Experimental Evidence that Empowerment May Drive Exploration in Sparse-Reward Environments

Francesco Massari
Swarthmore College
Swarthmore, USA
rmassar1@swarthmore.edu

Martin Biehl
Araya Inc.
Tokyo, Japan
martin@araya.org

Lisa Meeden
Department of Computer Science
Swarthmore College
Swarthmore, USA
meeden@cs.swarthmore.edu

Ryota Kanai
Araya Inc.
Tokyo, Japan
kanair@araya.org

Abstract—**Reinforcement Learning (RL) is known to be often unsuccessful in environments with sparse extrinsic rewards. A possible countermeasure is to endow RL agents with an intrinsic reward function, or 'intrinsic motivation', which rewards the agent based on certain features of the current sensor state. An intrinsic reward function based on the principle of empowerment assigns rewards proportional to the amount of control the agent has over its own sensors. We implemented a variation on a recently proposed intrinsically motivated agent, which we refer to as the 'curious' agent, and an empowerment-inspired agent. The former leverages sensor state encoding with a variational autoencoder, while the latter predicts the next sensor state via a variational information bottleneck. We compared the performance of both agents to that of an advantage actor-critic baseline in four sparse reward grid worlds. Both the empowerment agent and its curious competitor seem to benefit to similar extents from their intrinsic rewards. This provides some experimental support to the conjecture that empowerment can be used to drive exploration.**

Index Terms—**Advantage Actor-Critic, Empowerment, Information Bottleneck, Information Gain, Intrinsic Motivation, Reinforcement Learning, Variational Autoencoder**

I. INTRODUCTION

In Reinforcement Learning (RL), relying on feedback from the environment is effective, if rewards are readily dispensed after every action, but many problems offer only sparse guidance of such form. One way to solve this issue is to complement the extrinsic reward signal, which is received from the environment, with an intrinsic reward signal. The agent is endowed with an internal function that takes as input the current sensor state and rewards the agent according to how desirable it is from an intrinsic perspective. This internal reward thus functions as an *intrinsic motivation* [1], [2].

Since the key to success in sparse reward settings is exploration [3], the literature on intrinsic motivations features various approaches to drive exploratory behavior [1], [2]. One concept that has been proposed as an intrinsic motivation is *empowerment* [4], [5], which makes the agent maximize the control it has over its future sensor states.

In [2], it was noted that previous empowerment implementations had assumed accurate internal models of the environment as given and it was suggested that empowerment does not motivate the development of such models through exploration, because empowerment maximization does not imply the maximization of *information gain*. Information gain, (also known as *Bayesian surprise* [6], which is maximized by *knowledge seeking* agents [7]) directly measures improvements of the agent's internal model of the environment. However, it was later speculated in [5] that, in order to control its future sensor states, the agent has to accurately model the environment and that, therefore, empowerment maximization could also drive the improvement of such models, thereby leading to exploration. This would mean that empowerment maximization should be similarly useful as an exploration drive as information gain.

We here tested an agent that maximizes an empowerment-inspired intrinsic reward and one that maximizes an information-gain-inspired intrinsic reward. We find some promising evidence that the empowerment-inspired intrinsic reward also motivates exploration. However, further research is needed to establish whether, when it comes to motivating exploration, empowerment can be seen as an alternative to information gain.

II. BACKGROUND

A. Partially Observable Markov Decision Process

For the present work, learning is conceptualized as a *Partially Observable Markov Decision Process*, which is structured into a sequence of discrete time steps $t = 0, 1, 2, 3, \ldots$, later referred to as 'frames'. At every such time step, the agent senses a part $s_t \in \mathcal{S}$ of the environment state $e_t \in \mathcal{E}$. It then performs an action $a_t \in \mathcal{A}$, thus in turn affecting the next environment state e_{t+1}. The environment responds by supplying the agent with a new sensor state s_{t+1}, as well as an extrinsic reward $r_{t+1}^{ex} \in \mathcal{R} \subset \mathbb{R}$ [3].

B. Intrinsic Motivation

An intrinsic reward function takes in the current sensor state s_t and features of the agent's internal architecture and returns a real value r_t^{in}, which together with the extrinsic reward r_t^{ex} makes up the total reward $r_t = r_t^{in} + r_t^{ex}$ [2]. It is possible

to design this intrinsic reward function such that it encourages exploratory behavior [1], [8].

