Guiding Exploration Towards Impactful Actions

Vaibhav Saxena
Georgia Institute of Technology
vsaxena33@gatech.edu

Jimmy Ba
University of Toronto
jba@cs.toronto.edu

Danijar Hafner
University of Toronto
Google Research, Brain Team
mail@danijar.com

Abstract

To solve decision making tasks in unknown environments, artificial agents need to explore their surroundings. While simple tasks can be solved through naive exploration methods such as action noise, complex tasks require exploration objectives that direct the agent to novel states. However, current exploration objectives typically reward states purely based on how much the agent learns from them, regardless of whether the states are likely to be useful for solving later tasks. In this paper, we propose to guide exploration by empowerment to focus the agent on exploring regions in which it has a strong influence over its environment. We introduce a simple information-theoretic estimator of the agent's empowerment that is added as a reward term to any reinforcement learning method. On a novel BridgeWalk environment, we find that guiding exploration by empowerment helps the agent avoid falling into the unpredictable water, which substantially accelerates exploration and task learning. Experiments on Atari games demonstrate that the approach is general and often leads to improved performance.

1 Introduction

Reinforcement learning algorithms shape the behavior of artificial agents by maximizing the sum of expected rewards that an agent could achieve while interacting with the environment (Konda and Tsitsiklis, 1999), and train policies by crediting or discrediting actions based on their associated values (Williams, 1992). When environments provide sparse rewards, these values fail to assign credit to useful actions that would otherwise lead to long-term rewards (Sutton, 1984). In the past, researchers have addressed learning with sparse rewards (Ng et al., 1999) with some recent progress leveraging reward engineering derived from human experts for a number of tasks such as navigation (Chaplot et al., 2020; Batra et al., 2020) and object rearrangement in simulated environments (Szot et al., 2021). However, such engineered rewards do not generalize well to other complex tasks, which gives rise to the need for more general utility functions with minimum computation footprint and application to potentially all complex environments.

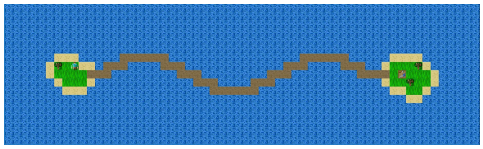


Figure 1: BridgeWalk Environment. The player spawns on the left island and collects reward once it crosses the bridge and reaches the right island. Movement of the player is deterministic on land and random in water.

In an attempt towards achieving globally optimal performance, RL agents need to devise intrinsic motivations to enable efficient learning of a diverse range of tasks (Singh et al., 2004). Even exploring diverse regions of the underlying state-spaces is exacerbated by the visual complexity of inputs from partially-observable environments. Recently, Sekar et al. (2020) proposed a method for inferring the expected novelty of a state by imagining future states using a learned world model. Such motivations for exploration are effective but also very general in the sense that the explored states would likely not be useful in solving later tasks.

