Intrinsic Control of Variational Beliefs in Dynamic Partially-Observed Visual Environments

Nicholas Rhinehart
UC Berkeley

Jenny Wang
UC Berkeley

Glen Berseth
UC Berkeley

John D. Co-Reyes
UC Berkeley

Danijar Hafner
University of Toronto

Chelsea Finn
Stanford University

Sergey Levine
UC Berkeley

Abstract

Humans and animals explore their environment and acquire useful skills even in the absence of clear goals, exhibiting intrinsic motivation. The study of intrinsic motivation in artificial agents is concerned with the following question: what is a good general-purpose objective for an agent? We study this question in dynamic partially-observed environments, and argue that a compact and general learning objective is to minimize the entropy of the agent’s state visitation estimated using a latent state-space model. This objective induces an agent to both gather information about its environment, corresponding to reducing uncertainty, and to gain control over its environment, corresponding to reducing the unpredictability of future world states. We instantiate this approach as a deep reinforcement learning agent equipped with a deep variational Bayes filter. We find that our agent learns to discover, represent, and exercise control of dynamic objects in a variety of partially-observed environments sensed with visual observations without extrinsic reward.

1 Introduction

Reinforcement learning offers a framework for learning control policies that maximize a given measure of reward – ideally, rewards that incentivize simple high-level goals, such as survival, accumulating a particular resource, or accomplishing some long-term objective. However, extrinsic rewards may be insufficiently informative to encourage an agent to explore and understand its environment, particularly when the environment is partially-observed: when the agent has a limited view of its environment. A generalist agent should instead acquire an understanding of its environment before a specific objective or reward is provided. This goal motivates the study of self-supervised / unsupervised reinforcement learning: algorithms that provide the agent with an intrinsically-grounded drive to acquire understanding and control of its environment in the absence of an extrinsic reward signal. Agents trained with intrinsic reward signals might accomplish tasks specified via simple and sparse rewards more quickly, or may acquire broadly useful skills that could be adapted to specific task objectives. Our aim is to design an embodied agent and a general-purpose intrinsic reward signal that leads to the agent controlling partially-observed environments when equipped only with a high-dimensional sensor (camera) and no prior knowledge.

A large body of prior methods for self-supervised reinforcement learning focus on attaining coverage, typically through novelty-seeking or skill-discovery objectives; see Hafner et al. [25] for a survey. As argued in prior work [18, 16, 14, 6], a compelling alternative to coverage suited to complex and dynamic environments is to minimize surprise, which incentivizes an agent to control aspects of its environment to achieve homeostasis within it – i.e. constructing and maintaining a niche where it can reliably remain despite external perturbations. We generally expect agents that succeed at minimizing surprise in complex environments to develop similarly complex behaviors; such acquired complex
Figure 1: Top row: The environment consists of a large number of objects, some of which (e.g., the goat) move and act in unpredictable ways, and are not observed unless the agent is nearby. Bottom row: If the agent maintains a latent state space model of the world, it has uncertain beliefs about unobserved objects, particularly those that are dynamic (like the goat). If the agent reduces the long-horizon average entropy of its beliefs, it will first seek out information (e.g., finding the goat), and then modify the environment to limit the range of states the goat can occupy even when it is no longer observed, for example by building a fence around it.

behaviors may be repurposed for other tasks [6]. However, these frameworks do not explicitly address the difficulty of controlling partially-observed environments: if an otherwise complex and chaotic environment contains a “dark room” (small reliable niche), an agent could minimize surprise simply by hiding in this room and refusing to make meaningful observations, thereby failing to explore and control the wider surrounding environment.

Consider Fig. 1, which depicts a partially-observed outdoor environment with various flora (trees, vegetables, and grass), fauna (a goat), weather, and an agent. We will discuss three different intrinsic incentives an agent might adopt in this environment. If the agent’s incentive is to (i) minimize the entropy of its next observation, it will seek the regions with minimal unpredictable variations in flora, fauna, and weather. This is unsatisfying because it merely requires avoidance, rather than interaction. Let us assume the agent will maintain a model of its belief about a learned latent state – the agent cannot observe the true state, instead it learns a state representation. Further, let us assume the agent maintains a separate model of the visitation of its latent state – we will refer to this distribution as its latent visitation. If the agent’s incentive is to (ii) minimize the entropy of belief (either at every step or at some final step), the agent will gather information and take actions to make the environment predictable: find and observe the changes in flora, fauna, and weather that are predictable and avoid those that aren’t. However, once it has taken actions to make the world predictable, this agent is agnostic to future change – it will not resist predictable changes in the environment. Finally, if the agent’s incentive is to (iii) minimize the entropy of its latent visitation, this will result in categorically different behavior: the agent will seek both to make the world predictable by gathering information about it and prevent it from changing. While both the belief and latent visitation entropy minimization objectives are worthwhile intrinsic motivation objectives to study, we speculate that an agent that is adept at preventing its environment from changing will generally learn more complex behaviors.

