Surprise-Adaptive Intrinsic Motivation for Unsupervised Reinforcement Learning

Adriana Hugessen* **Roger Creus Castanyer*** **Faisal Mohamed*** **Glen Berseth**
Université de Montréal and Mila Quebec AI Institute
{adriana.knatchbull-hugessen,roger.creus-castanyer,faisal.mohamed}@mila.quebec

Abstract

Both entropy-minimizing and entropy-maximizing objectives for unsupervised reinforcement learning (RL) have been shown to be effective in different environments, depending on the environment's level of natural entropy. However, neither method alone results in an agent that will consistently learn intelligent behavior across environments. In an effort to find a single entropy-based method that will encourage emergent behaviors in any environment, we propose an agent that can adapt its objective online, depending on the entropy conditions it faces in the environment, by framing the choice as a multi-armed bandit problem. We devise a novel intrinsic feedback signal for the bandit, which captures the agent's ability to control the entropy in its environment. We demonstrate that such agents can learn to optimize task returns through entropy control alone in didactic environments for both high- and low-entropy regimes and learn skillful behaviors in certain benchmark tasks. Videos and summarized findings can be found on our project webpage.

1 Introduction

Unsupervised reinforcement learning (URL), or learning without access to an extrinsic reward function, has recently gained significant attention, often as a pretraining method (Jaderberg et al., 2017) or as a reward bonus in sparse reward domains (Schmidhuber, 1991; Pathak et al., 2017; Burda et al., 2019b). A recent focus has been on developing objectives where the agent has no access to extrinsic rewards and instead develops emergent behaviors from intrinsic motivation alone (Lopes et al., 2012; Kim et al., 2020; Berseth et al., 2021). In this context, unsupervised RL holds the promise of being able to develop natural-like intelligence, i.e. generally-capable agents that can be deployed to solve diverse tasks across diverse environments. However, thus far, no single intrinsic motivation function has succeeded in capturing the complexity of motivation that gives rise to intelligent systems.

Interestingly, two seemingly opposing methods, surprise-minimization (Friston, 2010; Berseth et al., 2021) and surprise-maximization (Schmidhuber, 1991; Pathak et al., 2017; Hazan et al., 2019; Tiapkin et al., 2023), have been proposed as intrinsic motivations, with both methods performing well depending on the properties of the environment in which they are deployed. In general, surprise-minimizing methods (Berseth et al., 2021) perform well in environments with naturally high entropy that can be reduced through control, while curiosity-based methods (Pathak et al., 2017) are better suited to environments where explicit exploration is necessary to encounter novel information. However, both methods are known to possess failure modes when exposed to the opposite entropy regime (Schmidhuber, 2010; Sun & Firestone, 2020).

In this work, we propose an adaptive mechanism to select between maximizing and minimizing surprise in a given environment, based on the agent's ability to exert control over its entropy conditions, which we frame as a multi-armed bandit problem. We experimentally validate our *surprise-adaptive* agent by demonstrating its ability to mirror a surprise-maximizing or -minimizing agent in didactic

*Equal contribution.

low- and high-entropy environments, respectively, and, in doing so, perform well on these tasks without any access to extrinsic task rewards. In benchmark environments, we demonstrate more diverse emergent behaviors, as measured by the performance on extrinsic task reward, than observed from the single-objective agents.

2 Related work

There is a rich body of work in the field of unsupervised RL and intrinsic motivation, upon which our method builds. The most widely explored class of intrinsic objectives is related to improving state coverage through exploration bonuses which reward some measure of novelty. In low-dimensional settings, count-based methods (Auer, 2002; Bellemare et al., 2016; Machado et al., 2020) are simple and effective but do not always extend well to higher dimensions (Lobel et al., 2023). Another popular class of methods in high-dimensional settings uses prediction error as an exploration bonus (Schmidhuber, 1991; Pathak et al., 2017). A conceptually similar idea is that of entropy maximization (Hazan et al., 2019; Tiapkin et al., 2023; Jain et al., 2023), which seeks to maximize the entropy of the distribution of states experienced by the agent throughout its lifetime. Naive implementations of these novelty-seeking agents, however, can be susceptible to what is known as the "noisy-TV problem" (Schmidhuber, 2010), where the agent becomes transfixed by irreducible aleatoric noise in the environment. Various formulations have been developed to combat this issue, though issues often persist (Houthooft et al., 2016; Pathak et al., 2017; Burda et al., 2019b). Though these methods are generally implemented as bonuses to the extrinsic reward, some works have also investigated the ability of curiosity-driven agents to achieve good task rewards without any access to the extrinsic reward (Burda et al., 2019a)

An alternative class of intrinsic objectives also targets the scenario where no extrinsic rewards are available by incentivizing the agent to exert control over its environment. This class of methods is rooted in the free energy principle, a concept from neuroscience that posits that intelligent organisms seek out stable niches by learning to control their environment to minimize the entropy they experience over their lifetime (Friston, 2010). One prominent formulation in this class is that of empowerment, defined as the maximal mutual information between an agent's actions and future states (Klyubin et al., 2005; Karl et al., 2015; Zhao et al., 2020). However, empowerment is computationally difficult to estimate in large or continuous state and action spaces. A more tractable approximation to the free-energy principle was proposed by Berseth et al. (2021) as surprise-minimization. In this formulation, an upper bound on an agent's total trajectory entropy is minimized by rewarding the agent with the log-probability of the current state under the estimated state marginal distribution. This method has shown promising results in a diverse set of stochastic environments (Berseth et al., 2021; Rhinehart et al., 2021). Surprise-minimizing agents, however, can fall victim to the "dark room problem" (Sun & Firestone, 2020), where the agent discovers an area of the state-space without any stochastic dynamics and fails to seek out any additional experience.

Two recent works make efforts towards combining surprise-minimization and maximization objectives to avoid the degenerate cases of prior methods, either using a complex multi-agent paradigm (Fickinger et al., 2021) or learned skills (Zhao et al., 2022). In Fickinger et al. (2021) the authors seek to capture more complex behaviors by alternately minimizing and maximizing surprise in an adversarial game between surprise-minimizing and surprise-maximizing players. However, this adversarial approach is susceptible to unstable training dynamics. In Zhao et al. (2022), they circumvent the complexity of adversarial RL, instead training a single agent equipped with two different skill sets, one surprise-minimizing, and the other surprise-maximizing. This approach is conceptually and practically simpler than the multi-agent approach. However, neither method uses an adaptive mechanism to control the objective, instead using fixed-length windows to alternate between objectives. In contrast, our proposed method can adapt to entropy conditions online to bias the agent towards the objective with the greatest potential.