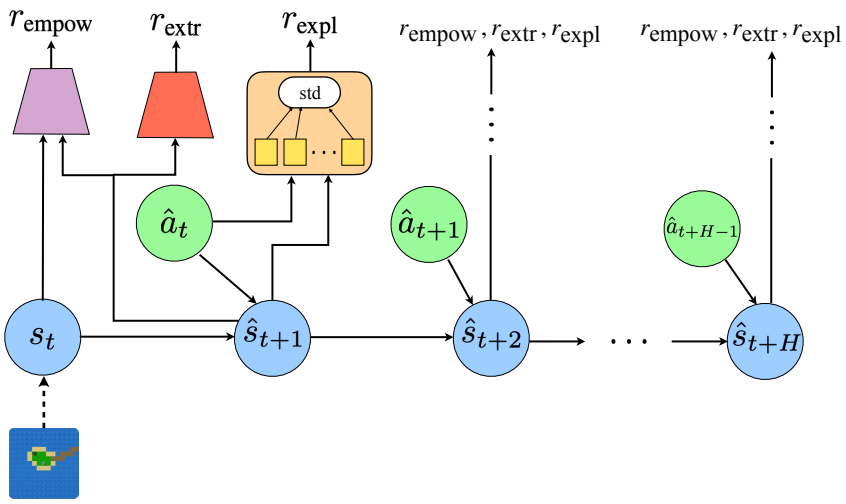


Figure 2: Learning with extrinsic and intrinsic rewards. The agent receives an initial observation from the environment and infers a posterior state representation. Using the learned world model, the agent generates experience with the environment while predicting extrinsic rewards, and intrinsic motivations for exploration and empowerment. The exploration reward is computed as the standard deviation of K independent representations of the state, while the empowerment reward is computed as the negative entropy of a distribution trained to decode actions from consecutive state pairs. Refer to Section 3.1 for more details.

In this paper, we propose empowerment as a means for directed exploration in partially-observable environments with visually complex outputs. We build upon the model-based RL framework in DreamerV2 (Hafner et al., 2020), where we learn a world model that can imagine future states and rewards, and use experience collected with the learned MDP to learn state-values and parameterized policies. We present an efficient way of computing the intrinsic empowerment reward using a parameterized function of consecutive state-pairs, trained by regressing the output towards the action that led to the transition between those states. We use the computed empowerment reward in conjunction with the exploration reward to direct exploration towards states where the agent can maximally influence outcomes. To evaluate our approach, we conduct experiments on five Atari games from the Arcade Learning Environment (Bellemare et al., 2012) and a novel simulation environment called BridgeWalk. The BridgeWalk environment is developed to specially evaluate agents on their ability to solve tasks in highly stochastic environments with sparse rewards.

2 Related Works

World Models Recent progress in deep learning has enabled learning of informative representations of high-dimensional temporal data (Srivastava et al., 2015; Vondrick et al., 2016). Latent variable models with autoregressive prediction in time (Denton and Fergus, 2018) have proven to be efficient methods of video prediction, with adversarial objectives helping generate sharp videos (Lee et al., 2018). Hierarchical models for modeling video (Sønderby et al., 2016; Zhao et al., 2017; Saxena et al., 2021) have opened up new directions for learning high-level planners. Such progress in learning recurrent latent-variable models has paved the way for model-based RL on partially-observable environments using learned world models. Recently, recurrent state-space models enabled successful planning for control tasks (Hafner et al., 2019a;b) and Atari games (Hafner et al., 2020).

Exploration By virtue of the RL objective, agents are penalized for taking low-value courses of actions making them prone to getting stuck on certain reward modes in the state-space. This makes exploration an intrinsically important task, especially in partially-observable environments where even evaluating effective exploration is hard. While some methods directly measure novelty of states to affect exploration (Lehman and Stanley, 2011; Bellemare et al., 2016; Burda et al., 2018), others measure disagreement within learned world models (Shyam et al., 2018; Sekar et al., 2020). Recently, Whitney et al. (2021) showed that decoupling exploration from task policies can be very effective for sample-efficient learning.

Mutual information objectives MI objectives have been extensively used for learning high-level representations of action and state spaces. VALOR (Achiam et al., 2018) learns 'skills' by encoding

a sampled skill variable into a trajectory using a policy, and then decoding the same skill using a probabilistic decoder. The decoder is trained to give high probability to the sampled skill that was used to generate the input trajectory. VALOR was preceded by VIC (Gregor et al., 2016), DIAYN (Eysenbach et al., 2018) which differ with regards to whether the MI was between skill and the whole trajectory, or just the initial and end states, or all states. Our method of computing empowerment employs a very similar technique where we maximize the mutual information between the action and next state, given current state, by training a decoder to output the action that caused the transition between the two states.