We present a concise and effective objective for self-supervised reinforcement learning in dynamic partially-observed environments: minimize the entropy of the agent’s latent visitation under a latent state-space model learned from exploration. Our method, which we call Intrinsic Control by Information Capture (IC2), results in an agent that learns to seek out and control factors of variation outside of its immediate observations. We instantiate this framework by simultaneously learning a state-space model as a deep variational Bayes filter along with a policy that employs the model’s beliefs. Our experiments show that our method learns to discover, represent, and control dynamic entities in partially-observed visual environments with no extrinsic reward signal, including in several 3D environments.

2 Maxwell’s Demon and Belief Entropy

The main concept behind our approach to self-supervised reinforcement learning is that incentivizing an agent to minimize the entropy of its beliefs about the world is sufficient to lead it to both gather information about the world and learn to control aspects of its world. Our approach is partly inspired by a well-known connection between information theory and thermodynamics, which can be illustrated informally by a version of the Maxwell’s Demon thought experiment [47, 42]. Imagine a container
We denote the undiscounted finite-horizon state visitation as \( d(s) \). The observed “TwoRoom” environment, depicted in Fig. 3. The environment has two rooms: an empty room on the left, and a “busy” room on the right, the latter containing two moving particles. The main question that we tackle in the design of our algorithm is: how can we formulate a general control over its observations. We observe this phenomenon in our experiments. By sorting the particles onto one side, the demon appears to reduce the disorder of the system, as measured by the thermodynamic entropy, \( S \), which increases the energy, \( F \) available to do work, as per Helmholtz’s free energy relationship, \( F = U - TS \). The ability to do work affords the agent control over the environment. This apparent violation of the second law of thermodynamics is resolved by accounting for the agent’s information processing needed to make decisions [5]. By concentrating the particles into a smaller region, the number of states each particle visits is reduced. This illustrates an example environment in which reducing the entropy of the visitation distribution results in an agent gaining the ability to do work. In the same way that Maxwell’s demon accumulates free energy via information-gathering and manipulating its environment, we expect self-supervised agents guided by belief visitation entropy minimization to accumulate the equivalent of potential energy in their corresponding sequential decision processes, which leads them to gain control over their environments.

3 Preliminaries

Our goal in this work will be to design self-supervised reinforcement learning methods in partially observed settings to acquire complex behaviors that both gather information and gain control over their environment. To this end, we will formulate the learning problem in the context of a discrete-time partially-observed controlled Markov process, also known as a controlled hidden Markov process (CHMP), which corresponds to a POMDP without a reward function. The CHMP is defined by a state space \( S \) with states \( s \in S \), action space \( A \) with actions \( a \in A \), transition dynamics \( P(s_{t+1}|s_t, a_t) \), observation space \( \Omega \) with observations \( o \in \Omega \), and emission distribution \( O(o_t|s_t) \). The agent is a policy \( \pi(a_t|o_{\leq t}) \). Note that it does not observe any states \( s \).

We denote the undiscounted finite-horizon state visitation as \( d^\pi(s) \) as \( \frac{1}{T} \sum_{t=0}^{T-1} \text{Pr}_\pi(s_t = s) \), where \( \text{Pr}_\pi(s_t = s) \) is the probability that \( s_t = s \) after executing \( \pi \) for \( t \) steps. Using \( d^\pi(s) \), we can quantify the average disorder of the environment with the Shannon entropy, \( H(d^\pi(s)) \). Prior work proposes surprise minimization \( \text{min}_s - \log \hat{p}(s) \) as an intrinsic control objective [15, 71, 6]; in Berseth et al. [6] (SMiRL), the agent models the state visitation distribution, \( d^\pi(s) \) with \( \hat{p}(s) \), which it computes by assuming access to \( s \). In environments in which there are natural sources of variation outside of the agent, this incentivizes the SMiRL agent, fully aware of these variations observed through \( s \), to take action to control them. In a partially-observed setting, it is a matter of interpretation of what the most natural extension of SMiRL is – whether (1) the observation should be treated as the state, or (2) an LSSM’s belief should be treated as an estimate of the latent state. Thus, our experiments include both interpretations, the latter constituting one of our methods. In the former, SMiRL’s model becomes \( \tilde{p}^\pi(o) \), which generally will enable the agent to ignore variations that it can prevent from appearing in its observations. We observe this phenomenon in our experiments.