Prior works have explored adaptivity in RL and found that it can be beneficial for learning (Badia et al., 2020; Moskovitz et al., 2021). Similar to our work, Moskovitz et al. (2021) uses a multi-armed

bandit to control a learning hyperparameter. However, their method relies on extrinsic rewards for providing feedback to the bandit, while our method derives rewards based only on intrinsic signals.

3 Background

Reinforcement learning. RL is a learning paradigm for sequential decision-making problems. In RL, an agent acts in an environment from which it receives observations and rewards. Formally, this process can be modelled as a Markov Decision Process (MDP) consisting of the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$ is the reward function, and γ is the discount factor. The goal of the RL agent is to find a policy π_ϕ that produces actions that maximize the expected sum of discounted future rewards.

$$\pi_\phi(a_t|s_t) = \text{argmax}_\phi \mathbf{E}_{p(\tau|\phi)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right] \tag{1}$$

In our experiments, we use the value-based method DQN (Mnih et al., 2015) to solve Equation 1.

Multi-armed bandits Multi-armed bandits can be thought of as a special case of RL where the state-space consists only of a single state. Typically evaluated based on regret, multi-armed bandit algorithms focus on the efficient trade-off between exploration and exploitation in order to find an optimal action while incurring the minimum amount of sub-optimal actions. In this work, we adopt one of the most popular algorithms, Upper Confidence Bounding (UCB)(Lai et al., 1985) for an efficient trade-off. The UCB algorithm adds a count-based exploration bonus to the current value estimate of an action before selecting the maximum valued arm:

$$a_t = \text{argmax}_{a \in \mathcal{A}} \left(Q_t(a) + \sqrt{\frac{\log t}{N_t(a)}} \right) \tag{2}$$

Entropy and surprise The notion of surprise derives from the optimization of the entropy of the state marginal distribution under the policy $\pi_\phi(a|s)$, which we denote $d^{\pi_\phi}(s_t)$. Given an estimate of this state marginal distribution, $p_{\theta_{t-1}}(s_t)$, we can express an estimate of the sum of the entropies of the state distribution across a trajectory (see Appendix A of Berseth et al. (2021) for a full derivation):

$$\sum_{t=0}^{T} \mathcal{H}(s_t) = \sum_{t=0}^{T} -\mathbf{E}_{s_t \sim d^{\pi_\phi}(s_t)} \left[\log d^{\pi_\phi}(s_t) \right] \leq \sum_{t=0}^{T} \mathbf{E}_{s_t \sim d^{\pi_\phi}(s_t)} \left[-\log p_{\theta_{t-1}}(s_t) \right] \tag{3}$$

Recalling Equation 1, we can see that minimizing the sum of the state entropy over a trajectory (Equation 3) corresponds to a surprise minimizing agent (Berseth et al., 2021) with a reward function given by:

$$r_{\text{s-min}}(s_t, a_t) = \log p_{\theta_t}(s_{t+1}) \tag{4}$$

Maximizing this objective corresponds to an RL agent with a reward function given by:

$$r_{\text{s-max}}(s_t, a_t) = -\log p_{\theta_t}(s_{t+1}) \tag{5}$$

which is similar to the rewards provided to the EntGame agent in Tiapkin et al. (2023).

Conceptually, this means that the agent is punished (or rewarded) if the observed state s_t is "surprising", that is, if it has high negative log-likelihood under the state marginal distribution estimated so far. Hence, we refer to Equation (4) as surprise-minimization and Equation (5) as surprise-maximization.

4 Surprise-adaptive bandit

Surprise-minimization and surprise-maximization are most effective under particular entropy conditions in the environment, surprise-minimization under high-entropy conditions (Berseth et al., 2021), and surprise-maximization under low-entropy conditions. An intrinsically motivated agent that could capitalize on the advantages each objective provides in its respective entropy regime would be a more powerful and versatile intrinsic learner. Hence, we propose an agent that can alternatively optimize for either objective, with an online adaptive mechanism for selecting the objective.

To design such an adaptive agent, we must first be able to optimize for either single-objective, which requires an estimation of the state marginal distribution at time t, parameterized by θ_t (denoted $p_{\theta_{t-1}}$ in Equation (3)). In general, this estimation can be quite complex; In Berseth et al. (2021), the authors propose a simplification which we adopt here. The method estimates θ_t by first selecting an appropriate distribution family to model observations (i.e. Gaussian, Bernoulli, etc.) and using maximum likelihood estimation to estimate the sufficient statistics of the distributions, fitted to the history of observed states through time t. Further details on estimating the state marginal distribution as well as ablations on the choice of distribution are provided in Appendix A.3.

To adaptively select between the two objectives online, we propose a multi-armed bandit approach. Provided with an appropriate performance signal, a bandit learns to select optimally between actions, trading off exploration with exploitation to minimize the overall regret, making it an appropriate choice for online adaption. The key question is what type of feedback is best to provide the bandit, given access only to intrinsic rewards. We propose a feedback mechanism grounded in the observation that the general goal in both surprise minimization and surprise maximization is for the agent to be able to affect a change in the level of surprise it experiences. In a low-entropy environment, the agent can best affect change by increasing entropy, and vice versa.

We propose using the absolute percent difference between the entropy of the state marginal distribution at the end of the mth episode ($H(p^{(m)}_{\theta_T})$) and that of a *random agent* in the same environment ($H(p^{\text{rand}}_{\theta_T})$) (Equation (6)). The motivation for this is as follows: a random agent cannot control the environment entropy as it cannot take any actions in response to feedback. Agents that produce state visitation distributions whose entropy significantly diverges from that of a random agent must therefore be exerting control over the entropy in the environment. We therefore provide feedback to the bandit which promotes agents that can exert such control by rewarding large deviations from a random agent. Since we are approximating the state marginal distributions by an analytical distribution, we can compute $H(p^{(m)}_{\theta_T})$ analytically from the estimated parameters (see A.3 for further details).

$$f_m = \left| \frac{H(p^{(m)}_{\theta_T}) - H(p^{\text{rand}}_{\theta_T})}{H(p^{\text{rand}}_{\theta_T})} \right| \tag{6}$$

The precise algorithm is as follows (Algorithm 1). At the start of training, we estimate the entropy of a random agent by collecting trajectories using a uniform random policy and averaging the entropy of the state marginal distributions, computed at the end of each trajectory (Line 2). Then, at the start of each episode m, we select an arm from the bandit, represented by binary indicator $\alpha^{(m)}$, according to the UCB algorithm (Line 10), which determines if the agent will receive rewards according to Equation 4 or Equation 5 during the upcoming episode. The agent is trained for a single episode, using any RL algorithm (Line 7). At the end of each episode, the bandit receives feedback f_m on its selection (Line 9). Algorithm 1 shows the full training procedure.