3 Method

Learning to solve tasks by reinforcement refers to collecting sequences of observations by acting in the environment, and then positively or negatively reinforcing the agent's actions based on the collected long-term rewards. Since these observations can be very high-dimensional and do not represent the entire summary of the agent's experience so far, we encode the entire experience of the agent in a trajectory into low-dimensional latent variables which are then used for decision-making. We assume the environment is operating as a partially-observable Markov decision process (POMDP), and the emission and state transitions of this POMDP as probability distributions parameterized using neural networks. Using predicted rewards extrinsic and intrinsic to the agent we then compute state-values and use them to reinforce actions of a parameterized policy. Our reward prediction heads for extrinsic, exploration, and empowerment rewards are trained separately from and without any effect on the trained world model.

3.1 Model Learning

We use a recurrent state-space model with discrete latent variables (Hafner et al., 2020) to model different components in the POMDP. We parameterize and learn the state transition distribution $p_\psi(s'|s,a)$, the decoder distribution $p_\psi(o|s)$, and an approximate encoder $q_\psi(s|o)$. Different components of the world model are illustrated in Figure 2. The decoder is assumed to be a multivariate Gaussian with diagonal covariance. The transition and encoder are assumed to be vectors of categorical distributions, trained using straight-through gradients (Bengio et al., 2013). We train the entire world model to maximize the log-likelihood of observations collected from the environment. Since computing the exact likelihood requires marginalizing over the entire latent space, we instead use the ELBO as our training objective. The training loss for the world model is given by

$$\mathcal{L}(\psi) \doteq \mathrm{E}_{q_\psi(s_{1:T}|o_{1:T},a_{1:T})}\Big[\sum_{t=1}^{T} \ln p_\psi(o_t|s_t,a_t) + \zeta\,\mathrm{KL}\left[q_\psi(s_t|o_t,s_{t-1})\;||\;p_\psi(s_t|s_{t-1})\right]\Big]. \quad (1)$$

After learning the world model, we also learn the extrinsic reward (r_{extr}) that the agent might obtain when interacting with the environment. We parameterize this reward head using a neural network and train it using a squared loss.

3.2 Intrinsic Motivations

In addition to training to maximize rewards extrinsic to our agent from the environment, we consider the interplay between two model-based intrinsic motivations - exploration and empowerment.

Exploration We compute the exploration reward at any state using the strategy developed by Sekar et al. (2020). In this approach, we compute K independent neural network representations of a latent state s obtained from the learned encoder, say $\{\tilde{s}^i\}_{i=1}^K$. The exploration reward for being in that state is computed as

$$r_{\text{expl}}(s) = \text{std}(\tilde{s}^1, \ldots, \tilde{s}^K). \quad (2)$$

If s is multi-dimensional, we compute the average standard deviation across all dimensions to obtain a scalar reward. This reward quantifies the epistemic uncertainty about different points in the state-space, and hence acts as an intrinsic motivation for the agent to visit unexplored regions of the environment.

Empowerment Klyubin et al. (2005) defined empowerment as the channel capacity of the agent's actuation channel, which could be interpreted as the amount of information that the agent could inject into the MDP that could later be captured in the next state. In more recent approaches, this objective was defined as the mutual information between the action and the next state distribution (Mohamed and Rezende, 2015), which can be broken down into two entropies as,

$$I(a_t; s_{t+1}|s_t) = H[a_t|s_t] - H[a_t|s_{t+1}, s_t], \tag{3}$$

where $H[a_t|s_t]$ denotes the marginal entropy of the action given the current state, and $H[a_t|s_{t+1}, s_t]$ denotes the conditional entropy of the action given the next state. Even though we know the action marginal $p(a_t|s_t)$ (agent's policy) and the state transition distribution $p(s_{t+1}|a_t, s_t)$ (via the learned world model), computing the posterior over the actions, $p(a_t|s_{t+1}, s_t)$, would be hard as it requires marginalization over the entire action space (Mohamed and Rezende, 2015). Hence, we compute an approximate posterior $q(a_t|s_t, s_{t+1})$ over the actions and use its entropy to approximate the conditional entropy term in the mutual information. Based on findings from (Hafner et al., 2020) that long-term maximization of action entropy does not substantially help the agent in seeking out states with high action entropy, we omit the entropy of the action marginal in our reward definition. Given a pair of consecutive states $(\hat{s}_t, \hat{s}_{t+1})$, we define the empowerment reward as

$$r_{\text{empow}}(\hat{s}_t, \hat{s}_{t+1}) = \mathrm{E}_{q(a_t|\hat{s}_t, \hat{s}_{t+1})}[\ln q(a_t \mid \hat{s}_t, \hat{s}_{t+1})]. \tag{4}$$

We normalized all rewards by dividing them by a running estimate of the standard deviation of the returns (Burda et al., 2018).