4 Control and Information Gathering via Belief Entropy Minimization

The main question that we tackle in the design of our algorithm is: how can we formulate a general and concise objective function that can enable an RL agent to gain control over its partially-observed environment, in the absence of any user-provided task reward? Consider the following partially-observed “TwoRoom” environment, depicted in Fig. 3. The environment has two rooms: an empty (“dark”) room on the left, and a “busy” room on the right, the latter containing two moving particles.

\[1\] In Maxwell’s original example, the demon sorts the particles into the two chambers based on velocity. Our example is closely related to Szilard’s engine [69, 46].
We defer the description of the LSSM learning and architecture to Section 5, and now motivate how we will use the LSSM for constructing an intrinsic control objective.

Expected model information gain

To this end, we learn a separate exploratory policy to maximize variability in the environment (latent state) from the true variability (true state). At timestep $t$, the exploration policy is employed as a way to encounter dynamic objects and collect data on which the LSSM that represents all of the underlying factors of variation.

In contrast to Schmidhuber [61], Houthooft et al. [27], Gheshlaghi Azar et al. [20], Sekar et al. [63]. The approach involves learning a latent state-space model (LSSM) [73, 39, 45, 22, 48, 74, 72, 41, 10, 23, 49, 59]. We intermittently refer to these dynamic components as “factors of variation” to distinguish the model’s representation of variability in the environment (latent state) from the true variability (true state). At timestep $t$, the LSSM represents the agent’s current belief as $q_\phi(z_t | o_{\leq t}, a_{\leq t-1})$, where $z_t$ is the model’s latent state. We defer the description of the LSSM learning and architecture to Section 5, and now motivate how we will use the LSSM for constructing an intrinsic control objective.

Representing variability with latent state-space models. In order for the agent to represent the dynamic components of the environment observed from images, our method involves learning a latent state-space model (LSSM) [73, 39, 45, 22, 48, 74, 72, 41, 10, 23, 49, 59]. We intermittently refer to these dynamic components as “factors of variation” to distinguish the model’s representation of variability in the environment (latent state) from the true variability (true state). At timestep $t$, the LSSM represents the agent’s current belief as $q_\phi(z_t | o_{\leq t}, a_{\leq t-1})$, where $z_t$ is the model’s latent state. We defer the description of the LSSM learning and architecture to Section 5, and now motivate how we will use the LSSM for constructing an intrinsic control objective.

Belief entropy and latent visitation entropy. Consider a policy that takes actions to minimize the entropy of the belief $H(q_\phi(z_t | o_{\leq t}, a_{\leq t-1}))$. This corresponds to the agent performing active state estimation [12, 75, 38], and is equivalent to taking actions to maximize expected latent-state information gain $I(o_t, z_t | o_{\leq t}, a_t)$ [2]. However, active state estimation is satisfied by a policy that simply collects informative observations, as it does not further incentivize actions to “stabilize” the environment by preventing the latent state from changing. Analogous to the standard definition of undiscounted state visitation, consider the undiscounted latent visitation: $d^\pi(z) = 1/T \sum_{t=1}^{T-1} \Pr_\pi(z_t = z)$, where $\Pr_\pi(z_t = z) = \mathbb{E}_{q_\phi}(z_t | o_{\leq t}, a_{\leq t-1})$ (the expected belief after executing $\pi$ for $t$ timesteps). Our goal is to minimize $H(d^\pi(z))$, as this corresponds to stabilizing the agent’s beliefs, which incentivizes both reducing uncertainty in each $q_\phi(z_t | o_{\leq t}, a_{\leq t-1})$, as well as constructing a niche such that each $q_\phi(z_t | o_{\leq t}, a_{\leq t-1})$ concentrates probability on the same latent states.

Discovering factors of variation. In order for belief entropy minimization to incentivize the agent to control entities in the environment, the LSSM’s belief must represent the underlying state variables in some way and model their uncertain evolution until either observed or controlled. For example, the demon in the thought experiment in Section 2 would have no incentive to gather the particles if it did not know that they existed. If the agent is unable to encounter or represent dynamic objects in its LSSM, it could choose not to move. However, if the LSSM properly captures the existence and motion of dynamic objects (even out of view), then the agent has an incentive to exert control over them, rather than avoid them by going to parts of the environment where it does not encounter them. While sufficient random exploration may result in a good-enough LSSM, making this approach generally practical requires a suitable exploration strategy to collect the experience necessary to train an LSSM that represents all of the underlying factors of variation.