To instantiate the surprise-adaptive agent, we construct an augmented MDP out of the original Markov process. Following Berseth et al. (2021), this augmented MDP has a state space that includes the original state s_t, as well as the sufficient statistics of the state marginal distribution θ_t. We additionally include $\alpha^{(m)}$, as defined above, to ensure the reward function remains Markovian (Castanyer et al., 2023).

In our experiments, all agents were trained using DQN (Mnih et al., 2015), using two convolutional neural networks (CNN) to encode the state. The first CNN encodes the observed state s_t, while the second encodes the sufficient statistic of the state marginal distribution θ_t along with the bandit choice $\alpha^{(m)}$ which is added as an additional channel before applying the CNN. The outputs of the CNNs are concatenated and passed through an MLP that outputs the Q-value. More details on environments and training can be found in Appendix A.

Algorithm 1 Surprise-adaptive agent

Initialize network parameters ϕ, replay buffer β, initial mean of bandit arms $\mu^{(0)}$, and initial optimization direction $\alpha^{(0)} \sim \text{Bern}(0.5)$
Compute $H(p^{\text{rand}}_{\theta_T})$ by rolling out random trajectories
for episode $m = 0, 1 \ldots,$ M **do**
  $s_o \sim p(s_0)$, reset θ_0, $\bar{s_0} = (s_0, \theta_0, 0, \alpha^{(m)})$ ▷ construct initial augmented state
  Set $r(s_t, a_t) = (-1)^{\alpha^{(m)}} - \log p_{\theta_t}(s_t)$ ▷ set reward function
  for $t = 0, \ldots, T$ **do**
    Collect experience and update policy $\phi \leftarrow RL(\phi, \beta)$ ▷ See Berseth et al. (2021)
  end for
  $\mu_i^{(m+1)} \leftarrow \mu_i^{(m)} + \frac{1}{N(i)}(f_m - \mu_i^{(m)})$ if $\alpha^{(m)} = i$ else $\mu_i^{(m)}$
  $\alpha^{(m+1)} \leftarrow \text{UCB}(\mu^{(m+1)})$ ▷ Choose new $\alpha^{(m+1)}$ based on UCB algorithm Lai et al. (1985)
end for

5 Experiments and analysis

To validate the usefulness and effectiveness of our method, we must demonstrate (1) Deficiencies in the single objective agents under particular entropy conditions and how these deficiencies arise from a lack of controllable entropy (2) Ability of the surprise-adaptive agent to select an objective based on the controllable entropy and to mimic the behavior of the single-objective agents and, finally (3) Correlation between entropy control and the emergence of intelligent behaviors.

With these goals in mind, we select several environments for evaluation; First, a set of didactic environments that are designed specifically to create low- and high-entropy conditions to demonstrate both the success and failure modes of single-objective entropy control. Second, a set of RL benchmark environments that are not selected with any particular entropy conditions in mind, and hence are demonstrative of how our algorithm could perform on arbitrary environments with unknown entropy conditions.

For the high-entropy environments, we select the *Tetris* environment used in Berseth et al. (2021) and construct the new *Butterflies* environment, shown in Figure 1. In *Tetris* the agent must survive as long as possible by clearing rows of blocks before they reach the top of the frame. In *Butterflies*, the agent (red) must find and catch butterflies (blue) that are moving randomly in the map, within a fixed-length episode. For the low-entropy environment, we construct a static maze environment (*Maze*), in which the agent navigates around a map with a single goal state, for a fixed-length episode. For both *Butterflies* and *Maze*, we construct small (10x10) and large (32x32) versions of the environments. More details on these new environments are available in Appendix A.2. For the benchmark environments, we select the MinAtar (Young & Tian, 2019) suite of tasks. This suite consists of simplified versions of five Atari games, which are designed to make the state space categorical and fully observable without frame-stacking. Finally, to experiment on image-based observations, we test on *Freeway* from the original Atari games suite (Bellemare et al., 2013)

Our analysis contrasts our method with the two dominant entropy-based intrinsic reward paradigms. Hence, we compare our method (**S-Adapt**) against an exclusively surprise-minimizing agent (**S-Min**) (Berseth et al., 2021) and an exclusively surprise-maximizing agent (**S-Max**). Here, the surprise-maximizing agent represents the space of curiosity and maximum entropy methods (Pathak et al., 2017; Hazan et al., 2019), though we note that our method could be implemented on top of any

desired curiosity-based method. Additionally, as baselines, we compare the entropy-based intrinsic agents against an agent trained on the extrinsic reward (**Extrinsic**), and a random agent (**Random**).

We compare the performance of the various agents both in terms of entropy control and emergent behaviors. As a measure of entropy control, we consider the average surprise the agent experiences across the episode. The metric for emergent behavior that we consider here is the undiscounted episode return, as previous work has argued that entropy control can correlate with task rewards in some environments (Berseth et al., 2021).

5.1 Failures of Single-Objective Entropy Control

First, we demonstrate, qualitatively and quantitatively, the success and failure modes of single-objective entropy-based agents, using the didactic environments.

Qualitatively, we demonstrate the behaviors of **S-Min** and **S-Max** in the *Maze* and *Butterflies* environments in Figure 1. We note that the **S-Min** agent achieves an interesting behavior of catching butterflies in the *Butterflies* environment, but learns a degenerate solution of standing in place in the *Maze* environment. On the other hand, the **S-Max** agent learns to navigate the *Maze* and reach the goal but fails to catch any butterflies in the *Butterflies* environment.

