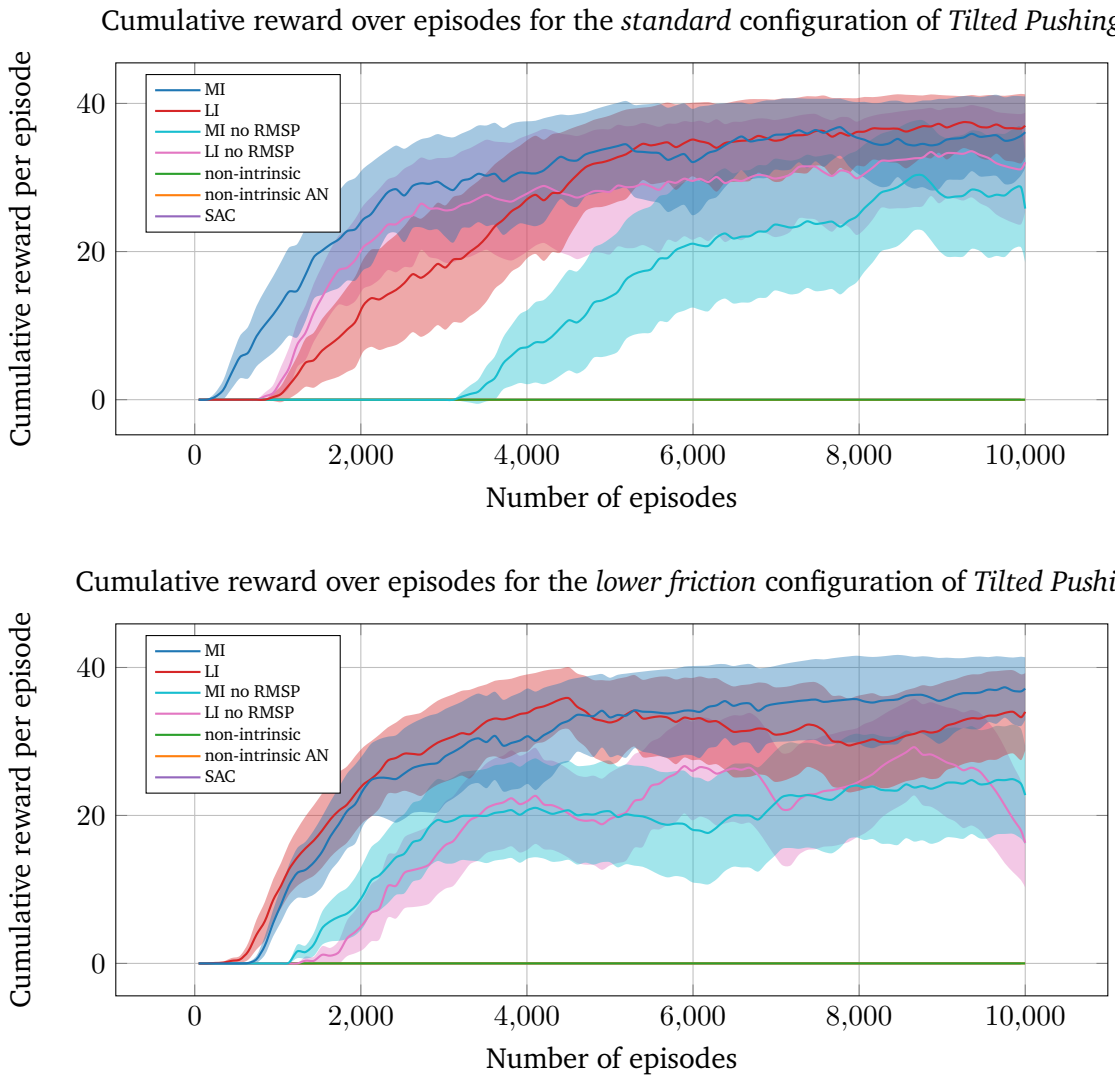


Figure 6.8: Cumulative per-episode reward for both *Tilted Pushing* configurations over the course of the training. The shaded area around each line is a half a standard deviation. Both non-intrinsic configurations and all SAC agents failed to find the objective and converged to local minima. Displayed in this graph is the evaluation reward, which is obtained by rolling out the learned model without considering the intrinsic reward. We apply exponential moving average smoothing with a discount of 0.9 to make the graph more readable. Each (agent-) configuration was run once, except for SAC, which we ran 5 times. For a description of the configurations, refer to Table 6.1.