3.3 Agent Behavior

Given a posterior state s_0, the agent imagines experience upto a horizon H, say $\{\hat{a}_{i-1}, \hat{s}_i, r^i_{\text{extr}}, r^i_{\text{empow}}, r^i_{\text{expl}}\}_{i=1}^{H}$. The predicted rewards and intrinsic motivations from the world model help learn the agent behavior in an actor-critic framework. We define the actor as a categorical distribution parameterized using a neural network, that aims to maximize the output of the critic. The critic deterministically outputs the expected sum of rewards (extrinsic or intrinsic) that the agent might achieve while interacting with the MDP learned by the world model.

Critic We learn a separate critic for each reward type by regressing a parameterized function towards the TD(λ) formulation of state values using a squared loss, given by,

$$\mathcal{L}_{critic}(\phi) \doteq \sum_{t=1}^{H} \frac{1}{2}\Big(v_\phi(\hat{s}_t) - \text{sg}\left(V_t^\lambda\right)\Big)^2, \tag{5}$$

where H is the maximum horizon length for which the agent interacts with the learned MDP in the world model, $\hat{s}_t$ is a latent state in the MDP, $v_\phi(\hat{s}_h)$ is the critic output corresponding to a reward head, $\text{sg}(.)$ is the stop-gradient operation, V^λ is the TD(λ) target that is computed by bootstrapping with a separate target state-value function (Mnih et al., 2015). We define V_t^λ recursively as

$$V_t^\lambda = \hat{r}_t + \begin{cases} v(\hat{s}_H), & \text{if } t = H \\ (1-\lambda)\, v(\hat{s}_t) + \lambda V_{t+1}^\lambda, & \text{if } t < H. \end{cases} \tag{6}$$

where $\hat{r}_h$ is the reward sample obtained while interacting with the learned world model.

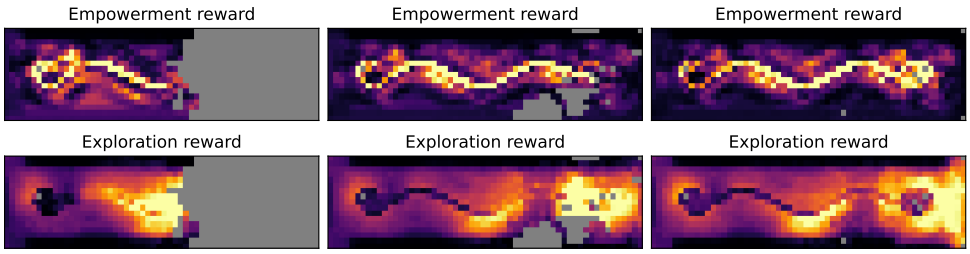
Since we learn a separate critic (and target critic) for each reward type, the total value of a state is computed by summing up the individual values as

$$v_{\text{total}}(\hat{s}_h) \doteq \beta_{\text{extr}} v_{\text{extr}}(\hat{s}_h) + \beta_{\text{expl}} v_{\text{expl}}(\hat{s}_h) + \beta_{\text{empow}} v_{\text{empow}}(\hat{s}_h), \tag{7}$$

where the β's are reward scale hyperparameters that sum to 1 (Badia et al., 2020). We implement these weighted critics by scaling the rewards by these tunable hyperparameters before constructing corresponding targets for each reward type.