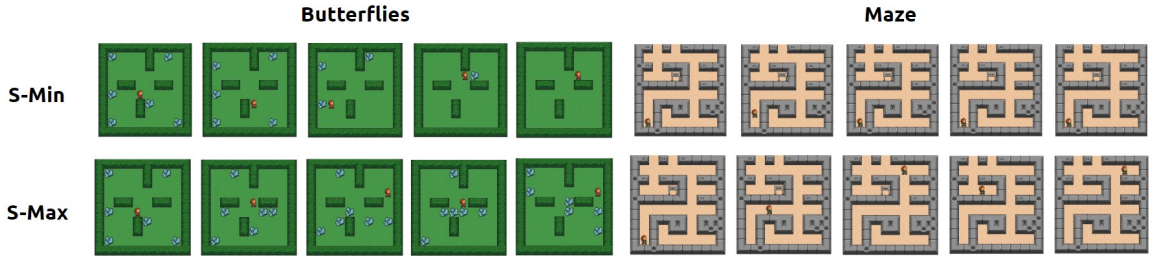
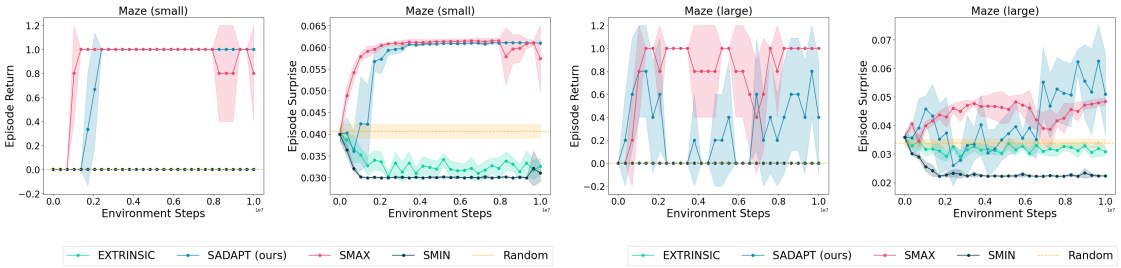


Figure 1: The *Butterflies* (left) and Maze environments (right). **S-Min** trains the agent to actively catch the butterflies in order to prevent diverse state configurations while at the same time preventing the agent to navigate around *Maze*. **S-Max** trains the agent to avoid catching butterflies while navigating the Maze efficiently. These two didactic environments show that current intrinsic objectives fail to provide generally useful objectives for RL agents and cannot adapt.

Quantitatively, we evaluate the average surprise and average extrinsic returns for the agents across training in all environments (Figures 2 to 4). Notably, the **S-Min** agent achieves the lowest or near-lowest entropy in all environments, while the **S-Max** agent achieves the highest or near-highest entropy in all environments, as expected.

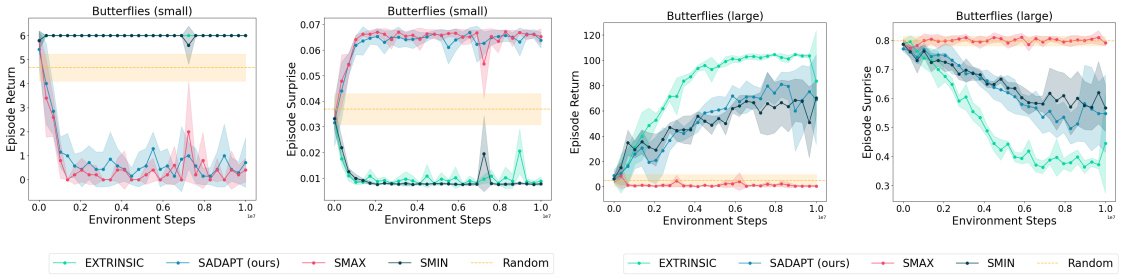


(a) Small Maze (10x10)

(b) Large Maze (32x32)

Figure 2: Average episode return (left) and surprise (right) versus environment interactions (average over 5 seeds, with one shaded standard deviation) in the *Maze* environment. **S-Max** and **S-Adapt** are the only objectives that allow the RL agents to consistently find the goal in the maze. These also cause the largest change in surprise when compared to the random agent.

However, we highlight that the qualitatively interesting direction for entropy control is correlated not with a single objective, but with the scale of the absolute difference in the final entropy achieved by the agent versus that of the **Random** agent. For example, in the *Maze* environment, the **S-Max** agent drives a significant increase in entropy over the **Random** agent, while the **S-Min** agent achieves a relatively small decrease (Figure 2). Similarly, in the *Butterflies* environment, the opposite holds in the large map (Figure 2b). Interestingly, in the small map, the **S-Min** and **S-Max** agents achieve roughly the same absolute change in entropy (Figure 2a). This is because in the smaller map, avoiding butterflies is equally challenging compared to catching butterflies, while in the larger map, the butterflies are easily avoided.



(a) Small Butterflies (10x10)

(b) Large Butterflies (32x32)

Figure 3: Average episode return (left) and surprise (right) versus environment interactions (average over 5 seeds, with one shaded standard deviation) in the *Butterflies* environment. **S-Min**, **Extrinsic** and even the **Random** agent catch most of the butterflies in the small grid. Because of the small size of the grid, surprise-minimization and surprise-maximization are equally effective in entropy control, and hence the **S-Adapt** agent converges to **S-Max**. In the larger grid, however, the **Random** agent can't catch many butterflies and hence has a high-entropy state distribution. Again, the **S-Max** agent learns to also avoid catching butterflies and the **S-Min** agent learns to catch butterflies. However, catching butterflies results in a significant change in the state-marginal entropy in this larger grid. The **S-Adapt** agent identifies this and converges to **S-Min**, resulting in agents that catch more than half of the butterflies without access to the extrinsic reward.

5.2 Adaptive Entropy Control

Capitalizing on the success modes of the single-objective agents, the proposed **S-Adapt** agent can adapt to the entropy landscape to achieve entropy control across all didactic environments (Figures 2 to 4). In *Maze*, the **S-Adapt** agent converges to a surprise-maximizing strategy similar to **S-Max**, as demonstrated by the high entropy achieved by the end of training (Figure 2). On the other hand, in *Tetris*, the **S-Adapt** agent converges to a surprise-minimizing strategy, achieving low entropy on par with the **S-Min** agent by the end of training (Figure 4). In the *Butterflies* environment, an interesting dichotomy in the **S-Adapt** agent's behavior arises. As noted in Section 5.1, in the small grid, both the **S-Min** agent and **S-Max** agent induce roughly the same amount of change in the entropy versus the **Random** agent, using equally challenging strategies (Figure 3a). Here, the **S-Adapt** agent converges to surprise-maximizing