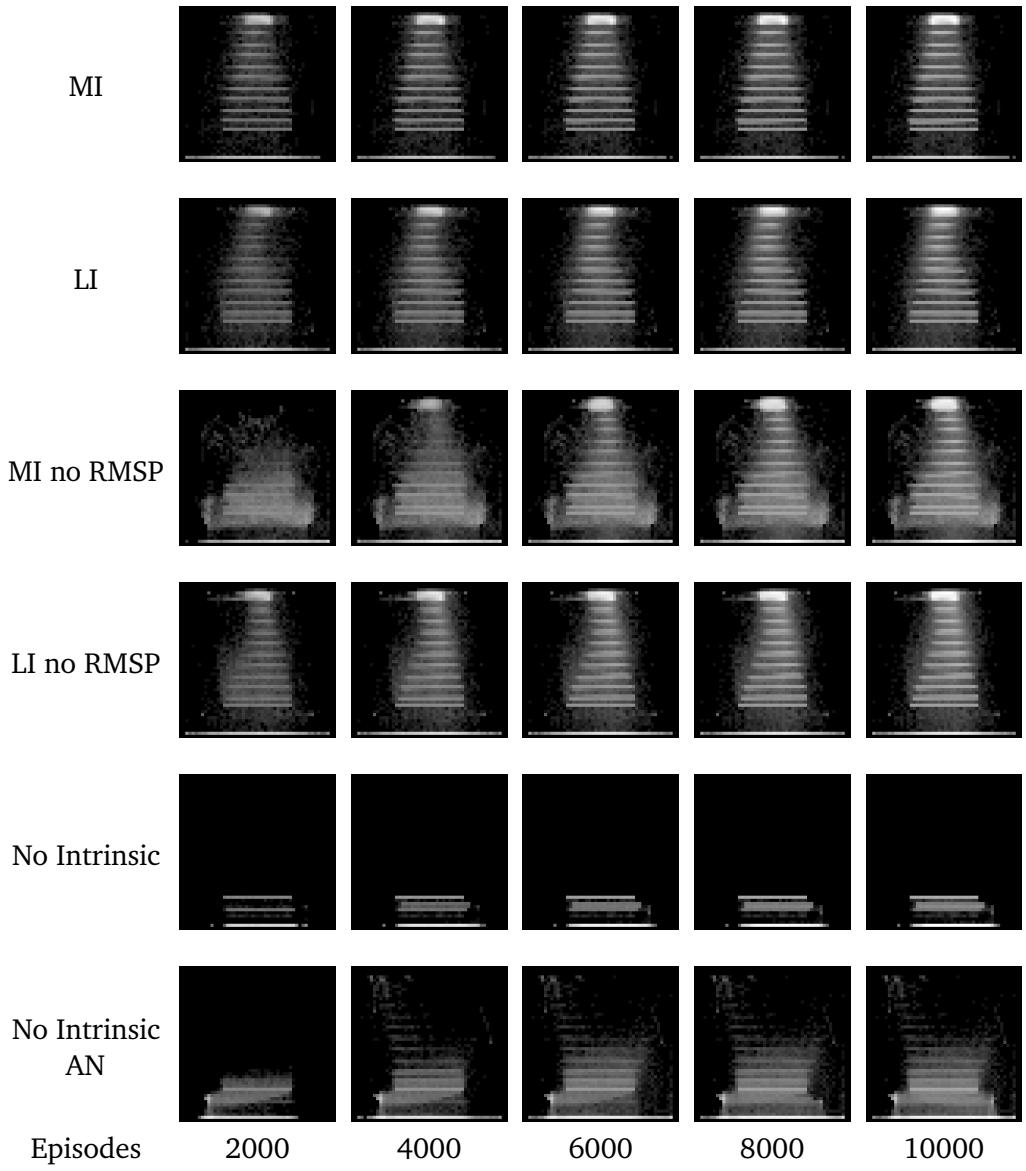


Figure 6.9: Cumulative state visitation histograms for the standard configuration of the *Tilted Pushing* environment. We visualize only the 2D position of the ball on the table. The coordinate origin is at the bottom of each image, meaning that the images are rotated 180°compared to the top-down view in Fig. 6.7a. The brightness of each pixel indicates how often the respective histogram bin has been visited. Each configuration was run once. For a description of the configurations, refer to Table 6.1.