Policy The agent selects actions using a policy that is parameterized by a softmax in action preferences, where the action preferences are computed using a neural network with parameters θ. This actor aims to maximize the output of the critic $v_{\text{total}}(\hat{s}_t)$, for each state $\hat{s}_t$ that it comes across while interacting with the MDP learned by the world model. We use reinforce gradients (Williams, 1992) to update the actor parameters using the loss

$$\mathcal{L}_{\text{actor}}(\theta) \doteq -\sum_{t=1}^{H} \Big(\big(V_t^\lambda - v_{\text{total}}(\hat{s}_t)\big) \ln p_\theta(\hat{a}_t \mid \hat{s}_t) + \eta\, \mathrm{H}[p_\theta(\hat{a}_t \mid \hat{s}_t)]\Big). \tag{8}$$



(a) Trained for 300k frames. (b) Trained for 500k frames. (c) Trained for 700k frames.

Figure 3: Learned empowerment and exploration rewards at different locations in the BridgeWalk environment (deterministic movement on land and random movement in water). Dark regions on the map show regions of low reward, bright yellow regions represent a high reward, and grey regions show locations that the agent did not visit or were inaccessible during all rollouts. Each episode starts with the agent on the island on the left. As training progresses, the agent updates its intrinsic motivations of being at each position. Empowerment is always low in the water where movement is random, hence restricting the exploration to interesting parts of the environment. This eventually results in finding the goal on the right side faster.

4 Experiments

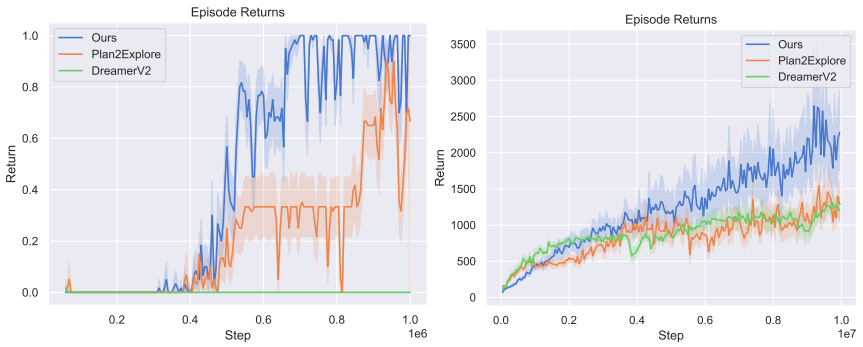
We performed experiments in a novel simulation environment called BridgeWalk, and five popular Atari games from the Arcade Learning Environment (Bellemare et al., 2012) - Enduro, Pong, Q*bert, Seaquest, and Space Invaders. Our experiments on BridgeWalk specifically evaluate the agent's ability to explore regions of the environment where it has maximal influence. In addition to reporting average rewards collected by the agent in all these environments, we also visualize the computed empowerment for trajectories in Seaquest and BridgeWalk. We consider one model-free baseline - DQN (Mnih et al., 2013) - and two model-based baselines - DreamerV2 (Hafner et al., 2020) and Plan2Explore (Sekar et al., 2020) - for comparison. We show that our agent clearly outperforms the DreamerV2 and Plan2Explore agents on BridgeWalk, and beats all three baselines on most Atari games.

Training Details We use the same network architectures for the encoder, decoder, and different components of the world model, across all environments. Our image encoder consists of 4 convolutional layers (LeCun et al., 1989) with channel depth doubled at every convolution, and halved at every deconvolution. We use the ELU non-linearity (Clevert et al., 2016) after each hidden layer of all neural networks. The components for extrinsic, exploration and empowerment reward heads are parameterized using a neural network with 4 hidden layers with 400 activations each and a ELU non-linearity. Our best performing β hyperparameters for weighing different rewards for BridgeWalk are 0.45, 0.1, and 0.45 for extrinsic, empowerment, and exploration rewards respectively, and the same for Atari games are 0.8, 0.1, and 0.1 respectively. We train the agent on 1M frames of experience in the BridgeWalk environment and 10M frames for all Atari games. We use the Adam optimizer (Kingma and Ba, 2014) for training, with $\epsilon =$ 1e-5 and learning rate of 4e-5 for the actor parameters, 1e-4 for the critic parameters, and 1e-4 for the parameters of the world model. Our experiments took about 22 hours to train on BridgeWalk, and about 2 days 22 hours to train on Atari using a single Nvidia Titan Xp GPU.