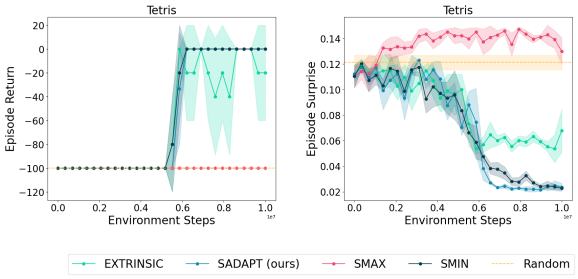
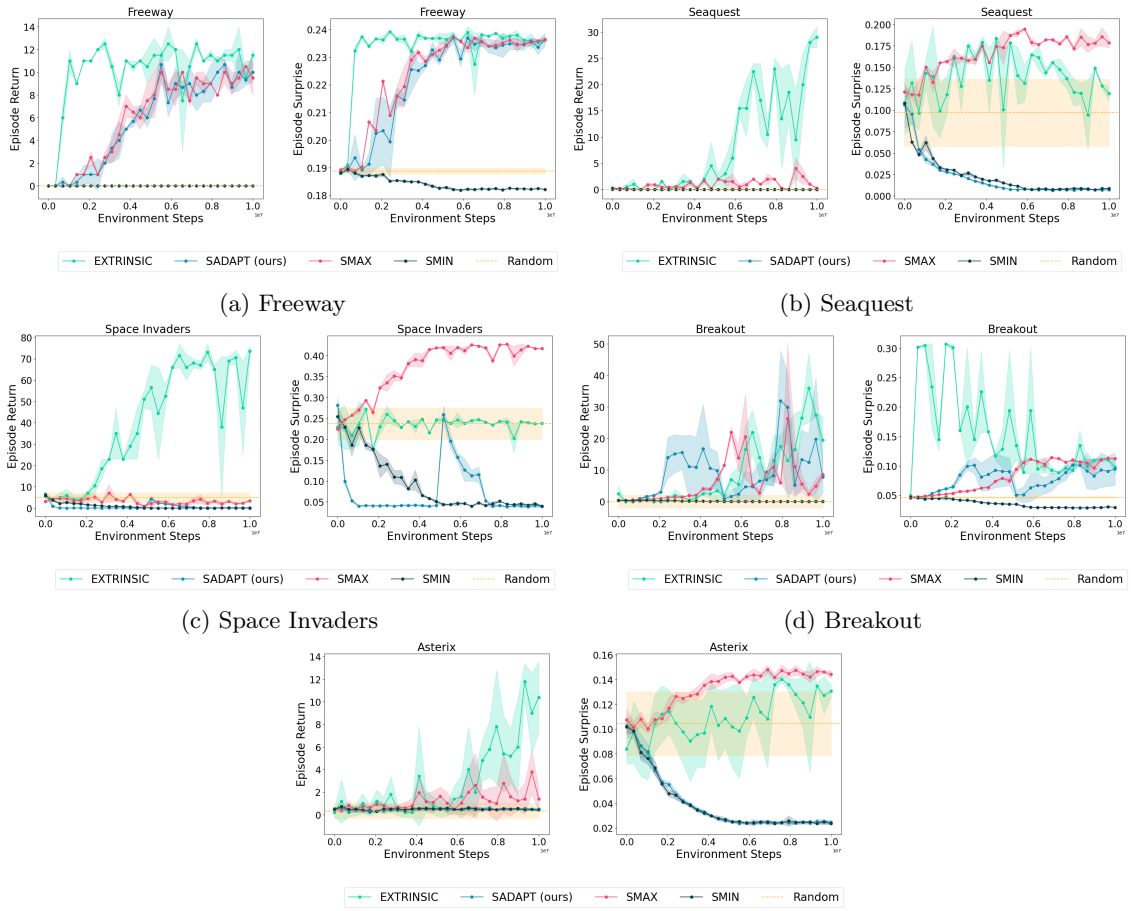


Figure 4: Average episode return (left) and surprise (right) versus environment interactions (average over 5 seeds, with one shaded standard deviation) in *Tetris*. **S-Min**, **S-Adapt**, and **Extrinsic** agents solve the game (i.e. consistently survive for more than 200 steps). Interestingly, the surprise-minimizing objective, which **S-Adapt** converges to, turns out to be a better learning signal than the row-clearing extrinsic reward in *Tetris*.

behavior. However, as the size of the grid is increased, and the density of butterflies decreases, the effect of minimizing entropy becomes much stronger versus the **Random** agent and the **S-Adapt** agent correctly converges to the surprise-minimizing strategy (Figure 2b). More details on the effect of butterfly density on the behavior of the **S-Adapt** agent can be found in Appendix B

Our results have shown that the **S-Adapt** agent can successfully recreate the performance of the **S-Min** and the **S-Max** agents in their respective didactic environments. Next, we investigate controlling entropy across the MinAtar benchmark, shown in Figure 5. Notably, these environments were not constructed with any particular entropy regime in mind. Thus, these results are demonstrative of how the proposed algorithm could perform in an arbitrarily chosen environment.



(a) Freeway

(b) Seaquest

(c) Space Invaders

(d) Breakout

(e) Asterix

Figure 5: Average episode return (left) and surprise (right) versus environment interactions (average over 5 seeds, with one shaded standard deviation) in the MinAtar suite of environments. In all environments the **S-Adapt** agent is able to select the direction for entropy optimization which is most controllable, as demonstrated by the change in entropy from the beginning to the end of training. The **S-Adapt** agent indeed demonstrates emergent behaviors in certain environments, such as *Freeway* where it achieves rewards on par with that of the **Extrinsic** agent. However, in certain environments, like *Seaquest*, *Space Invaders* and *Asterix*, the extrinsic reward is not closely correlated with entropy control, with the **Random** agent and the **Extrinsic** agent achieving similar entropy.

Here again, we see that the **S-Adapt** agent can reliably select the objective with the greatest controllable entropy. Though the difference between **S-Min** and **S-Max** agents in terms of divergence with the **Random** agent is not as strong in some environments, the **S-Adapt** agent consistently

chooses the objective with the relatively larger change in entropy. This provides confirmation that our bandit algorithm can successfully select for controllable entropy in arbitrary environments.

5.3 Emergent Behaviour

Finally, for these objectives to be useful, it is important that they correlate with the emergence of interesting behaviors. Indeed, we note that the extrinsic rewards in the didactic environments generally correlate closely with one of two single-objective agents (Figures 2 to 4). This suggests that these environments have good potential for entropy-based control to elicit emergent behaviors. Importantly, however, the extrinsic reward does not correlate well with strictly one of **S-Min** or **S-Max** in *all* environments. In *Maze*, **S-Max** achieves high rewards, while in *Butterflies* and *Tetris*, **S-Min** achieves high rewards. On the other hand, the **S-Adapt** agent achieves high task rewards, on par or better than the **Extrinsic** agent across *all* didactic environments.

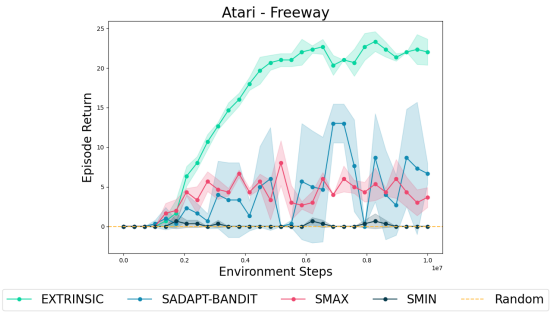


Figure 6: Average episode return versus environment interactions (average over 5 seeds, with one shaded standard deviation) in the Atari Freeway environment. The **S-Adapt** agent learns useful behaviours (making progress in the original task) from image-based observations. The **Extrinsic** agent achieves the highest returns as it exploits the task rewards, the **S-Max** agent achieves slightly lower returns than the **S-Adapt** agent, while the **S-Min** agent achieves zero returns.