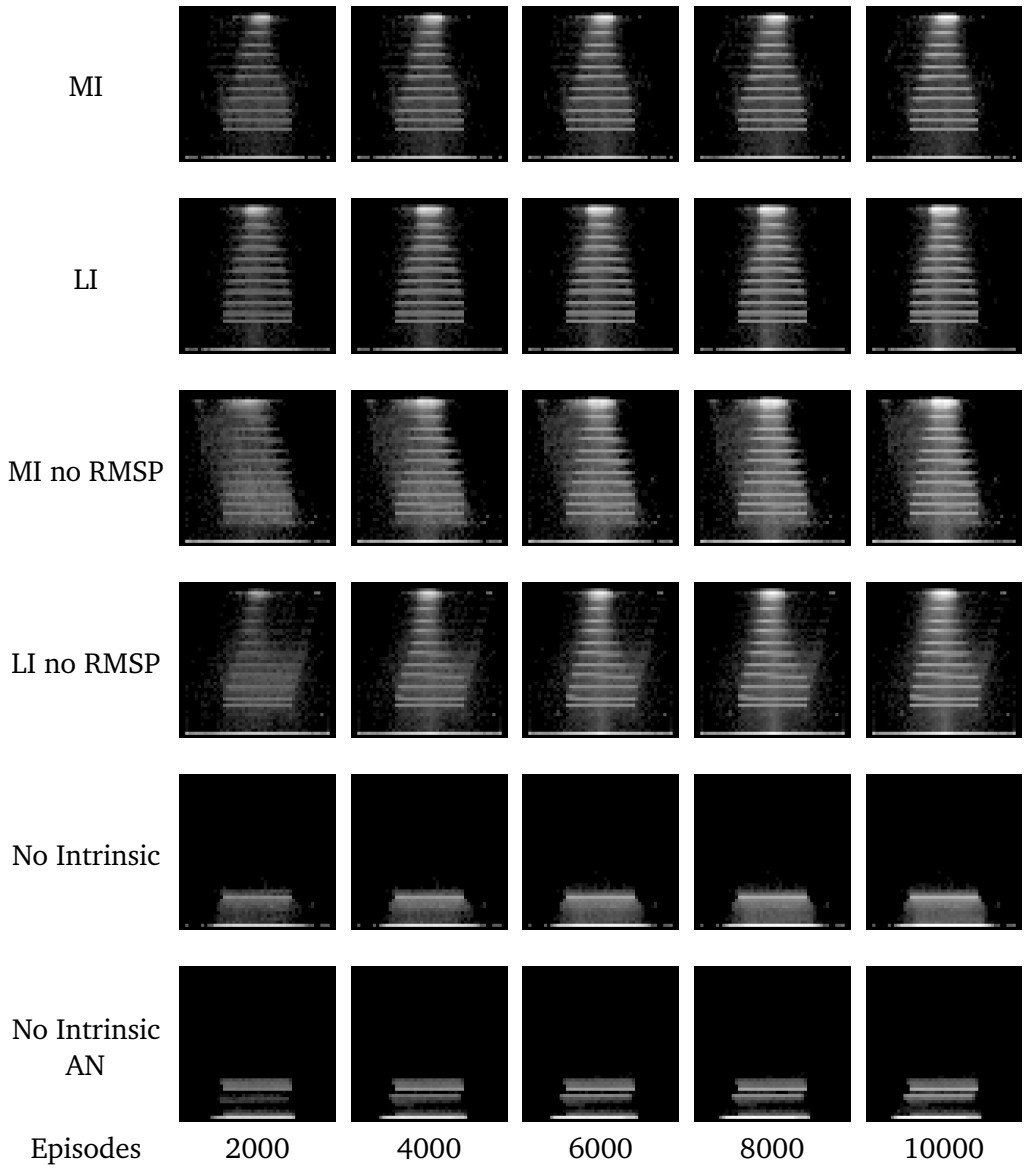
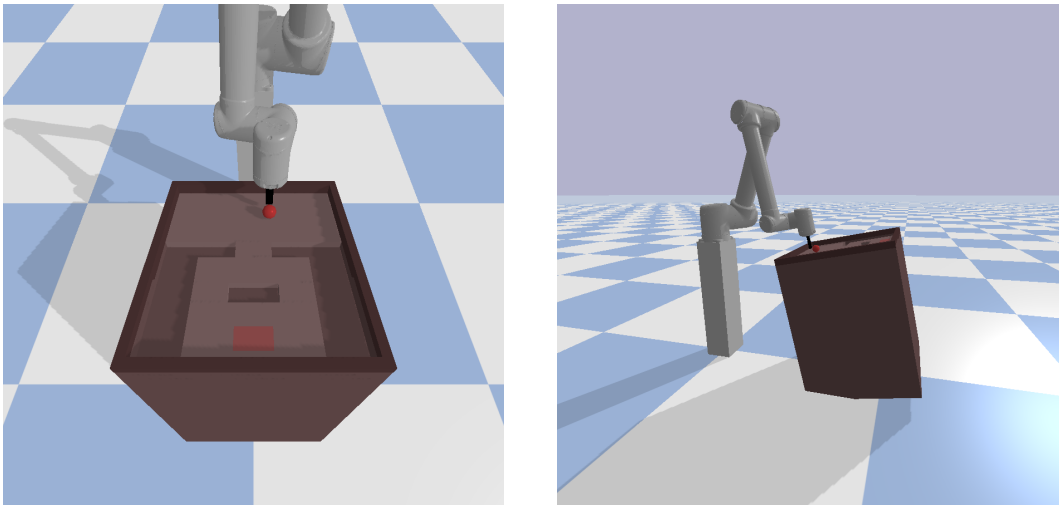


Figure 6.10: Cumulative state visitation histograms for the lower friction configuration *Tilted Pushing* environment. For a further description of this figure, refer to Fig. 6.9.

6.4 Tilted Pushing Maze

In this experiment, we take the idea of *Tilted Pushing* one step further and test the limits of our method. As shown in Fig. 6.11, we use the same setup as before, but remove parts of the surface of the table. In doing so, we create a maze that the agent has to navigate with the ball in order to reach the target zone. The resulting task poses a much greater challenge than before, since the agent now has to perform multiple curves with the ball to avoid the holes in the table.



(a) Top-down view (b) Side view

Figure 6.11: Visualization of the *Tilted Pushing Maze* environment.

The results of this experiment are shown in Fig. 6.12 and the corresponding state coverage histograms in Fig. 6.13. As the graph shows, two out of four of our intrinsic configurations found the target zone within 35,000 episodes. Judging from the state coverage histograms, the other two configurations brought the ball close to the target zone multiple times, but did not yet discover it when the training terminated. However, it is likely that with a longer training time these configurations would have found the target as well.