To this end, we learn a separate exploratory policy to maximize expected model information gain, similar to Schmidhuber [61], Houthooft et al. [27], Gheshlaghi Azar et al. [20], Sekar et al. [63]. The exploration policy is employed as a way to encounter dynamic objects and collect data on which the...
model performs poorly. The exploration policy does not need to learn accurate estimates of the latent state and its uncertainty -- it is merely incentivized to visit where the current model is inaccurate. Expected information gain about model parameters $\theta$ is relative to a set of prior experience $D$ and a partial trajectory $h_{1:t}$, given as $I(o_t;\theta|h_{1:t}, D) = \mathbb{E}_{o_t}KL(p(\theta|o_t, h_{1:t}, D) \| p(\theta|h_{1:t}, D))$. Note that model information gain is distinct from the information an agent may gather to reduce its belief entropy $H(q_\phi(z_{1:t}|o_{\leq t}, a_{\leq t-1}))$ within the current episode. Computing the full model parameter prior $p(\theta|h_{1:t}, D)$ and posterior $p(\theta|o_t, h_{1:t}, D)$ is generally computationally expensive, and also requires evaluating an expectation over observations -- instead, we approximate this expected information gain following a method similar to Sekar et al. [63]: we use an ensemble of latent dynamics models, $E = \{p_{\theta_i}(z_{1:T}|o_{1:T}, a_{1:T})\}_{i=1}^K$ to compute the variance of latent states estimated by $q_\phi(z_{1:t}|o_{\leq t}, a_{\leq t-1})$. We build the ensemble throughout training using the method of Izmailov et al. [28]. Thus, the exploration reward is given as: $r_e = \text{Var}(\theta_i)|\log p_\theta(z_{t-1}, a_{t-1})|z_t \sim q_\phi$.

5 The IC2 Algorithm

Now we describe how we implement our intrinsic control algorithm. The main components are the latent-state space model, the latent visitation model, and the exploration and control policies.

Latent state-space model. In our CHMP setting, the agent only has access to partial observations $o$ of the true state $s$. In order to estimate a representation of states and beliefs, we employ a sequence-based Variational Autoencoder to learn latent variable belief, dynamics, and emission models. We formulate the variational posterior to be $q_\phi(z_{1:T}|o_{1:T}, a_{1:T}) = \prod_{t=1}^T q_\phi(z_t|o_{\leq t}, a_{\leq t-1})$ and the generative model to be $p(z_{1:T}|o_{1:T}, a_{1:T}) = \prod_{t=1}^T p_\theta(o_t|z_t)p_\theta(z_t|z_{t-1}, a_{t-1})$. Denoting $h_t \equiv (o_{\leq t}, a_{\leq t-1})$, the log-evidence of the model and its lower-bound are:

$$
\begin{align*}
\log p(o_{1:T}|a_{1:T-1}) &= \log \mathbb{E}_{z_{1:T}\sim p(z_{1:T}|a_{1:T})} \prod_{t=1}^T p(o_t|z_t) \\
&= \sum_{t=1}^T \mathbb{E}_{q_\phi(z_t|h_t)} \left[p_\theta(o_t|z_t)\right] - \mathbb{E}_{q_\phi(z_{t-1}|h_{t-1})} \left[\text{KL}(q_\phi(z_t|h_t) \| p_\theta(z_t|z_{t-1}, a_{t-1}))\right].
\end{align*}
$$

Given a dataset $D = \{(o_{1:T}, a_{1:T})_i\}_{i=1}^N$, Eq. 1 is used to train the model via $\max_{\phi, \theta} \frac{1}{|D|} \sum_{i=1}^{|D|} \mathcal{L}(\phi, \theta; o_{1:T}, a_{1:T})$. The focus of our work is not to further develop the performance of LSSMs, but improvements in the particular LSSM employed would translate to improvements in our method. In practice, we implemented an LSSM in PyTorch [56] similar to the categorical LSSM architecture described in [24]. In this case, both the belief prior and belief posterior are distributions formed from products of $K_1$ categorical distributions over $K_2$ categories:

$$
g(z; v) = \prod_{k_1=1}^{K_1} \prod_{k_2=1}^{K_2} V_{k_1,k_2} \tilde{v}_{k_1,k_2},$$

where the vector of $K_1 \cdot K_2$ parameters, $v$, is predicted by neural networks: $v_{\text{prior}} = f_\phi(o_{\leq t}, a_t)$ for the posterior, and $v_{\text{prior}} = f_\theta(o_{\leq t}, a_t)$ for the prior. We found it effective to construct $f_\phi$ and $f_\theta$ following the distribution-sharing decomposition described in [30, 31, 10], in which the posterior is produced by transforming the parameters of the prior with a measurement update. We implemented the prior as an RNN, and the posterior as an MLP, and defer further details to the supplementary material.
We now describe several rewards that can be constructed with the LSSM, and how they connect to Algorithm 1.