C. Empowerment

Empowerment seeks to quantify the level of control an agent has over its environment. The central idea is that of viewing the environment as an information channel between actions and future sensor states. Via its actions, an agent can influence its environment, which in turn determines the agent's next sensor state. It follows that having control over one's future sensor states requires having some control over the environment. Note that the influence an agent has over its environment only contributes to its empowerment in so far as it is observable by the agent. Empowerment then quantifies this control as the channel capacity (in state e_t) from actions A_t to future sensor states S_{t+1}, i.e. the maximum amount of information that *could* be transmitted from actions to future sensor states [4], [5]:

$$\mathfrak{E}(e_t) = \max_{p(A_t)} I(A_t; S_{t+1}|e_t). \tag{1}$$

Unfortunately, the channel capacity is notoriously expensive to compute and true empowerment often remains an ideal to be approximated in practice. There exist approximate solutions to this problem using neural networks [9], [10], but this paper will present a novel approach, similar to that published in [11], that offers the advantage of relative simplicity, while being similar in structure to a recently proposed implementation of intrinsic motivation [12].

D. Variational Autoencoders

Variational Autoencoders (VAE) are identity function approximators that map each input x to a distribution $q(Z|x)$ over latent codes z, which is parameterized by ϕ and conditioned on the input data. To restore the original data, a single code z is sampled from the distribution and subsequently decoded by another probability distribution $p(X|z)$ over possible inputs, which is parameterized by θ and conditioned on the sampled code [13].

The loss function, which is minimized during training, is the following [13]:

$$\mathcal{L}(x, \theta, \phi) = \mathbb{E}_{q_\phi(Z|x)}[-\log p_\theta(x|Z)] \tag{2}$$
$$+ KL[q_\phi(Z|x) \,||\, p(Z)].$$

III. METHODS

A. 'Curious' Agent

The information-gain-inspired agent that served both as a competitor and progenitor to the empowerment-inspired agent is itself based on an agent presented in [12] by Klissarov et al. In the following, we will refer to this agent as the *curious* agent. It is trained using the standard advantage actor-critic (A2C) algorithm (see Algorithm 1), which we adapted from the 'Torch_AC' package [14]. Both the critic, as well as the actor are implemented using fully connected neural networks.

The main modification of the standard A2C algorithm was the introduction of a VAE, which is trained to compress the

Algorithm 1: The A2C learning algorithm with variational state encoding for intrinsic motivation. The only difference to standard A2C training is that, every N frames ($N = 128$, in our case), a VAE is trained on the previously collected sensory data. At *every* frame, the VAE tries to compress the current sensor state s_t and the resulting loss $\mathcal{L}(s_t, \theta, \phi)$ is used as intrinsic reward. The pseudocode is adapted from [12], and was the main source to inform the implementation of all agents presented in this paper.

---

for *Episode* $= 0, 1, 2, \ldots$ **do**
    Initialize dataset $\mathcal{D}$ and insert s_0;
    for $t = 0, 1, 2, \ldots, T$ **do**
        take action a_t and observe next state s_{t+1} and
         extrinsic reward r^{ex}_{t+1};
        compute intrinsic reward $r^{in}_t = \mathcal{L}(s_t, \theta, \phi)$;
        store tuple $(s_{t+1}, a_t, r^{in}_{t+1}, r^{ex}_{t+1})$ in $\mathcal{D}$;
        if $mod(t, N)$ **then**
            train the actor and critic on return
            $G_t = \sum_t r^{ex}_t + \beta r^{in}_t$;
            train the VAE on the collected states s in $\mathcal{D}$;
            initialize dataset $\mathcal{D}$ and insert s_t in $\mathcal{D}$
        end
    end
end

---

incoming sensor states. All VAEs featured in the following set of experiments were adapted from [15].