4.1 BridgeWalk

The BridgeWalk environment consists of two islands connected by a single-step-wide bridge surrounded by water on all sides. At the start of an episode, the player spawns on one island and collects a reward of 1.0 if it is able to cross the bridge and reach the other island, at which point the episode terminates. The player's movement on the map remains deterministic while it wanders anywhere on land (island or bridge), but becomes stochastic when it steps into the water, where it can be driven randomly in any direction or in the direction of the water current, as defined during environment setup. We have made the code for this environment available at: `github.com/danijar/bridgewalk`.

During initial phases of training, when the agent has not seen any reward, all policies retain a high entropy. When the agent's objective is to explore, the policy tries to push the player to explore new regions of the environment, both in water and land. If the player steps into the water, the current



(a) BridgeWalk. (b) Seaquest.

Figure 4: Episodic returns on BridgeWalk and Seaquest (Atari 2600). We average results over 3 random seeds, and report the mean and one standard deviation around it. Our agent with the empowerment and exploration objectives outperforms the purely extrinsic baseline (DreamerV2) as well as one with added exploration (Plan2Explore). On BridgeWalk, we observe that exploration is specially critical to finding the goal state in the sparsely rewarding environment. Additionally, the empowerment objective helps the agent achieve maximum average reward in roughly half the number of training frames.

may push the agent back towards the region it has already explored, in which case the exploration objective could actually incentivize the agent to stay on land, as that is the only way the agent can reach new areas on the map. When the water current is random, the exploration objective can no longer incentivize the agent to stay on land. This is because the exploration objective treats all 'new' regions of the environment the same. This is where the empowerment objective helps the agent explore meaningful regions on map by incentivizing to stay in regions where its actions can influence outcomes, i.e. explore areas of the environment that are not stochastic. This enables the agent to discover rewards faster, using which the agent can learn a useful extrinsic value function.

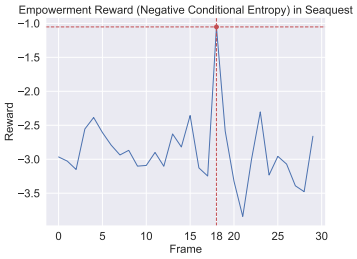
Figure 4a shows the episodic returns of the agent as training progresses in terms of number of frames observed in the environment. Without any intrinsic motivations, the DreamerV2 agent could not attain any reward in 1M frames of training. The Plan2Explore agent was able to discover the second island, however only converges to the maximum possible average reward after 900k frames, while our agent with the added empowerment objective converged in 600k frames of training.

Figure 3 illustrates the learned empowerment and exploration rewards at different positions on the BrideWalk environment map at different phases of training. As training progresses, the agent discovers more regions on the map, while at the same time learning intrinsic rewards in those regions. We observe that the agent learns a high empowerment reward on the islands and the bridge, and a low empowerment reward everywhere else. The exploration reward is high near the edges of the unexplored regions on the map. As training reaches 500k frames, the empowerment reward is still low on the goal island whereas the exploration reward is high, because the agent is yet to make more visits to this region. As training reaches 700k frames, the agent has visited the goal island often, and now the exploration reward no longer incentivizes the agent to be on either the bridge or the island, as those regions are already explored. However, the empowerment reward incentivizes the agent to visit the goal island, as there the agent obtains more deterministic outcomes of its actions.

4.2 Atari

We performed experiments on 5 popular Atari games - Enduro, Pong, Q*bert, Seaquest, and Space Invaders. Table 1 shows the average rewards receive by our agent on these games, along with comparison to baselines.