**Algorithm summary.** The entropy of the current belief measures the agent’s certainty of the latent state, and byCertainty.

The latent state information gain (not the model information gain in Section 4) is:

\[ r_{\text{infogain}}^t = -\sum_{z_t} \pi_t(z_t) \log \bar{q}_t(z_t) - \sum_{z_t} \pi_t(z_t) \log \bar{q}_t(z_t) \]  

The expected negative surprise of the current belief under the latent visitation of implementation, although prior work does so in the case of a single policy [41]. Our method is represented by the LSSM. We do not back-propagate the policies’ losses to the LSSM for simplicity of implementation, although prior work does so in the case of a single policy [41]. Our method is agnostic to the subroutine used to improve the policies. In our implementation, we employ PPO [62].

**Algorithm 1 Intrinsic Control via Information Capture (IC2)**

```plaintext
1: procedure IC2(Env, K, M, N, L)
2: Initialize \( \pi_e, \pi_c, q_0, p_0(\theta) \) \( \forall \theta \), \( D \leftarrow \emptyset \).
3: for episode = 0, ..., M do
4: \( D_e \leftarrow \text{Collect}(N, \text{Env}, \pi_e); D_c \leftarrow \text{Collect}(N, \text{Env}, \pi_c); \)
5: \( D \leftarrow D \cup D_e \cup D_c \)
6: Update \( q_0, p_0 \) using SGD on Eq. 1 with data \( D \) for \( L \) rounds.
7: Update \( \pi_c \) using SGD on PPO’s objective with rewards \( r_c \) with data \( D_c \)
8: Update \( \pi_e \) using SGD on PPO’s objective with rewards defined by Eq. (2) with data \( D_c \).
```

**Latent visitation model.** Our agent cannot, in general, evaluate \( d^x(z) \) or \( H(d^x(z)) \); at best, it can approximate them. We do so by maintaining a within-episode estimate of \( d^x(z) \) by constructing a mixture across the belief history samples \( \bar{q}_t(z) = \frac{1}{L} \sum_{t=0}^{L} q_0(z_t | o_{\leq t}, a_{\leq t-1}) \). This corresponds to a mixture (across time) of single-sample estimates of each \( \mathbb{E}_n q_0(z_t | o_{\leq t}, a_{\leq t-1}) \). Given \( \bar{q}_t(z) \), we have an estimate of the visitation of the policy, and we can use this as part or all of the reward signal of the agent. To implement \( \bar{q}_t(z) \), we found that simply recording each belief sufficed.

**5.1 Belief-based objectives**

We now describe several rewards that can be constructed with the LSSM, and how they connect to our main objective. In what follows, we denote \( q_0(z_t | o_{\leq t}, a_{\leq t-1}) = q_t(z_t) \) for brevity. While our primary objective is “niche expansion,” we also explore other intrinsic objectives based on the LSSM and the latent visitation model. These rewards are visualized in Fig. 4.

**Niche Expansion.** Our primary objective is the KL from the belief to the belief visitation, which we term “niche expansion”. This incentivizes the agent to reduce belief visitation entropy by bringing the current belief towards the current latent visitation distribution, and to expand current belief to encourage exploration, and thus potentially find a broader niche. The niche expansion reward is:

\[ r_{\text{nic}}^t \equiv -\mathbb{KL}(q_t(z_t) || \bar{q}_t(z)) = \mathbb{E}_{q_t(z_t)}[\log \bar{q}_t(z) - \log q_t(z_t)]. \]  

**Niche Creation.** The expected negative surprise of the current belief under the latent visitation distribution measures how unsurprising the current belief is relative to the agent’s experience in the episode so far. One interpretation of this reward is that it is SMiRL [6] applied to samples of latent beliefs estimated from a LSSM in a CHMP. Minimizing this reduces the number of visited environment states and increases the agent’s certainty. The niche creation reward is:

\[ r_{\text{nic}}^t \equiv -H(q_t(z_t)) = \mathbb{E}_{q_t(z_t)}[\log \bar{q}_t(z_t)]. \]  

**Certainty.** The entropy of the current belief measures the agent’s certainty of the latent state, and by extension, about the aspects of the environment captured by the latent. Minimizing the belief entropy increases the agent’s certainty. It is agnostic to predictable changes, and penalizes unpredictable changes. We define the certainty reward as the negative belief entropy:

\[ r_c^t \equiv -H(q_t(z_t)) = \mathbb{E}_{q_t(z_t)}[\log q_t(z_t)] \]  