After every 128 frames, the VAE, as well as the actor and critic networks, are trained on the collected data (see Algorithm 1). At every frame, the current sensor state is compressed and decoded by the VAE, yielding a loss in the form of (2). Similar to the approach in [16], which uses random network distillation, a new sensor state that is completely unlike anything the agent has perceived before will be comparatively difficult to compress, since the VAE has not been trained on similar states. Therefore, the loss can be interpreted as signaling 'novelty' and used as intrinsic reward:

$$r^{in}_t(s_t, \theta, \phi) = \mathcal{L}(s_t, \theta, \phi). \tag{3}$$

The resulting value for the intrinsic reward is first scaled by a factor, denoted β, and then added to the extrinsic reward to yield the total reward (see Algorithm 1). Klissarov et al. used only the KL term of the loss as intrinsic reward, due to its similarity to Bayesian surprise [6], [12], while we obtained better results with the present approach. To ensure the fairness of the comparison with the empowerment-inspired agent below, we chose to work with the version that performed best.

B. Empowerment Agent

The architecture of the curious agent above underwent a number of incremental changes that brought the intrinsic reward closer to signaling empowerment. The first step consisted in

turning the VAE into a predictive model. Instead of using one and the same sensor state as both input and target, the state at time step $t + 1$ served as a target vector, while the input was the sensor state at t. The VAE was thus turned into a *deep variational information bottleneck* [17], which we will henceforth refer to simply as a *predictor*.

When future sensor states are used as targets, (2) is turned into the following predictor loss function

$$\mathcal{L}_{\text{pred}}(x, x', \theta, \phi) = \mathbb{E}_{q_\phi(Z|x)}[-\log p_\theta(x'|Z)] \qquad (4)$$
$$+ KL[q_\phi(Z|x) \,||\, p(Z)],$$

where x refers to the input, x' to the target, while the other variables are defined as in (2). The second modification was to add a second predictor, which differs from the first one in only one aspect. Instead of decoding the future sensor state s_{t+1} from only the preceding state s_t, the second predictor takes in the preceding action a_t as additional input. This second predictor will subsequently be referred to as the *action-predictor*.

In a state of high empowerment, the action a_t should bear more information about the subsequent sensor state s_{t+1}. To obtain an estimate of how much information about the future the pair (s_t, a_t) contains, over and above the information available from only s_t, one can compare the predictive performance of both predictors [11]. We chose the loss function as a performance metric, since it is already implicitly used as such during training. We then defined the intrinsic reward, which we want to be correlated with empowerment, as follows:

$$r_t^{in} = \mathcal{L}_{\text{pred}}(s_t, s_{t+1}, \theta_2, \phi_2) - \mathcal{L}_{\text{pred}}((s_t, a_t), s_{t+1}, \theta_1, \phi_1). \quad (5)$$

In (5), the two terms refer to the losses resulting from the prediction of s_{t+1} by the action-predictor with parameters θ_1, ϕ_1 and predictor with parameters θ_2, ϕ_2. Both losses are calculated as defined in (4).

We now show that, under assumptions 1 through 3 below, the intrinsic reward of (5) can be related to empowerment. As shown in [17], the predictor loss of (4) approximates a lower bound L for the information bottleneck objective [17, Eq.3]

$$\mathcal{L}_{\text{pred}}(x, x', \theta, \phi) \approx L(x, x') \leq I(Z; X') - \alpha I(Z; X), \quad (6)$$

where x denotes the input, x' the target, Z the latent code, and α a Lagrange multiplier. Under assumptions 1 and 2 below, we get

$$r_t^{in} \approx I(Z; S_{t+1}) - \alpha I(Z; S_t) \qquad (7)$$
$$- I(Z; S_{t+1}) + \alpha I(Z; S_t, A_t).$$

The right side of (7) simplifies to

$$\alpha I(Z; A_t \mid S_t). \qquad (8)$$

Then, by maximizing (5), the actor also maximizes (8). Under assumption 3 below, it thereby also maximizes $I(S_{t+1}; A_t \mid S_t)$, which is a lower bound on the empowerment according to (1).

This argument rests on the following three assumptions:

1) The latent variables of the two predictors can be treated as the same random variable Z.
2) The difference of the losses in (5) approximates the difference between the actual information bottleneck objective functions for both predictors according to (6).
3) Maximizing (8) implies maximization of $I(S_{t+1}; A_t \mid S_t)$.