In the Seaquest game, the player uses a submarine to shoot down elements coming from both the left and right end of the frame. We observed that the empowerment objective helps the agent learn to keep the submarine close to the center of the frame as that enables it to create most affect on its movement through its actions. On the other hand, we observed that the DreamerV2 policy tried to keep the submarine close to either edge of the frame as that gives the agent reward faster, but was a sub-optimal strategy as the player would get killed faster, leading to shorter episodes and low episodic rewards. We illustrate the empowerment reward of one experience trajectory in Figure 5, where we observe that the empowerment reward peaks when the submarine moves down from the surface into



(a) Empowerment rewards.

(b) Frames 17, 18, 19.

Figure 5: (Left) Empowerment rewards received by the agent during a 30-frame segment of the game Seaquest. (Right) Game screenshots around the time when the agent achieved maximum empowerment reward, which is when the submarine moves away from the surface into the playing area.

the water as there it can predict the outcomes of its actions. We also show the average rewards as training progresses in Figure 4b.

In Space Invaders, the player aims to shoot down aliens arranged as a 2D grid before they reach the player's cannon. We observed that while the DreamerV2 policy destroyed aliens row-wise, our agent with the added empowerment objective destroyed them column-wise, as that required shooting from a single location resulting in more controllable outcomes for the agent. The player could also hide its cannon behind stationary bunkers, which are destroyed both by the player's and aliens' fire. Both our agent and DreamerV2 learned to use the bunkers to shield themselves while not shooting at the same from below, which the Plan2Explore policy could not learn in the finite training horizon, resulting in lower returns for that policy.

Policies with both extrinsic and intrinsic motivations learned to beat the Pong simulator by bouncing the ball off the floor, with the Plan2Explore policy converging the fastest. We also observed that normalizing rewards caused the actor and the policy to start to diverge after training for 3M frames, which is why we skipped reward normalization when training on Pong. In Enduro, the exploration objective seemed to worsen the player's performance, and our best performing model was one with extrinsic rewards from the MDP and the empowerment objective only.

5 Conclusion

In this paper, we addressed the problem of directed exploration in partially-observable environments. We proposed to use an empowerment objective in conjunction with an exploration objective as intrinsic motivation for exploring the environment. Such an objective ensures exploration in regions of the environment where the agent can maximally influence changes in the visited states and eventually solve tasks to collect rewards.

We also showcased a new simulation environment called BridgeWalk, where the player spawns on an island and is supposed to cross a single-step-wide bridge to another island to collect reward. The player's movement stays deterministic on land and becomes random in water. We showed that an exploration objective is critical for the agent to discover the goal state in this environment where the reward is extremely sparse, and that a combination of empowerment and exploration objectives helps the agent discover the goal state much faster than when just exploration is used. To show that our method generalizes well to complex tasks, we conducted experiments on five Atari games, on which our agent outperformed baselines on majority of the games.

References

J. Achiam, H. Edwards, D. Amodei, and P. Abbeel. Variational option discovery algorithms. *CoRR*, abs/1807.10299, 2018. URL http://arxiv.org/abs/1807.10299.

A. P. Badia, B. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, Z. D. Guo, and C. Blundell. Agent57: Outperforming the atari human benchmark. *CoRR*, abs/2003.13350, 2020. URL https://arxiv.org/abs/2003.13350.

Table 1: Quantitative comparison of RL agents after training on 10M frames. We report the average over 3 random seeds and one standard deviation.

	Enduro	Pong	Q*bert	Seaquest	Space Invaders
Random	0	-20.4	157	110	179
DQN	592 ± 4	3 ± 9	1804 ± 957	333 ± 41	689 ± 7
DreamerV2	542 ± 467	17 ± 2	$\mathbf{5758 \pm 1796}$	1267 ± 372	583 ± 33
Plan2Explore	406 ± 16	18 ± 2	1225 ± 915	1627 ± 945	565 ± 22
Empower + Explore (Ours)	$\mathbf{632 \pm 908}$	$\mathbf{19 \pm 1}$	3450 ± 2103	$\mathbf{2600 \pm 1600}$	$\mathbf{747 \pm 220}$
Human	368	-3	18900	28010	3690

D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *CoRR*, abs/2006.13171, 2020. URL https://arxiv.org/abs/2006.13171.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *CoRR*, abs/1207.4708, 2012. URL http://arxiv.org/abs/1207.4708.