We believe that there are two reasons that prevent the intrinsic agents from finding the target earlier in this environment. The first reason is the limited ability of the CEM planner to find globally optimal solutions. Since CEM is a local optimization procedure, it tends to

converge to local minima. In case of sparse environments, those minima are often policies that do not move at all to avoid the small action penalties. We mitigated this issue to some degree by taking the initial sample partially from a buffer of previously optimized action trajectories. Hence, once a good trajectory had been found, it could be reused in a later episode. However, this technique does not help us in finding completely novel trajectories. In fact, it might even discourage the planner from exploring new trajectories, as any new trajectory it samples has to compete with fully optimized trajectories from the buffer in order to make it into the next iteration.

The second and more fundamental reason is that the state space of this environment might be too large to be exhaustively explored within the given number of episodes. Although the state space of this environment is equal in dimensions to the state space of the *Tilted Pushing* environment, its target state is much harder to reach. Thus, in *Tilted Pushing* the agents were simply much more likely to randomly come across the target state during exploration than they are in this environment. Consequently, the reason the agents found the target state in the previous environment was not that they exhaustively explored the entire state space, but rather that they explored enough to come across it with a reasonable probability. Since the agents in this environment must first carefully navigate around the holes to reach the target state, many action trajectories that would have been successful in *Tilted Pushing* simply end up in holes in this environment. Hence, it can be expected that the *Tilted Pushing Maze* must be explored to a much higher degree until there is a reasonable probability of finding the target state.

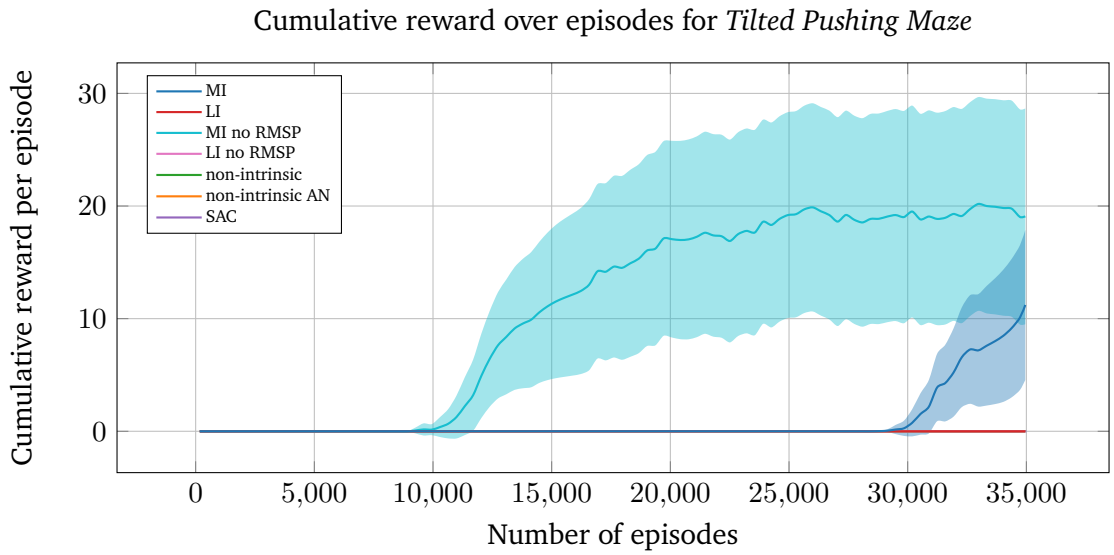


Figure 6.12: Cumulative per-episode reward on *Tilted Pushing Maze* over the course of the training. The shaded area around each line is a half a standard deviation. Only the *MI* and *MI no RMSP* configurations managed to find the target zone with the ball. Displayed in this graph is the evaluation reward, which is obtained by rolling out the learned model without considering the intrinsic reward. We apply exponential moving average smoothing with a discount of 0.9 to make the graph more readable. Each (agent-) configuration was run once, except for SAC, which we ran 5 times. For a description of the configurations, refer to Table 6.1.