Assumptions 1 appears reasonable, since the latent variables for both predictors are of the same type and are used in the same way. Assumption 2 should be true at least when the bounds are similarly far away from the true values. Lastly, Assumption 3 should hold, if the encoder is forced to only inject information into Z that is relevant for predicting S_{t+1}, as is the case if Z's capacity is low enough.

Other than these changes, the training algorithm remained as it was for the curious agent (see Algorithm 1).

There are multiple implementations of empowerment-like reward functions in the skill discovery literature [18]–[20]. Replacing skills with mere actions in these publications could lead to an alternative to our approach. Note however, that the empowerment-like reward is not considered or evaluated as an exploration drive in these publications. Note also that the empowerment estimation method in [21] cannot be used for exploration, since the modelled channel form actions to sensors has a fixed noise term.

IV. EXPERIMENTS

A. Environments

All environments used to experiment with the two agents above, were taken from the 'MiniGrid' package [22]. The environments are simple grid worlds, which contain a number of simple objects that can be manipulated by the agent. The agent only directly perceives a square area with a side length of seven tiles, which is located mostly in front of it.
All environments offer sparse rewards of magnitude one when a single 'goal event' is reached, which always concludes the episode.

1) MultiRoom-N3-S4-v0: Here, the agent is placed in the first of three rooms of side length two. The three rooms are connected by doors, which are closed, but not locked. A goal tile is located in the third room and dispenses an extrinsic reward when reached by the agent. The relative locations of the three rooms is randomly reinitialized after each episode. We added this environment to the existing Minigrid-Multiroom configurations following the example of [12].

2) DoorKey-8x8-v0: This environment consists of two rooms separated by a wall, which taken together form a square of side length six. The agent is placed into the first room together with a key. It has to find the key, pick it up, carry it to a locked door in the wall, open the door and walk to a goal tile in the second room. It is only when it has reached the goal tile that it receives an extrinsic reward. The location of the wall and the door, the placement of agent and key in the first room, and the location of the goal tile in the second room vary randomly between episodes.

3) KeyCorridor-S3R1-v0: The agent is placed in the middle of a corridor of length five and has to turn in order to open a door, behind which it finds a key. It needs to use the key to open a second door on the other side of the corridor. A ball is located behind this second door. Picking it up yields an extrinsic reward of one.

B. Training

The framework for the structure of the training process, including the handling of model parameter storage, command-line prompts, and log files, was based on the 'RL-Starter-Files' package [23]. The curious, the empowerment-inspired, as well as the A2C baseline agents were trained on a certain number of frames per run, depending on the environment. For every environment, the network parameters were first manually optimized before performing ten independent training runs (see Appendix A).

V. RESULTS

The main metric of interest is the mean reward per episode, which is calculated using a sliding window over the last eight episodes. Each figure below plots the mean reward per episode, averaged over the ten independent runs, along with its standard deviation (shaded area).

For each agent and environment, the frame number at which the mean reward per episode first reached a value of one, was recorded, yielding ten values per agent. These values will subsequently be referred to as the *time-to-success* of an agent. If an agent never solves a task, the time-to-success is set to ten million frames. By analyzing the time-to-success data, the question of whether one agent learns more quickly than another can be statistically addressed using a one-sided t-test. A difference is considered significant if $p < 0.05$, where p denotes the p-value. The direction of the hypotheses was informed by the plots of the training progress. Descriptive statistics for all agents and environments are summarized in Table IV.

A. MultiRoom-N3-S4-v0

The time-to-success of the curious agent was significantly shorter compared to that of the A2C baseline and the empowerment-inspired agent. The empowerment-inspired agent also solved the task more quickly than the baseline (see Table I and Fig. 1).

	A2C	curious	power
curious	$t = 4.65$, $p < 0.001$	—	$t = 4.19$, $p < 0.001$
power	$t = 4.39$, $p < 0.001$	—	—

TABLE I: This table includes the t and p-values of the directed t-tests conducted on the time-to-success data from the MultiRoom-N3-S4-v0 environment. For any cell in the table, the alternative hypothesis is that the 'row-agent' learns significantly faster than 'column-agent'. 'Power' refers to the empowerment-inspired agent.
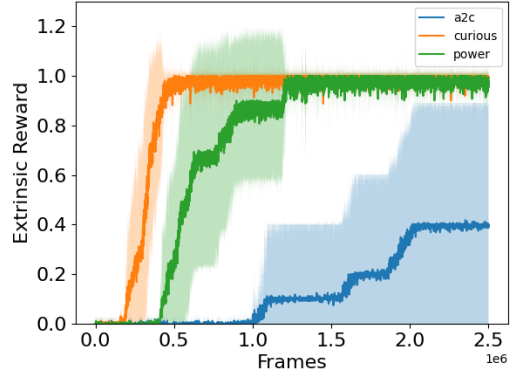

