*The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019*

# Curiosity Driven by Self Capability Prediction

Nicolas Bougie[*1*2] Ryutaro Ichise[*2*1]

[*1] Sokendai, The Graduate University for Advanced Studies
[*2] National Institute of Informatics

Reinforcement learning is a powerful method to solve tasks using a reward signal; however, it struggles in sparse reward scenarios. One solution to this problem is the use of reward shaping but, it requires complicated human engineering in complex environments. Instead, our solution relies on exploration driven by curiosity. In this paper, we formulate the curiosity as the ability of the agent to predict its knowledge about the task. The prediction is based on the combination of intermediate goals and deep learning. Our end-to-end method scales to high-dimensional state spaces such as images. As proof-of-concept, we present a preliminary implementation of our algorithm using only raw pixels as input.

## 1. Introduction

Reinforcement learning (RL) methods have led to remarkable successes in a wide variety of tasks. RL can be used to train an algorithm to learn policies by optimizing a reward function. For instance, they have been used in autonomous vehicle control [Abbeel et al., 2007] or robotic control [Levine et al., 2016]. Another significant technique is the combination of deep neural networks and Q-learning, resulting in "Deep Q-Learning" (DQN) [Mnih et al., 2013], able to achieve human performance on many tasks including Atari video games [Bellemare et al., 2015]. However, in many real-world tasks, rewards are sparse or poorly defined, which entails that they learn slowly.

In order to guide the agent, an additional intrinsic signal can be provided to the agent. Multiple techniques have been tested. For example, Racaniere et al., base the exploration of the agent on the surprise - the ability of the agent to predict future [Racanière et al., 2017]. Pathak et al., estimate the surprise of the agent by predicting the consequences of the actions of the agent on the environment [Pathak et al., 2017]. Namely, they use an inverse model and the prediction error as the intrinsic reward. Another attempt aims to predict the features of a fixed random neural network on the observation of the agent [Burda et al., 2018]. Nevertheless, the low sample efficiency doesn't show clearly how to adapt this method to large scale tasks.

We propose an alternative solution to the curiosity mechanism by defining the exploration bonus as the capability of the agent to predict the sub-tasks that it masters - the agent learns to predict its own capabilities. We introduce the idea of goals to automatically decompose a task into several easier sub-tasks. Namely, given the current observation, the agent learns to predict which intermediate goals it masters. By acquiring knowledge about its abilities, we can improve exploration by forcing the agent to explore unknown parts of the environment. Our method relies on two deep neural networks: one to embedded the states and goals and the other one to predict the capabilities of the agent. In order to measure the distance between a goal and an observation, we base the goals and states representation on a latent variable model, a variational autoencoder [Kingma and Welling, 2014]. In the preliminary implementation, our architecture can learn policies in large continuous states spaces with sparse rewards. Note that our agent can learn policies from raw pixels without any supervision.

Contact: Nicolas Bougie, Sokendai, The Graduate University for Advanced Studies, Tokyo, Japan, +81-3-4212-2000, nicolas-bougie@nii.ac.jp

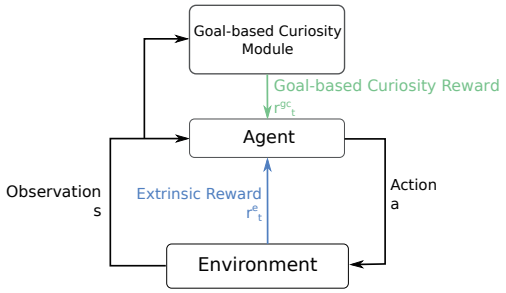


Figure 1: The agent in a state $s$ interacts with the environment by performing an action $a$ and, receives an extrinsic reward $r^e$. A policy $\pi(s_t;\theta_P)$ is trained to optimize the sum of $r^e$ and $r^{gc}$. The intrinsic reward $r^{gc}$ is generated by the goal-based curiosity module to favor the exploration of novel states.

## 2. Method

### 2.1 Curiosity as Reward Signal

Training an agent in a sparse reward environment is challenging since the agent generally receives no reward or a negative reward. We can introduce a new bonus to encourage the agent to explore sparse reward scenarios. In addition to the extrinsic rewards $r^e$ of the environment, we introduce a goal-based curiosity reward signal $r^{gc}$ (Figure 1). At time $t$ the agent receives the sum of these two rewards $r_t = r^e_t + r^{gc}_t$. To encourage the agent to explore the environment, we design $r^{gc}$ to be higher in novel states than in frequently visited states.