M. G. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. *CoRR*, abs/1606.01868, 2016. URL http://arxiv.org/abs/1606.01868.

Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Y. Burda, H. Edwards, A. J. Storkey, and O. Klimov. Exploration by random network distillation. *CoRR*, abs/1810.12894, 2018. URL http://arxiv.org/abs/1810.12894.

D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *CoRR*, abs/2007.00643, 2020. URL https://arxiv.org/abs/2007.00643.

D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv: Learning*, 2016.

E. Denton and R. Fergus. Stochastic video generation with a learned prior. *CoRR*, abs/1802.07687, 2018. URL http://arxiv.org/abs/1802.07687.

B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. *CoRR*, abs/1802.06070, 2018. URL http://arxiv.org/abs/1802.06070.

K. Gregor, D. J. Rezende, and D. Wierstra. Variational intrinsic control. *CoRR*, abs/1611.07507, 2016. URL http://arxiv.org/abs/1611.07507.

D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pages 2555–2565, 2019a.

D. Hafner, T. P. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. *CoRR*, abs/1912.01603, 2019b. URL http://arxiv.org/abs/1912.01603.

D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. *CoRR*, abs/2010.02193, 2020. URL https://arxiv.org/abs/2010.02193.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

A. Klyubin, D. Polani, and C. Nehaniv. Empowerment: a universal agent-centric measure of control. 1:128–135 Vol.1, 2005. doi: 10.1109/CEC.2005.1554676.

V. Konda and J. Tsitsiklis. Actor-critic algorithms. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *CoRR*, abs/1804.01523, 2018. URL http://arxiv.org/abs/1804.01523.

J. Lehman and K. O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, page 211–218, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450305570. doi: 10.1145/2001576.2001606. URL https://doi.org/10.1145/2001576.2001606.

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. URL http://arxiv.org/abs/1312.5602.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

S. Mohamed and D. J. Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. 2015. doi: 10.48550/ARXIV.1509.08731. URL https://arxiv.org/abs/1509.08731.

A. Y. Ng, D. Harada, and S. J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606122.

V. Saxena, J. Ba, and D. Hafner. Clockwork variational autoencoders. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29246–29257. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/f490d0af974fedf90cb0f1edce8e3dd5-Paper.pdf.

R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. Planning to explore via self-supervised world models. *CoRR*, abs/2005.05960, 2020. URL https://arxiv.org/abs/2005.05960.

P. Shyam, W. Jaskowski, and F. Gomez. Model-based active exploration. *CoRR*, abs/1810.12162, 2018. URL http://arxiv.org/abs/1810.12162.

S. Singh, A. G. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, page 1281–1288, Cambridge, MA, USA, 2004. MIT Press.

C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 3745–3753, USA, 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9. URL http://dl.acm.org/citation.cfm?id=3157382.3157516.

N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *CoRR*, abs/1502.04681, 2015. URL http://arxiv.org/abs/1502.04681.

R. S. Sutton. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, 1984. AAI8410337.

A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. X. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra. Habitat 2.0: Training home assistants to rearrange their habitat. *CoRR*, abs/2106.14405, 2021. URL https://arxiv.org/abs/2106.14405.

C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, 2016.

W. F. Whitney, M. Bloesch, J. T. Springenberg, A. Abdolmaleki, and M. A. Riedmiller. Rethinking exploration for sample-efficient policy learning. *CoRR*, abs/2101.09458, 2021. URL https://arxiv.org/abs/2101.09458.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

S. Zhao, J. Song, and S. Ermon. Learning hierarchical features from generative models. *CoRR*, abs/1702.08396, 2017. URL http://arxiv.org/abs/1702.08396.