Fig. 1: Training results for the MultiRoom-N3-S4-v0 environment. The curious agent (orange) performs best, followed by the empowerment-inspired agent (green), here labelled 'power', and lastly the A2C baseline (blue).

B. DoorKey-8x8-v0

In terms of pure time-to-success there was no significant difference between the three agents (see Table II and Fig. 2).

Interestingly, closer inspection of the data reveals that for both the baseline and the empowerment agent, the third run was the only one during which they failed to solve the task, thus resulting in one outlier time-to-success value of $1e7$ for the two agents. Even for the curious agent, the third run was the only trial during which it solved the task only after more or less $2.5e6$ frames, instead of after around $1.5e6$ frames. After removing the data from the third run and repeating the statistical analysis, we found that the curious agent learned more quickly than both the A2C baseline and the empowerment-inspired agent, while there was no difference in terms of time-to-success between the empowerment-inspired agent and the A2C baseline (see Table II).
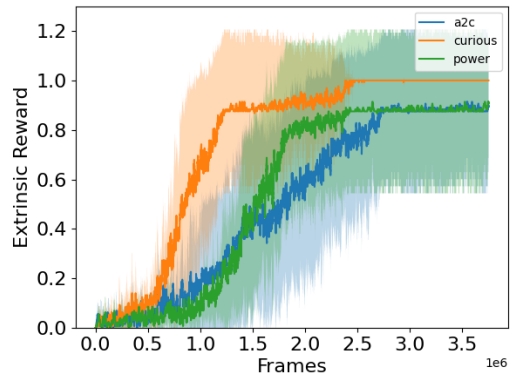

Fig. 2: Training results for the DoorKey-8x8-v0 environment. The curious agent (orange) seems to perform best, followed by the empowerment-inspired agent (green), here labelled 'power', and lastly the A2C baseline (blue). These intuitions are in part statistically confirmed, however only after outlier values from the third run are removed.

	A2C	curious	power
curious	$t = 1.68$, $p = 0.067$, $t* = 2.29$, $p* = 0.021$	—	$t = 1.52$, $p = 0.085$, $t* = 2.26$, $p* = 0.024$
power	$t = 0.11$, $p = 0.46$, $t* = 0.64$, $p* = 0.27$	—	—

TABLE II: This table includes the t and p-values of the directed t-tests conducted on the time-to-success data from the DoorKey-8x8-v0 environment. For any cell in the table, the alternative hypothesis is that the 'row-agent' learns significantly faster than the 'column-agent'. After the outlier time-to-success values from the third run were removed, the tests were repeated (results marked with a $*$).

C. KeyCorridor-S3R1-v0

The empowerment-inspired agent learned more quickly than both the A2C baseline and the curious agent. However, there was no significant difference in terms of time-to-success between the curious agent and the A2C baseline (see Table III and Fig. 3).

	A2C	curious	power
A2C	—	$t = 1.1$, $p = 0.28$	—
power	$t = 1.86$, $p = 0.041$	$t = 3.5$, $p = 0.001$	—

TABLE III: This table includes the t and p-values of the directed t-tests conducted on the time-to-success data from the KeyCorridor-S3R1-v0 environment. For any cell in the table, the alternative hypothesis is that the 'row-agent' learns significantly faster than the 'column-agent'.
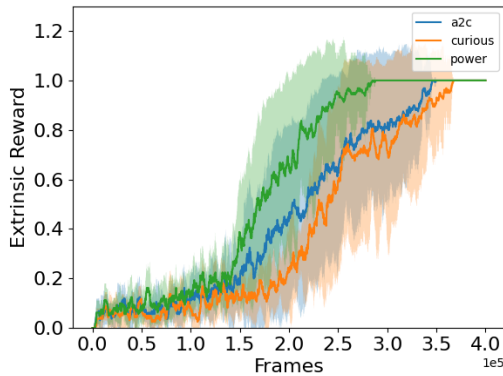

Fig. 3: Training results for the KeyCorridor-S3R1-v0 environment. As suggested by the plots and confirmed by statistical analyses of the time-to-successs, the A2C baseline (blue) and curious agent (orange) perform equally, while the empowerment-inspired agent (green), labelled 'power', learns more quickly than both.