Additionally, in some MinAtar environments, the entropy-based agents exhibit emergent behavior similar to that of the **Extrinsic** agent. In the *Freeway* environment (Figure 5a), the **S-Adapt** agent achieves competitive rewards with the **Extrinsic** agent. A similar result is observed in *Breakout* (Figure 5d). However, other environments, like *Space Invaders* and *Seaquest* (Figures 5b and 5c) do not appear to be good candidates for intrinsic entropy control, since the **Extrinsic** and **Random** agents achieve similar entropy.

Finally, we investigate the emergence of interesting behaviors in a more complex, image-based environment using *Atari Freeway* (Figure 6) as a case study. Unlike the previous environments, observations in pixel space are non-binary and hence cannot be modeled using Bernoulli distributions. Instead, we model the state marginal using a Gaussian distribution (see Appendix A.3 for more details). The results show that both the **S-Max** and **S-Adapt** agents achieve respectable results as compared to the extrinsic agent. Moreover, in this environment, the emergent behavior of the **S-Adapt** agent is qualitatively different from both **S-Max** and **S-Min** agents; The **S-Adapt** agent solves the game more frequently than the **S-Max** agent. This hints that mixing entropy maximization and minimization in one adaptive objective induces emergent behaviors that cannot be learned by exclusively optimizing for surprise minimization or maximization alone.

6 Conclusion

Our experiments demonstrate encouraging results for a surprise-adaptive agent. The **S-Adapt** agent can select the objective with the more controllable landscape across both didactic environments and benchmark environments. Moreover, the **S-Adapt** agent inherits the emergent behaviors of the single-objective agents, achieving high rewards across all didactic environments, which neither of the single-objective agents nor the extrinsic agent is able to achieve. Further work is needed to understand exactly under what conditions such emergent behaviors can manifest, and how to elicit them more reliably in arbitrary environments like MinAtar. Possible directions for improvement here could include better methods for estimating the state marginal distribution with more accuracy. Moreover, an interesting extension to this work would be to apply an adaptive agent in the continual learning setting, where adaptation can occur at any time, not only at episode end.

Acknowledgments

We want to acknowledge funding support from NSERC, FRQNT, and CIFAR and compute support from the Digital Research Alliance of Canada, Mila IDT and NVidia.

References

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *International conference on machine learning*, pp. 507–517. PMLR, 2020.

Christopher Bamford. Griddly: A platform for ai research in games. *Software Impacts*, 8:100066, 2021.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Glen Berseth, Daniel Geng, Coline Manon Devin, Nicholas Rhinehart, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. Smirl: Surprise minimizing reinforcement learning in unstable environments. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=cPZOyoDloxl.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=rJNwDjAqYX.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=H1lJJnR5Ym.

Roger Creus Castanyer, Joshua Romoff, and Glen Berseth. Improving intrinsic exploration by creating stationary objectives. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=YbZxT0SON4.

Arnaud Fickinger, Natasha Jaques, Samyak Parajuli, Michael Chang, Nicholas Rhinehart, Glen Berseth, Stuart Russell, and Sergey Levine. Explore and control with adversarial surprise. *arXiv preprint arXiv:2107.07394*, 2021.

Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2): 127–138, 2010.

Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pp. 2681–2691. PMLR, 2019.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Shengyi Huang, Rousslan Fernand JulienDossa Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João GM Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *The Journal of Machine Learning Research*, 23(1):12585–12602, 2022.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJ6yPD5xg.

Arnav Kumar Jain, Lucas Lehnert, Irina Rish, and Glen Berseth. Maximum state entropy exploration using predecessor and successor representations. *Advances in Neural Information Processing Systems*, 36, 2023.

Maximilian Karl, Justin Bayer, and Patrick van der Smagt. Efficient empowerment. *arXiv preprint arXiv:1509.08455*, 2015.

Kuno Kim, Megumi Sano, Julian De Freitas, Nick Haber, and Daniel Yamins. Active world model learning with progress curiosity. In *International conference on machine learning*, pp. 5306–5315. PMLR, 2020.

Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 IEEE congress on evolutionary computation*, volume 1, pp. 128–135. IEEE, 2005.

Tze Leung Lai, Herbert Robbins, et al. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Sam Lobel, Akhil Bagaria, and George Konidaris. Flipping coins to estimate pseudocounts for exploration in reinforcement learning. In *International Conference on Machine Learning*, pp. 22594–22613. PMLR, 2023.

Manuel Lopes, Tobias Lang, Marc Toussaint, and Pierre-Yves Oudeyer. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Advances in neural information processing systems*, pp. 206–214, 2012.

Marlos C Machado, Marc G Bellemare, and Michael Bowling. Count-based exploration with the successor representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5125–5133, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Ted Moskovitz, Jack Parker-Holder, Aldo Pacchiano, Michael Arbel, and Michael Jordan. Tactical optimism and pessimism for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12849–12863, 2021.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Nicholas Rhinehart, Jenny Wang, Glen Berseth, John Co-Reyes, Danijar Hafner, Chelsea Finn, and Sergey Levine. Information is power: intrinsic control via information capture. *Advances in Neural Information Processing Systems*, 34:10745–10758, 2021.

Jürgen Schmidhuber. Curious model-building control systems. In *Proc. international joint conference on neural networks*, pp. 1458–1463, 1991.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE transactions on autonomous mental development*, 2(3):230–247, 2010.

Zekun Sun and Chaz Firestone. The dark room problem. *Trends in Cognitive Sciences*, 24(5):346–348, 2020.

Daniil Tiapkin, Denis Belomestny, Daniele Calandriello, Eric Moulines, Remi Munos, Alexey Naumov, Pierre Perrault, Yunhao Tang, Michal Valko, and Pierre Menard. Fast rates for maximum entropy exploration. In *International Conference on Machine Learning*, pp. 34161–34221. PMLR, 2023.

Kenny Young and Tian Tian. Minatar: An atari-inspired testbed for thorough and reproducible reinforcement learning experiments. *arXiv preprint arXiv:1903.03176*, 2019.

Andrew Zhao, Matthieu Gaetan Lin, Yangguang Li, Yong jin Liu, and Gao Huang. A mixture of surprises for unsupervised reinforcement learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=OHkq7qNr72-.