**State Infogain.** The latent state information gain (not the model information gain in Section 4) measures how much more certain the belief is compared to its temporal prior. Gaining information does not always coincide with being certain, because an infogain agent may cause chaotic events in the environment with outcomes that it only understands partially. As a result, it has gained more information than standing still but has also become less certain. We define the infogain reward as:

\[ r_s^t \equiv \mathbb{E}_{q_{t-1}(z_{t-1})}\mathbb{KL}(q_t(z_t) || p(z_t | z_{t-1}, a_{t-1})) = \mathbb{E}_{q_t(z_t)} q_{t-1}(z_{t-1})[\log q_t(z_t)/p(z_t | z_{t-1}, a_{t-1})]. \]  

**5.2 Algorithm summary**

Conceptual pseudocode for Intrinsic Control via Information Capture (IC2) is presented in Alg. 1. The algorithm begins by initializing the LSSM, \( q_0 \) and \( p_0 \), as well as two separate policies: one trained to collect difficult-to-predict data with the exploration objective with rewards defined by \( r_c \), \( \pi_e \), and one trained to maximize a intrinsic control objectives defined by one of Eqs. (2) to (5).

We represent each policy \( \pi(a_t | v_{\text{posterior}, t}) \) as a two-layer fully-connected MLP with 128 units. Recall that \( v_{\text{posterior}} \) is the vector of posterior parameters, which enables the policy to use the memory represented by the LSSM. We do not back-propagate the policies’ losses to the LSSM for simplicity of implementation, although prior work does so in the case of a single policy [41]. Our method is agnostic to the subroutine used to improve the policies. In our implementation, we employ PPO [62].
6 Experiments

Our experiments are designed to answer the following questions: **Q1: Intrinsic control capability**: Does our latent visitation-based self-supervised reward signal cause the agent to stabilize partially-observed visual environments with dynamic entities more effectively than prior self-supervised stabilization objectives? **Q2: Properties of IC2 objectives**: What types of emergent behaviors does each belief-based objective described in Section 5.1 evoke?

In order to answer these questions, we identified environments with the following properties (i): partial observability, (ii): dynamic entities that the agent can affect, and (iii): high-dimensional observations. Because many standard RL benchmarks do not contain the significant partial-observability that is prevalent in the real world, it is challenging to answer these questions with them. Instead, we create several environments, and employ several existing environments we identified to have these properties. In what follows, we give an overview of the experimental settings and conclusions. We defer comprehensive details to the supplementary material.

6.1 Environments

**TwoRoom.** As previously described, this environment has two rooms: an empty (“dark”) room on the left, and a “busy” room on the right, the latter containing moving particles (colored dots) that move unless the agent “tags” them, which permanently stops their motion, as shown in Fig. 5. The agent (white dot) can observe a small area around it (grey square), which it receives as an image. In this environment, control corresponds to finding and stopping the particles. The action space is \( A = \{ \text{left, right, up, down, tag, no-op} \} \), and the observation space is normalized RGB-images: \( \Omega = [0, 1]^{30 \times 30} \). An agent that has significant control over this environment should tag particles to reduce the uncertainty over future states. To evaluate policies, we use the average fraction of total particles locked, the average fraction of particles visible, and the discrete true-state visitation entropy of the positions of the particles, \( H(d^T(s_d)) \). We employed two versions of this environment, with details provided in the supplementary material. In the large environment, the agent observes a 5x5 area inside a 15x15 area, and the busy room contains 5 particles.

**VizDoom DefendTheCenter.** The VizDoom DefendTheCenter environment shown in Fig. 6 is a circular arena in which a stationary agent, equipped with a partial field-of-view of the arena and a weapon, can rotate and shoot encroaching monsters \([32]\). The action space is \( A = \{ \text{turn left, turn right, shoot} \} \), and the observation space is normalized RGB-images: \( \Omega = [0, 1]^{64 \times 64} \). In this environment, control corresponds to reducing the number of monsters by finding and shooting them. We use the average original environment return, (which no policy has access to during training), the average number of killed monsters at the end of an episode, and the average number of visible monsters to measure the agent’s control.