		MultiRoom	DoorKey	KeyCorridor
A2C	M:	6.7e6	2.8e6	2.5e5
	SD:	4.3e6	2.6e6	0.7e5
curious	M:	3.3e5	1.2e6	2.8e5
	SD:	0.8e5	0.5e6	0.6e5
power	M:	6.7e5	2.7e6	2.1e5
	SD:	2.4e5	2.9e6	0.4e5

TABLE IV: Mean (M) and standard deviation (SD) of the time-to-success for all agents and environments.

VI. DISCUSSION

This work, albeit with a slightly modified approach, partly replicates the finding by Kilssarov et al. [12] that variational state encoding can be leveraged to build an intrinsically motivated agent, which outcompetes classical A2C. On the other hand, the benefit provided by the intrinsic reward depends on the task at hand and, in some cases, the surprise signal does not lead to significant improvements. Further, the alternative intrinsic reward function, inspired by the notion of empowerment, is similarly successful at counteracting sparsity of rewards and might therefore present a possible, scalable, empowerment approximation, which merits further investigation. This might be considered a hint that empowerment can drive exploration and that it may do so to a similar extent as an analogous intrinsic motivation inspired by the notion of information gain. However, significant assumptions and approximations are made in both agents and more research beyond these preliminary results is needed to establish whether empowerment maximization consistently leads to exploration.

APPENDIX A
NETWORK ARCHITECTURES AND TRAINING PARAMETERS

The observation arrays provided by the environment are converted into binary form, omitting the color information, before being forward propagated through the VAE/predictor. A given binary representation features a 1 at index $[i, j, k, l]$, if and only if the original observation features object k in state l, where k and l are integers, at position (i, j) and a 0 otherwise. The encoder and decoder of all VAEs/predictors feature one hidden layer and encoder and decoder of the same VAE/predictor always have the same number of hidden units (see Table V). All layers are densely connected. We used the ReLU activation function for the hidden layers, a linear activation function for the bottleneck layer, and a sigmoid for the output of the decoders. All models were trained with a learning rate of 0.001 and the *Adam* optimizer.

Agent	Hidden units	Latent dimensions	β
curious	512, 512, 265	256, 256, 128	2e-4, 1e-4, 1e-4
power	512, 32, 265	256, 16, 128	1e-4, 1e-4, 0.125e-4

TABLE V: Network parameters of the VAE/predictor models and β values for all environments. In every cell, the first value corresponds to the MultiRoom-N3-S4-v0 environment, followed by the values for the DoorKey-8x8-v0 and KeyCorridor-S3R1-v0 environments, respectively.

REFERENCES

[1] Oudeyer, P.-Y. and Kaplan, F., "What is intrinsic motivation? a typology of computational approaches," eng, *Frontiers in neurorobotics*, vol. 1, 2009, ISSN: 1662-5218.

[2] Schmidhuber, J., "Formal theory of creativity, fun, and intrinsic motivation (1990–2010)," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 230–247, 2010. DOI: 10.1109/TAMD.2010.2056368.

[3] Sutton, R. S. and Barto, A., *Reinforcement Learning An Introduction*, eng, ser. Adaptive computation and machine learning series. Cambridge, Mass: MIT Press, 2018, ISBN: 978026203924.

[4] Klyubin, A. S., Polani, D., and Nehaniv, C. L., "Empowerment: A universal agent-centric measure of control," in *2005 IEEE Congress on Evolutionary Computation*, vol. 1, 2005, 128–135 Vol.1.

[5] Salge, C., Glackin, C., and Polani, D., "Empowerment–an introduction," in *Guided Self-Organization: Inception*, Springer, 2014, pp. 67–114.