Ruihan Zhao, Pieter Abbeel, and Stas Tiomkin. Efficient online estimation of empowerment for reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=u2YNJPcQlwq.

A Environment and Training Details

A.1 Training Details

All agents were trained using DQN (Mnih et al., 2015). Reward values are normalized by subtracting the rolling mean and dividing by the standard deviation before fitting the Q network. For the **S-Adapt** agent, we use the original UCB algorithm with exploration coefficient 2 in the *Maze* (large) and MinAtar environments, for all other environments we set the exploration coefficient to $\sqrt{2}$. we trained all agents using the implementation of DQN from CleanRL (Huang et al., 2022). We trained all agents with a learning rate of 0.0001 with Adam optimizer, a discount factor of 0.99, a batch size of 32, a replay buffer size of 1M, and for 10M environment interactions. We use epsilon-greedy for exploration with a linearly decaying epsilon from a value of 1 to 0.01, decaying over the first 10% of timesteps in all environments except MinAtar and Atari which decays over the first 50% of time steps. Model architecture details for each environment are provided in the next section.

A.2 Environments

Tetris We take the *Tetris* environment directly from the implementation provided by the authors of (Berseth et al., 2021). In this environment, the agent receives 0 at all steps, except for a losing step which results in a -100 reward. The maximum episode length is 200. Environment observations and the sufficient statistic of the state marginal are flattened before being fed into two independent two-layer MLPs with hidden dimensions 120 and 84. The outputs of the MLPs are concatenated and passed through a linear layer that outputs the Q-value.

Maze We constructed custom *Maze* environments (small and large) using the Griddly platform (Bamford, 2021). A pixel-rendering of the small and large mazes used in our experiments can be found in Figure 7. The task reward in both environments is +1 when the agent reaches the goal and 0 otherwise.

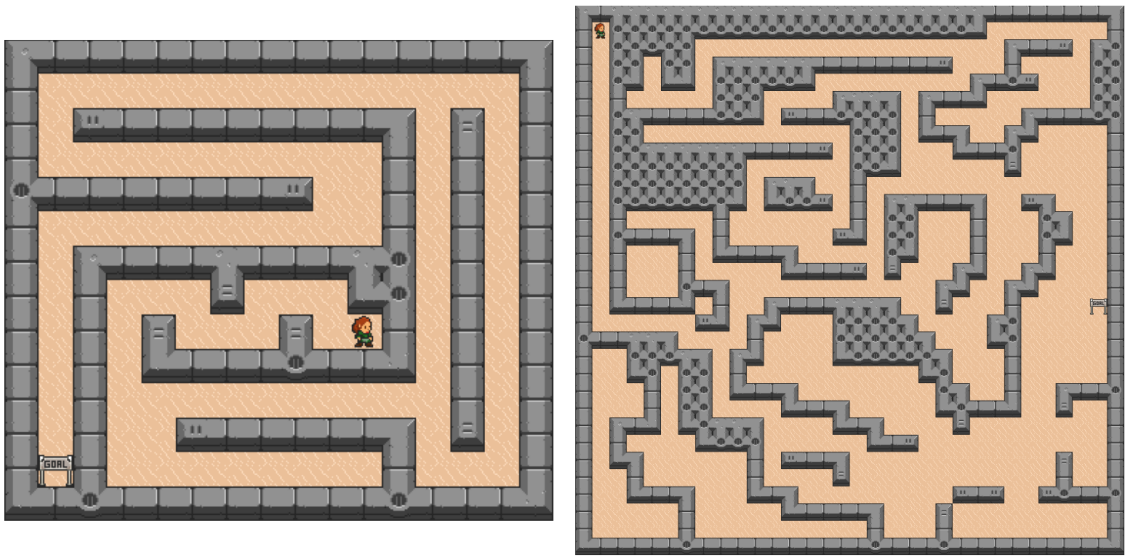


Figure 7: Pixel-rendering of the small maze (left) and the large maze (right)

The size of the small maze is 10x10 and the episode length is 100. Environment observations and the sufficient statistic of the state marginal are passed through two independent CNNs with a single convolutional layer. The outputs of the CNNs are concatenated and passed through a single-layer MLP with hidden dimension 512 that outputs the Q-value.

The size of the large maze is 32x32 and the episode length is 250. Environment observations and the sufficient statistic of the state marginal are passed through two independent CNNs with three convolutional layers with kernel size of (3,3), a stride value of 2 and a padding value of 1. The outputs of the CNNs are concatenated and passed through a single-layer MLP with hidden dimension 512 that outputs the Q-value.

Butterflies We constructed the custom *Butterflies* environment (small and large) using the Griddly platform (Bamford, 2021). The task reward in both environments is +1 when the agent catches a butterfly and 0 otherwise.

The size of the small map is 10x10 and the episode length is 100, while the size of the large map is 32x32 and the episode length is 500. We use the same architecture as the *Maze* environment for estimating the Q-value.

MinAtar In MinAtar environments, we use the same architecture as the *Butterflies* environments and we set the episode length to 500.

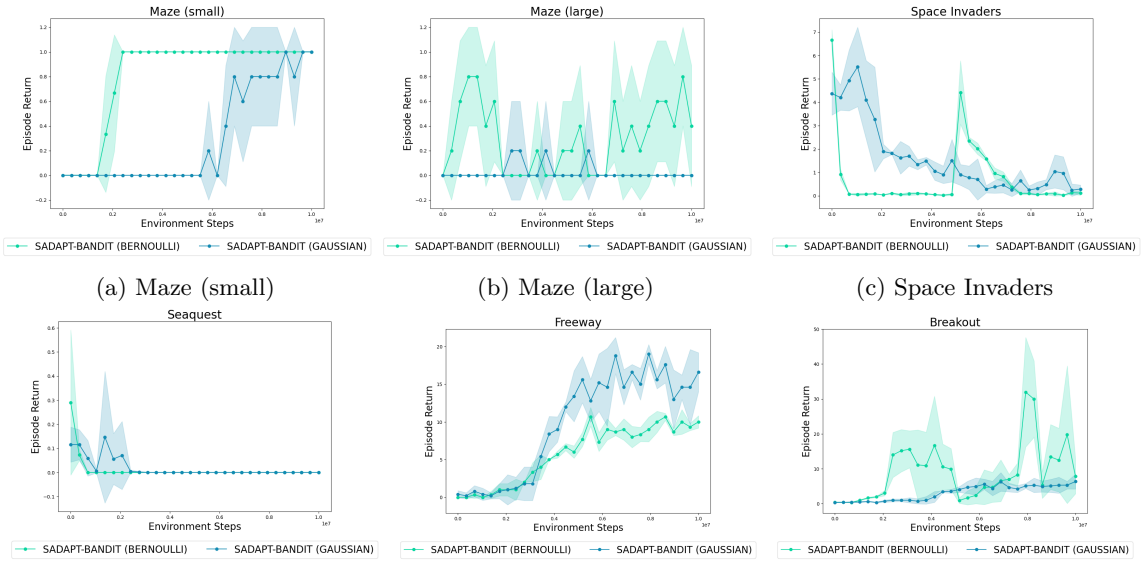
Atari In Atari *Freeway* environment, we use the same architecture and pre-processing as in Mnih et al. (2015). We use the same multiple CNN architecture as the *Maze* environment for estimating the Q-values from the augmented state with sufficient statistics.