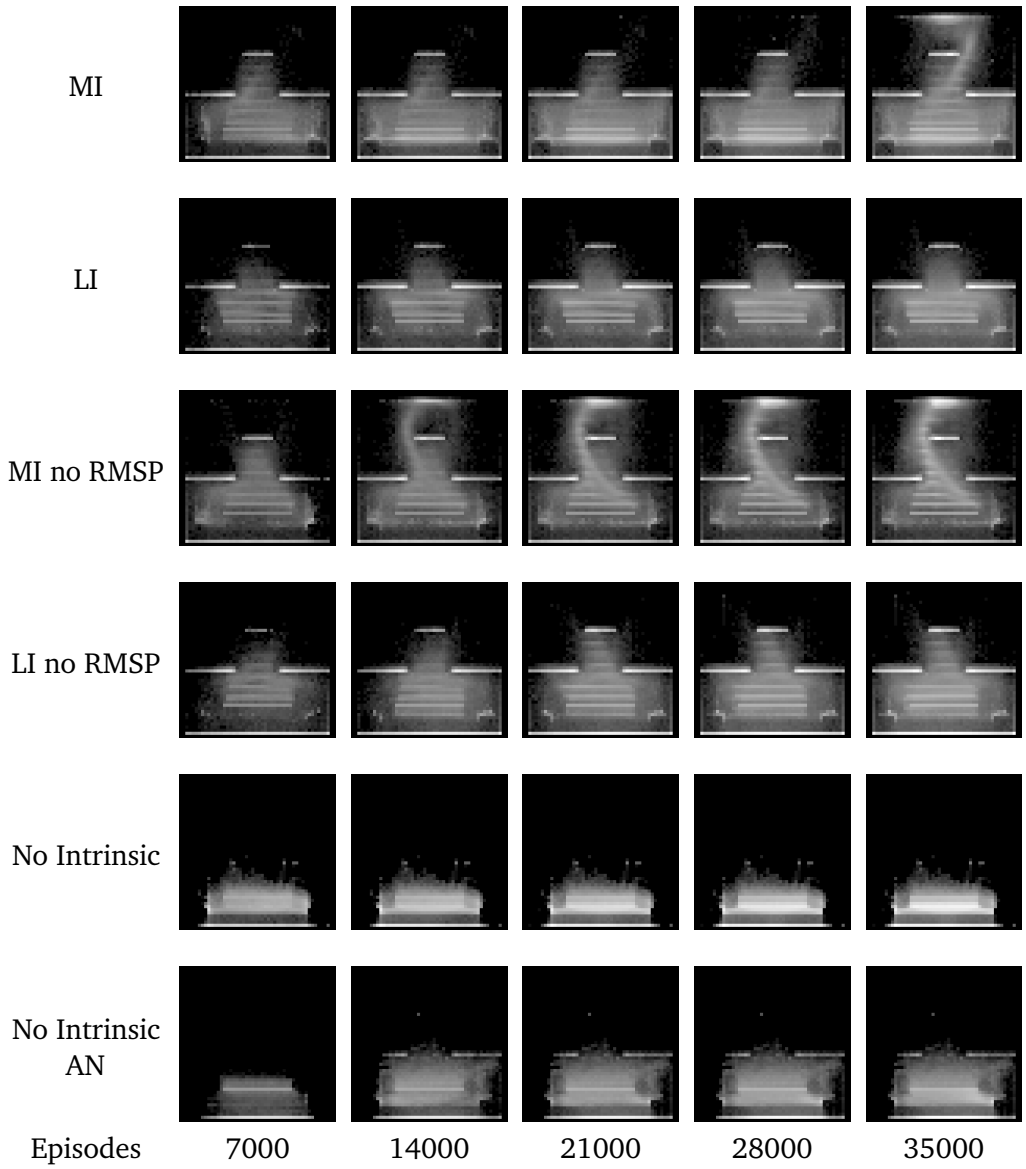


Figure 6.13: Cumulative state visitation histograms for the *Tilted Pushing Maze* environment. We visualize only the 2D position of the ball on the table. The coordinate origin is at the bottom of each image, meaning that the images are rotated 180°compared to the top-down view in Fig. 6.7a. The brightness of each pixel indicates how often the respective histogram bin has been visited. Each configuration was run once. For a description of the configurations, refer to Table 6.1.

7 Discussion and Future Work

Our contributions in this work are as follows. First, we provided an in-depth formal analysis of the Expected Free Energy functional. In that analysis, we took a close look at the origin of the intrinsic term and at the claim that the EFE resolves the exploration-exploitation dilemma. We argued that the EFE does not resolve the exploration-exploitation dilemma in practice, as the weighting between the extrinsic and intrinsic terms remains a challenge. Furthermore, we showed formally that the information gain only arises naturally from the EFE if one makes the assumption that the agent's hidden desire distribution is always equal to its predictive distribution. Since this assumption is violated in most cases, we concluded that the information gain does not arise naturally from the EFE. To support this argument, we presented an example task in which only the agent following the approximated EFE exhibited exploratory drive, while the one following the exact EFE did not. Finally, we questioned the origin of the EFE and, like Millidge, Tschantz, and Buckley [43], arrived at the conclusion that there is a more naturally arising objective that does not encourage exploratory behavior.

Our second contribution is the development of a method that is capable of applying Active Inference to complex Reinforcement Learning tasks. To achieve this, we first showed how a classical reward-based RL task can be reformulated to fit into the framework of AI. We then showed how to train a transition and reward model with multi-step predictions, and proposed to extend the idea of multi-step predictions to the reward model. For the planning step, we developed an adaptation of the Cross Entropy Method, that reuses previous trajectories to obtain a better initialization and is thereby able to create more promising trajectories.

The third contribution of this work lies in our approximation of the information term. We argued that the only approximation that is efficient enough for our use case is a Nested Monte Carlo approximation. As parameter samples are extremely expensive in our case, we showed empirically how the reuse of samples, despite breaking the i.i.d. assumption, can significantly improve the sample efficiency of the NMC approximator.