[6] Itti, L. and Baldi, P. F., "Bayesian surprise attracts human attention," in *Advances in Neural Information Processing Systems 18*, Weiss, Y., Schölkopf, B., and Platt, J. C., Eds., MIT Press, 2006, pp. 547–554. [Online]. Available: http://papers.nips.cc/paper/2822-bayesian-surprise-attracts-human-attention.pdf.

[7] Orseau, L., Lattimore, T., and Hutter, M., "Universal knowledge-seeking agents for stochastic environments," in *Algorithmic Learning Theory*, Jain, S., Munos, R., Stephan, F., and Zeugmann, T., Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 158–172, ISBN: 978-3-642-40935-6.

[8] Schmidhuber, J., "Curious model-building control systems," in *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, 1991, 1458–1463 vol.2.

[9] Binas, J., Ozair, S., and Bengio, Y., "The journey is the reward: Unsupervised learning of influential trajectories," *CoRR*, vol. abs/1905.09334, 2019. arXiv: 1905.09334. [Online]. Available: http://arxiv.org/abs/1905.09334.

[10] Mohamed, S. and Jimenez Rezende, D., "Variational information maximisation for intrinsically motivated reinforcement learning," in *Advances in Neural Information Processing Systems*, Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., Eds., vol. 28, Curran Associates, Inc., 2015, pp. 2125–2133. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/e00406144c1e7e35240afed70f34166a-Paper.pdf.

[11] Abril, I. M. de and Kanai, R., "A unified strategy for implementing curiosity and empowerment driven reinforcement learning," *CoRR*, vol. abs/1806.06505, 2018. arXiv: 1806.06505. [Online]. Available: http://arxiv.org/abs/1806.06505.

[12] Klissarov, M., Islam, R., Khetarpal, K., and Precup, D., "Variational state encoding as intrinsic motivation in reinforcement learning," in *Task-Agnostic Reinforcement Learning Workshop at Proceedings of the International Conference on Learning Representations*, 2019.

[13] Kingma, D. P. and Welling, M., "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[14] Willems, L., *Torch_ac*, https://github.com/lcswillems/torch-ac, 2019.

[15] Tiao, L., *A tutorial on variational autoencoders with a concise keras implementation*, 2020. [Online]. Available: https://tiao.io/post/tutorial-on-variational-autoencoders-with-a-concise-keras-implementation/.

[16] Burda, Y., Edwards, H., Storkey, A., and Klimov, O., "Exploration by Random Network Distillation," *arXiv:1810.12894 [cs, stat]*, Oct. 2018, arXiv: 1810.12894. [Online]. Available: http://arxiv.org/abs/1810.12894 (visited on 11/06/2018).

[17] Alemi, A., Fischer, I., Dillon, J., and Murphy, K., "Deep variational information bottleneck," in *ICLR*, 2017. [Online]. Available: https://arxiv.org/abs/1612.00410.

[18] Gregor, K., Rezende, D. J., and Wierstra, D., "Variational Intrinsic Control," *arXiv:1611.07507 [cs]*, Nov. 2016, arXiv: 1611.07507. [Online]. Available: http://arxiv.org/abs/1611.07507 (visited on 08/07/2017).

[19] Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S., "Diversity is All You Need: Learning Skills without a Reward Function," *arXiv:1802.06070 [cs]*, Oct. 2018, arXiv: 1802.06070. [Online]. Available: http://arxiv.org/abs/1802.06070 (visited on 06/05/2021).

[20] Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K., "Dynamics-Aware Unsupervised Discovery of Skills," en, *arXiv:1907.01657 [cs, stat]*, Feb. 2020, arXiv: 1907.01657. [Online]. Available: http://arxiv.org/abs/1907.01657 (visited on 06/05/2021).

[21] Zhao, R., Tiomkin, S., and Abbeel, P., "Learning Efficient Representation for Intrinsic Motivation," *arXiv:1912.02624 [cs, stat]*, Aug. 2020, arXiv: 1912.02624. [Online]. Available: http://arxiv.org/abs/1912.02624 (visited on 06/05/2021).

[22] Chevalier-Boisvert, M., Willems, L., and Pal, S., *Minimalistic gridworld environment for openai gym*, https://github.com/maximecb/gym-minigrid, 2018.

[23] Willems, L., *Rl-starter-files*, https://github.com/lcswillems/rl-starter-files, 2018.