A.3 Estimation of State Marginal Distribution

In all binary environments (*Tetris, Maze, Butterflies, MinAtar*), the observed state s_t is a binary entity map of size $H \times W \times C$, where H is the height of the map, W is the width of the map and C is the number of channels, with each channel representing a single object type in the environment. A value of one is set in the (h, w) position of channel c (denoted $s_t^{h,w,c}$) if an object of type c currently occupies the (h, w) position in the map, and zero otherwise. The state marginal distribution is estimated as $H \times W \times C$ independent Bernoulli distributions, with probability $p_t^{h,w,c} = \frac{\sum_{t'=0}^{t} s_{t'}^{h,w,c}}{t}$, which constitutes a sufficient statistic for the Bernoulli distribution. Hence, the sufficient statistic of the entire state marginal distribution is given by $\theta_t = \{p_t^{h,w,c} : h \in H, w \in W, c \in C\}$ and is the same shape as the observations s_t.

The choice of the Bernoulli distribution is justified by the binary nature of the data. However, we perform an ablation using a Gaussian distribution as an alternative to confirm the validity of this choice (Figure 8).

In the image-based environment (Atari *Freeway*), the observed state s_t is an image. Here, we use a Gaussian distribution for the state marginal estimation. Using the same notation as above, the sufficient statistics for the Gaussian distribution are given by empirical mean and variance $\mu_t^{h,w,c} = \frac{\sum_{t'=0}^{t} s_{t'}^{h,w,c}}{t}$, $\sigma_t^{h,w,c} = \frac{\sum_{t'=0}^{t} (\mu_t^{h,w,c} - s_{t'}^{h,w,c})^2}{t}$. The sufficient statistic for the entire state marginal distribution is then given by $\theta_t = \{\mu_t^{h,w,c}, \sigma_t^{h,w,c} : h \in H, w \in W, c \in C\}$.



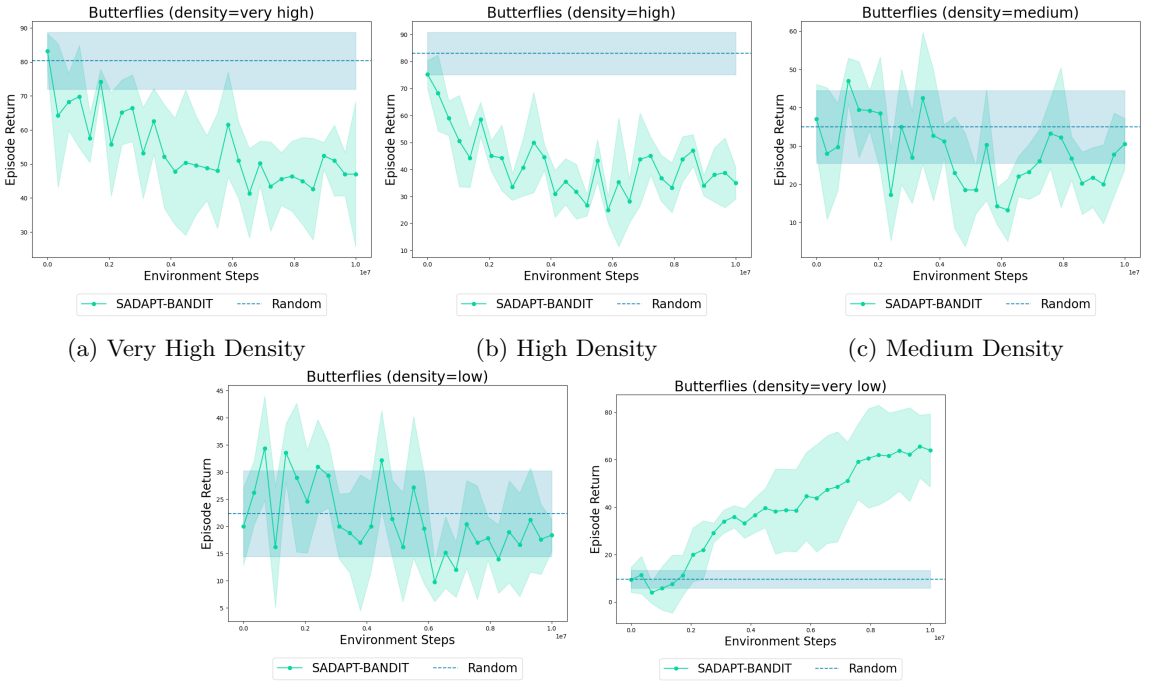
(a) Maze (small) (b) Maze (large) (c) Space Invaders

(d) Seaquest (e) Freeway (f) Breakout

Figure 8: Average episode return of the **S-Adapt** (average over 5 seeds, with one shaded standard deviation), using Gaussian and Bernoulli distributions for estimating the state marginal distribution.

B Additional Experiments

Here we present additional results on the impact of butterfly density on the behavior of the **S-Adapt** agent.



(a) Very High Density (b) High Density (c) Medium Density

(d) Low Density (e) Very Low Density

Figure 9: Average episode return of the **S-Adapt** agent (average over 5 seeds, with one shaded standard deviation) over various butterflies densities in the *Butterflies* (large) environment. At (very) high density (Figures 9a and 9b), the **Random** agent resembles the **S-Min** agent and catches large number of butterflies as indicated by the high episode return. Hence, the **S-Adapt** agent converges to surprise-maximization to induce large absolute difference in entropy from the **Random** agent and avoids butterflies as indicated by the low episodic return. In contrast, at very low density (Figure 9e), the **Random** agent is unable to catch butterflies and resembles the **S-Max** agent. The **S-Adapt** agent converges to surprise-minimization and almost catches all the butterflies as indicated by the high episodic return. At medium and low densities (Figures 9c and 9d), the **S-Adapt** agent oscillates between surprise-maximization and surprise-minimization as they roughly induce the same absolute difference in entropy.