Furthermore, we proposed to replace the Mutual Information term in the planning objective by Lautum Information. We argued that the NMC approximation of the Lautum Information allows for a better reuse of samples, and showed empirically that the Lautum Information estimator converges faster than the Mutual Information estimator. However, we were not yet able to show that the faster convergence of this estimator yields any advantage for the exploratory capabilities of the agent in practice. To our knowledge, Lautum Information has not yet been used in the context of Bayesian Optimal Experimental Design.

Finally, we thoroughly evaluated our method in four experiments, ranging from the classical *Mountain Car* to two complex robotic manipulation tasks. All environments were specifically designed to be challenging to explore, with three of them containing dead-ends that can trap the agent in an irrecoverable state. Over the course of these experiments, we showed that our method induces systematic exploration behavior and is capable of solving even the most challenging of these environments. Neither the non-intrinsic configurations, nor the maximum entropy method SAC managed to solve the robotic manipulation tasks. Hence, we conclude that the information seeking behavior of our agents is beneficial for solving hard exploration problems with sparse rewards.

Limitations and future work

As mentioned before, an issue of our method is the limited ability of the CEM planner to find globally optimal policies. This issue could be tackled by replacing our planner with a more sophisticated algorithm, like Monte Carlo Tree Search (MCTS). MCTS had remarkable success in the application to board games [73, 74], and is also applicable on continuous action spaces [75, 76]. In MCTS, one makes use of a policy to decide which actions are worth pursuing and which are not. A challenge that would arise from using MCTS for Active Inference is that the policy would have to be trained to predict interesting actions. However, unlike in regular RL problems, where the optimal Q function is fixed, our notion of an interesting action constantly changes due to the intrinsic term. Hence, one would either have to ignore the intrinsic term during training of the policy or find a way to track the changing values of the intrinsic term with the policy. Recently, the latter method has been used on two pixel-based RL tasks [77].

Another issue of our method is the large number of episodes the agent requires to understand the environment and solve the task. Since our agents start their tasks with no prior knowledge, they do not have any choice but to exhaustively search the entire state space for reward. Hence, an interesting future research direction is to provide the agent

with inductive biases that allow them to generalize more quickly and learn the model in fewer episodes. In prior work inductive biases have been introduced to neural networks modelling fluid systems [78] and robot dynamics [79].

A further potential future research direction is to extend our approach to work with POMDPs instead of MDPs. The merit of such an extension is that it enables the method to work on visual data, where typically the system state cannot be fully observed. In the context of robotic manipulation, processing visual data is particularly interesting, as it allows the incorporation of vision based tactile sensors [80, 81].

Bibliography

[1] Robert MacDougall. “The significance of the human hand in the evolution of mind”. In: *The American Journal of Psychology* 16.2 (1905), pp. 232–242.

[2] Richard W Young. “Evolution of the human hand: the role of throwing and clubbing”. In: *Journal of Anatomy* 202.1 (2003), pp. 165–174.

[3] Agnes Lacreuse and Dorothy M Fragaszy. “Manual exploratory procedures and asymmetries for a haptic search task: A comparison between capuchins (Cebus apella) and humans”. In: *Laterality: Asymmetries of Body, Brain and Cognition* 2.3-4 (1997), pp. 247–266.

[4] Michael T Turvey and Claudia Carello. “Dynamic touch”. In: *Perception of space and motion* (1995), pp. 401–490.

[5] Karl J Friston et al. “Action and behavior: a free-energy formulation”. In: *Biological cybernetics* 102.3 (2010), pp. 227–260.

[6] Karl Friston et al. “Active inference and epistemic value”. In: *Cognitive neuroscience* 6.4 (2015), pp. 187–214.

[7] Daniel P Palomar and Sergio Verdú. “Lautum information”. In: *IEEE transactions on information theory* 54.3 (2008), pp. 964–975.

[8] Guillermo Oliver, Pablo Lanillos, and Gordon Cheng. “Active inference body perception and action for humanoid robots”. In: *arXiv preprint arXiv:1906.03022* (2019).

[9] Léo Pio-Lopez et al. “Active inference and robot control: a case study”. In: *Journal of The Royal Society Interface* 13.122 (2016), p. 20160616.

[10] Manuel Baltieri and Christopher L Buckley. “An active inference implementation of phototaxis”. In: *Artificial Life Conference Proceedings 14*. MIT Press. 2017, pp. 36–43.

[11] Corrado Pezzato, Riccardo Ferrari, and Carlos Hernández Corbato. “A novel adaptive controller for robot manipulators based on active inference”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 2973–2980.

[12] Mohamed Baioumy et al. “Active inference for integrated state-estimation, control, and learning”. In: *arXiv preprint arXiv:2005.05894* (2020).

[13] Corrado Pezzato et al. “Active inference for fault tolerant control of robot manipulators with sensory faults”. In: *International Workshop on Active Inference*. Springer. 2020, pp. 20–27.

[14] Takazumi Matsumoto and Jun Tani. “Goal-directed planning for habituated agents by active inference using a variational recurrent neural network”. In: *Entropy* 22.5 (2020), p. 564.

[15] Ozan Çatal et al. “Deep active inference for autonomous robot navigation”. In: *arXiv preprint arXiv:2003.03220* (2020).