**OneRoomCapture3D** The MiniWorld framework is a customizable 3D environment simulator in which an agent perceives the world through a perspective camera \([9]\). We used this framework to build the environment in Fig. 7 which an agent and a bouncing box both inhabit a large room; the agent can lock the box to stop it from moving if it is nearby, as well as constrain the motion of the box by standing nearby it. In this environment, control corresponds to finding the box and either trapping it near a wall, or tagging it. The action space is \( A = \{ \text{turn left } 20^\circ, \text{turn right } 20^\circ, \text{move forward, move backward, tag} \} \), and observation space is normalized RGB-images: \( \Omega = [0, 1]^{64 \times 64} \). We use the average fraction of time the box is captured, the average time the box is visible, and differential entropy (Gaussian-estimated) of the true-state visitation of the box’s position \( H(d^T(s_d)) \) to measure the agent’s ability to control the environment.

6.2 Comparisons

In order to answer **Q1**, we compare to SMiRL \([6]\) and a recent empowerment method \([78]\) that estimates the current empowerment of the agent from an image. We use the empowerment estimate...
as a reward signal for training a policy. In order to answer Q2, we ensured each environment was instrumented with the aforementioned metrics of visibility and control, and deployed Algorithm 1 separately with each of Eqs. (2) to (5), as well as a simple sum of Eq. (3) and Eq. (5). We also compare to a “random policy” that chooses actions by sampling from a uniform distribution over the action space, as well as an “oracle policy” that has access to privileged information about the environment state in each environment. We perform evaluation at 5e6 environment steps with 50 policy rollouts per random seed, with 3 random seeds for each method (150 rollouts total). 

Exploration: Note that while our method employs an exploration policy, it is used only to improve the LSSM – the data collected by the exploration policy is not used by the on-policy learning of the control policy (although it could be used, in principle, by an off-policy learning procedure). No method, including ours, is provided with an explicit separate mechanism to augment data collection for policy optimization. All control policies employed in our methods and the baselines follow the same exploration procedure that occurs as part of PPO optimization.

Oracle policies. The oracle policies are designed to maximize the control performance in each environment as much as practically possible to put the control performance of our intrinsic objectives in context. Their purpose is to approximate the highest possible performance in each environment, for which access to privileged information can be convenient. In the TwoRoom and OneRoomCapture3D environments, the oracle policies are scripted, using access to the environment state, to seek out the nearest unfrozen object and freeze it, thereby achieving high performance on the lock, capture, and entropy control metrics. In the DefendTheCenter environment, the oracle policy is a neural network trained using PPO to shoot as many monsters as possible by accessing the true extrinsic task reward of the environment.

Results summary. Our primary results are presented in Fig. 8. We observe the Niche Expansion (NE) and Niche Creation+Infogain (NCI) rewards to yield policies that exhibit a high degree of control over each environment – finding and stopping the moving objects, and finding and shooting
the monsters, in the complete absence of any extrinsic reward signal. The results show that employing the NC objective accompanied by a within-episode bonus term – resulting in either the NE objective or the NCI objective – achieves the best control performance across the environments tested. The Infogain-based agent generally seeks out and observes, but does not interfere with, the dynamic objects; the high Visibility metric and low control metrics (Lock, Capture, Kill) in each environment illustrate this. Qualitative results of this phenomenon are illustrated in Fig. 9, in which the infogain signal is high when stochastically-moving monsters are visible, and low when they are not. We observe the method of Berseth et al. [6] tends to learn policies that hide from the dynamic objects in the environment, as indicated by the low control and visibility values of the final policy. Furthermore, we observe the method of Zhao et al. [78] not to exhibit controlling behavior in these partially-observed environments, instead it views the dynamic objects in the TwoRoom and OneRoomCapture3D Environments somewhat more frequently than the random policy. We present videos of all policies in the supplementary material.

Detailed DefendTheCenter results. We observe the infogain-based agent to rotate the camera until a close monster is visible and then oscillate the camera centered on the monster. The infogain leads to the agent maintaining a large visual stimulus that is difficult for the model to predict due to the stochastic motion of the monster, an interpretation supported by Fig. 8k, where Infogain and NCI agents have the highest number of visible monsters in their field of view. We also observe that the NCI policy exhibits similar behavior to Infogain, which we speculate may be due to the infogain component of the objective dominating the NC component. We observe the NC agent to minimize visibility of the monsters in the following way: the agent tends to shoot the immediately visible monsters upon first spawning and then toggle between turning left and right. Since there is no no-op action, this allows the agent to see as few monsters as possible, since its view is centered on a region of the map it just cleared. We observe the NE agent to typically rotate in one direction and reliably shoot monsters. Our interpretation of this is that the NE reward is causing the agent to seek out and reduce the number of surprising monsters more often than the NC agent due to the belief entropy bonus term that NE contains, which encourages more exploration compared to NC. The Certainty agent has similar behavior to the NC agent but we observe that it is not as successful and has fewer monster kills and more visible monsters than NC.