[16] Pablo Lanillos and Gordon Cheng. “Active inference with function learning for robot body perception”. In: *Proc. Int. Workshop Continual Unsupervised Sensorimotor Learn.* 2018, pp. 1–5.

[17] Cansu Sancaktar, Marcel AJ van Gerven, and Pablo Lanillos. “End-to-end pixel-based deep active inference for body perception and action”. In: *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE. 2020, pp. 1–8.

[18] Alexander Tschantz, Anil K Seth, and Christopher L Buckley. “Learning action-oriented models through active inference”. In: *PLoS computational biology* 16.4 (2020), e1007805.

[19] Beren Millidge. “Deep active inference as variational policy gradients”. In: *Journal of Mathematical Psychology* 96 (2020), p. 102348.

[20] Greg Brockman et al. “Openai gym”. In: *arXiv preprint arXiv:1606.01540* (2016).

[21] Kai Ueltzhöffer. “Deep active inference”. In: *Biological cybernetics* 112.6 (2018), pp. 547–573.

[22] Otto van der Himst and Pablo Lanillos. “Deep active inference for partially observable MDPs”. In: *International Workshop on Active Inference*. Springer. 2020, pp. 61–71.

[23] Alexander Tschantz et al. “Reinforcement learning through active inference”. In: *arXiv preprint arXiv:2002.12636* (2020).

[24] Yuval Tassa et al. *dm_control: Software and Tasks for Continuous Control*. 2020. arXiv: 2006.12983 [cs.RO].

[25] Michael Janner et al. “When to trust your model: Model-based policy optimization”. In: *arXiv preprint arXiv:1906.08253* (2019).

[26] Marc Deisenroth and Carl 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.

[27] Kurtland Chua et al. “Deep reinforcement learning in a handful of trials using probabilistic dynamics models”. In: *arXiv preprint arXiv:1805.12114* (2018).

[28] Danijar Hafner et al. “Learning latent dynamics for planning from pixels”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2555–2565.

[29] Mona Buisson-Fenet, Friedrich Solowjow, and Sebastian Trimpe. “Actively learning gaussian process dynamics”. In: *Learning for dynamics and control*. PMLR. 2020, pp. 5–15.

[30] Yi Sun, Faustino Gomez, and Jürgen Schmidhuber. “Planning to be surprised: Optimal bayesian exploration in dynamic environments”. In: *International Conference on Artificial General Intelligence*. Springer. 2011, pp. 41–51.

[31] Jan Storck, Sepp Hochreiter, Jürgen Schmidhuber, et al. “Reinforcement driven information acquisition in non-deterministic environments”. In: *Proceedings of the international conference on artificial neural networks, Paris*. Vol. 2. Citeseer. 1995, pp. 159–164.

[32] Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. “Model-based active exploration”. In: *International conference on machine learning*. PMLR. 2019, pp. 5779–5788.

[33] Jürgen Schmidhuber. *What’s Interesting?* 1997.

[34] Nuttapong Chentanez, Andrew Barto, and Satinder Singh. “Intrinsically motivated reinforcement learning”. In: *Advances in neural information processing systems* 17 (2004).

[35] Deepak Pathak et al. “Curiosity-driven exploration by self-supervised prediction”. In: *International conference on machine learning*. PMLR. 2017, pp. 2778–2787.

[36] Volodymyr Mnih et al. “Asynchronous methods for deep reinforcement learning”. In: *International conference on machine learning*. PMLR. 2016, pp. 1928–1937.

[37] Matthias Schultheis et al. “Receding horizon curiosity”. In: *Conference on Robot Learning*. PMLR. 2020, pp. 1278–1288.

[38] Karl J Friston, Jean Daunizeau, and Stefan J Kiebel. “Reinforcement learning or active inference?” In: *PloS one* 4.7 (2009), e6421.

[39] Thomas Parr and Karl J Friston. “Generalised free energy and active inference”. In: *Biological cybernetics* 113.5 (2019), pp. 495–513.

[40] Zekun Sun and Chaz Firestone. “The dark room problem”. In: *Trends in cognitive sciences* 24.5 (2020), pp. 346–348.

[41] Karl Friston et al. “Perceptions as hypotheses: saccades as experiments”. In: *Frontiers in psychology* 3 (2012), p. 151.

[42] Karl Friston et al. “Active inference and learning”. In: *Neuroscience & Biobehavioral Reviews* 68 (2016), pp. 862–879.

[43] Beren Millidge, Alexander Tschantz, and Christopher L Buckley. “Whence the expected free energy?” In: *Neural Computation* 33.2 (2021), pp. 447–482.

[44] Timothy P Lillicrap et al. “Continuous control with deep reinforcement learning”. In: *arXiv preprint arXiv:1509.02971* (2015).

[45] Scott Fujimoto, Herke Hoof, and David Meger. “Addressing function approximation error in actor-critic methods”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1587–1596.

[46] Volodymyr Mnih et al. “Playing atari with deep reinforcement learning”. In: *arXiv preprint arXiv:1312.5602* (2013).

[47] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[48] Lancelot Da Costa et al. “Active inference on discrete state-spaces: a synthesis”. In: *Journal of Mathematical Psychology* 99 (2020), p. 102447.

[49] Karl Friston et al. “Sophisticated inference”. In: *Neural Computation* 33.3 (2021), pp. 713–763.

[50] David Opitz and Richard Maclin. “Popular ensemble methods: An empirical study”. In: *Journal of artificial intelligence research* 11 (1999), pp. 169–198.

[51] Jouko Lampinen and Aki Vehtari. “Bayesian approach for neural networks—review and case studies”. In: *Neural networks* 14.3 (2001), pp. 257–274.

[52] Laurent Valentin Jospin et al. “Hands-on Bayesian Neural Networks–a Tutorial for Deep Learning Users”. In: *arXiv preprint arXiv:2007.06823* (2020).

[53] Qiang Liu and Dilin Wang. “Stein variational gradient descent: A general purpose bayesian inference algorithm”. In: *arXiv preprint arXiv:1608.04471* (2016).

[54] Pieter-Tjerk De Boer et al. “A tutorial on the cross-entropy method”. In: *Annals of operations research* 134.1 (2005), pp. 19–67.

[55] David McAllester and Karl Stratos. “Formal limitations on the measurement of mutual information”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 875–884.

[56] David Barber Felix Agakov. “The im algorithm: a variational approach to information maximization”. In: *Advances in neural information processing systems* 16.320 (2004), p. 201.

[57] Adam Foster et al. “Variational bayesian optimal experimental design”. In: *arXiv preprint arXiv:1903.05480* (2019).

[58] Mohamed Ishmael Belghazi et al. “Mine: mutual information neural estimation”. In: *arXiv preprint arXiv:1801.04062* (2018).

[59] Ben Poole et al. “On variational bounds of mutual information”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5171–5180.

[60] Monroe D Donsker and SR Srinivasa Varadhan. “Asymptotic evaluation of certain Markov process expectations for large time. IV”. In: *Communications on Pure and Applied Mathematics* 36.2 (1983), pp. 183–212.

[61] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. “Estimating divergence functionals and the likelihood ratio by convex risk minimization”. In: *IEEE Transactions on Information Theory* 56.11 (2010), pp. 5847–5861.

[62] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. “f-gan: Training generative neural samplers using variational divergence minimization”. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2016, pp. 271–279.

[63] Tom Rainforth et al. “On nesting monte carlo estimators”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4267–4276.

[64] Yarin Gal, Rowan McAllister, and Carl Edward Rasmussen. “Improving PILCO with Bayesian neural network dynamics models”. In: *Data-Efficient Machine Learning workshop, ICML*. Vol. 4. 34. 2016, p. 25.

[65] Juan Camilo Gamboa Higuera, David Meger, and Gregory Dudek. “Synthesizing neural network controllers with probabilistic model-based reinforcement learning”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 2538–2544.

[66] Mikael Henaff, William F Whitney, and Yann LeCun. “Model-based planning with discrete and continuous actions”. In: *arXiv preprint arXiv:1705.07177* (2017).

[67] Danijar Hafner et al. “Dream to control: Learning behaviors by latent imagination”. In: *arXiv preprint arXiv:1912.01603* (2019).

[68] Danijar Hafner et al. “Mastering atari with discrete world models”. In: *arXiv preprint arXiv:2010.02193* (2020).

[69] Tuomas Haarnoja et al. “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.

[70] John Schulman et al. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[71] Antonin Raffin et al. *Stable Baselines3*. `https://github.com/DLR-RM/stable-baselines3`. 2019.

[72] Luis Pineda et al. “MBRL-Lib: A Modular Library for Model-based Reinforcement Learning”. In: *Arxiv* (2021). URL: `https://arxiv.org/abs/2104.10159`.

[73] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *nature* 529.7587 (2016), pp. 484–489.

[74] David Silver et al. “Mastering chess and shogi by self-play with a general reinforcement learning algorithm”. In: *arXiv preprint arXiv:1712.01815* (2017).

[75] Timothy Yee, Viliam Lisỳ, and Michael H Bowling. “Monte Carlo Tree Search in Continuous Action Spaces with Execution Uncertainty.” In: *IJCAI*. 2016, pp. 690–697.

[76] Thomas M Moerland et al. “A0c: Alpha zero in continuous action space”. In: *arXiv preprint arXiv:1805.09613* (2018).

[77] Zafeirios Fountas et al. “Deep active inference agents using Monte-Carlo methods”. In: *arXiv preprint arXiv:2006.04176* (2020).

[78] Anuj Karpatne et al. “Physics-guided neural networks (pgnn): An application in lake temperature modeling”. In: *arXiv preprint arXiv:1710.11431* 2 (2017).

[79] Michael Lutter, Christian Ritter, and Jan Peters. “Deep lagrangian networks: Using physics as model prior for deep learning”. In: *arXiv preprint arXiv:1907.04490* (2019).

[80] Mike Lambeta et al. “Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 3838–3845.

[81] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. “Gelsight: High-resolution robot tactile sensors for estimating geometry and force”. In: *Sensors* 17.12 (2017), p. 2762.