Detailed OneRoomCapture3D results. Figs. 8b and 8f show that the NE scores lower (better) than NCI in the box entropy metric yet slightly worse in the Captured metric due to (1) differences in how each agent tends to capture the box and (2) higher variance in the NCI policy optimization. (1) The NE agent tends to move towards the box and execute the “freeze” action to stop the box from moving, whereas the NCI agent tends to move toward the box and trap it next to the wall without executing the freeze action, which causes the box to bounce back and forth between the wall and the agent every timestep, which is measured as Captured. We hypothesize NCI does this because allowing the box to retain some movement gives the agent more Infogain reward. (2) Higher variance in the policy optimization of the NCI agent resulted in some runs that performed worse, which impacted both the entropy and capture metrics, but had a stronger effect on the box capturing metric.
7 Related Work

Much of the previous work on learning without extrinsic rewards has been based either on (i) exploration [8, 54, 53], or (ii) some notion of intrinsic control, such as empowerment [35, 50, 31]. Exploration approaches include those that maximize model prediction error or improvement [60, 44, 66, 57], maximize model uncertainty [27, 67, 65, 58, 20], maximize state visitation [4, 19, 70, 26], maximize surprise [60, 1, 68], and employ other novelty-based exploration bonuses [43, 7, 33, 34]. Our method can be combined with prior exploration techniques to aid in optimizing our proposed objective, and in that sense our work is largely orthogonal to prior exploration methods.

Prior works on intrinsic control include empowerment maximization [35, 36, 51], observational surprise minimization [15, 17, 71, 6, 55], and skill discovery [3, 37, 21, 11, 64, 76]. Observational surprise minimization seeks policies that make observations predictable and controllable, and is closely connected to entropy minimization, as entropy is defined to be the expected surprise. In Friston et al. [18], the notion of Free Energy Minimization corresponds to minimizing observational entropy, and states that the entropy of hidden states in the environment is bounded by the entropy of sensory observations. However, the proof assumes a diffeomorphism to hold between states and observations, which is explicitly violated in CHMPs and any real-world setting, as agents cannot perceive the state of anything outside their egocentric sensory observations. Similarly, empowerment [36, 29, 31, 78] is a measure of the degree of control an agent could have over future observations, whereas state visitation entropy is a measure of the degree of the control an agent has over the underlying environment state. Our approach seeks to infer and gain control over a representation of the environment’s state, as opposed to the agent’s observations. We demonstrate environments where minimizing observational surprise and maximizing empowerment leads to degenerate solutions that ignore important factors of variation, whereas our approach identifies and controls them. Representation learning methods have been explored in a variety of prior work, including, but not limited to, [40, 73, 30, 13, 52, 77, 22, 41]. Our approach employs a representation learning method to build a latent state-space model [73, 39, 30, 22, 48, 74, 72, 41, 10, 23, 49, 59].

8 Discussion

We presented IC2, a method for intrinsically motivating an agent to discover, represent, and exercise control of dynamic objects in a partially-observed environments sensed with visual observations. We found that our method approached expert-level performance on several environments and substantially surpassed prior work in its unsupervised control capability. While our experiments represent a proof-of-concept that illustrates how latent state belief entropy minimization can incentivize an agent to both gather information and gain control over its environment, there are a number of exciting future directions. First, our method is inspired by a connection between thermodynamics and information theory, but the treatment of this connection is informal. Formalizing this connection could lead to an improved theoretical understanding of how IC2 and other intrinsic motivation methods can lead to desirable behavior, and perhaps allow deriving conditions on environments under which such desirable behaviors would emerge; such a characterization might invoke the fact that the Certainty reward degenerates to activate state estimation unless the environment contain sources of uncertainty outside of the robot’s state. Second, a precise characterization of environments in which it may not be desirable to deploy IC2 would be useful, for example, (i) if the Niche-based approaches were deployed simultaneously on multiple robots in the same environment, each robot’s goal would likely learn to trap or disable the other robots, in order to collapse its uncertainty about them, which may be undesirable, or (ii) if the Niche-based approaches were deployed without safeguards in an environment with living things, similarly, the agent may learn to trap or destroy them, unless the agent estimated it could better avoid future uncertainty by cooperating with them. Finally, since IC2 relies on the LSSM to estimate beliefs, it is limited by the capacity of LSSMs to properly model beliefs – an exciting future direction and an active area of research is extending the capabilities of LSSMs, which would enable study of the behavior that emerges when IC2 is deployed in more complex environments.

Acknowledgements This research was supported by ARL DCIST CRA W911NF-17-2-0181, the Office of Naval Research, and an NSERC Vanier Canada Graduate Scholarship.
References