The policy $\pi(s_t;\theta_P)$ is represented by a deep neural net-

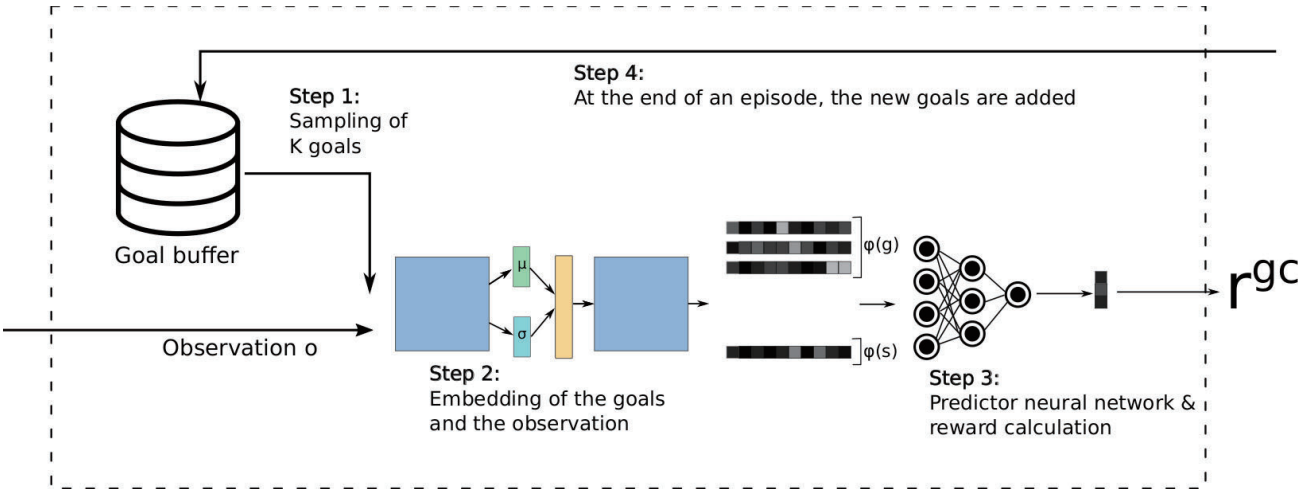


Figure 2: Goal-based curiosity module. The module takes as input an observation $o$ and, at the beginning of every episode randomly samples multiple goals. The goals and the observation are embedded during step 2, $\phi(g)$ and $\phi(s)$ respectively. Step 3 predicts the probability that each goal is mastered and given this vector of probabilities, calculates the new reward signal $r^{gc}$. A the end of each episode, the new goals are added to the goal buffer based on the experienced states.

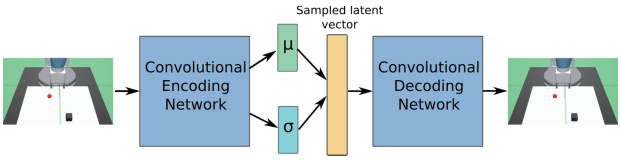


Figure 3: Variational Auto Encoder structure. The input image is passed through an encoder network which outputs the parameters $\mu$ and $\sigma$ of a multivariate Gaussian distribution. A latent vector is sampled and the decoder network decodes it into an image.

works. Its parameters $\theta_P$ are optimized to maximize the following equation:

$$\max_{\theta_P} \mathbb{E}_{\pi(s_t;\theta_P)}[\sum_t r_t] \tag{1}$$

In this work, we use twin delayed deep deterministic policy gradients (TD2) [Silver et al., 2014] as policy learning method. Our main contribution is to design a new exploration mechanism, the goal-based curiosity module that we describe in the following section. Given the current observation, the module generates a goal-based curiosity reward signal $r^{gc}$.

## 2.2 Goal-based Curiosity Module

We based the goal-based curiosity module (GCM) on the following intuition. If the agent can predict whether or not it can achieve a goal given an observation then, we can reward more the agent when the uncertainty to solve it is high - to force the agent to explore novel states.

In sparse reward environments, reaching the final goal may be infrequent, entailing that for most of the episodes the agent only experiences failures. Therefore, training a probabilistic model for predicting if the final goal is mastered is highly inaccurate. Instead, we propose to estimate if the agent can solve multiple intermediate goals. Since these goals are easier to master (to reach), the estimation becomes more accurate while providing information about the agent's knowledge. Our model can be trained using multiple goals which produces a vector of probabilities. The method to select the goal is explained in Section 2.2.1.

In details, the GCM is an end-to-end module. It takes as input the current observation and produces $r^{gc}$. The algorithm can be broken down in fours parts (Figure 2). First, at the beginning of an episode, $K$ goals are randomly sampled with $1 \leq K \in \mathbb{N}$. Second, at every step, the goals and the current observation are embedded by a variational autoencoder $f : O \rightarrow \mathbb{R}^n$ (Figure 3). Third, the agent predicts the probability that each goal can be achieved. Our implementation relies on a deep predictor neural network which predicts the probability that a goal can be reached $\hat{f} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow [0-1]$. Finally, at the end of the episode, the GCM is updated according to the experienced observations and the predictor neural network is retrained to fit with the new knowledge of the agent.

To produce the reward signal and improve exploration, we design the reward to be higher in novel states. Namely, we take advantage of uncertainty given the probabilities that the goals are mastered by the agent. We give details about the reward calculation is Section 2.2.2.

### 2.2.1 Goals

We define the goals $g \in G$ as $f_g : S \rightarrow \{0, 1\}$ that defines if the goal is achieved by the agent. In the case that a goal $g$ is solved in a state $s$, $f_g(s) = 1$. In order to keep a consistent representation, we suppose the goals as $G = \mathbb{R}^n$. Note that we assume the goal space $G$ to be the same as the state space $S$.

At each iteration, a sub-sets $g'$ of goals latent $\phi(g)$ are sampled given a distribution function $f_p$:

$$g' = \{\phi(g) \sim f_p(g)\} \text{ sample K goals} \in G \tag{2}$$

In the current implementation, the probability of sampling a goal $f_p(g)$ is uniform for all the goals. In future work, we anticipate more complex distributions to take into about the difficulties of the goals.

During training, the goals aim to provide additional feedback to the agent to improve exploration. At the end of an episode, we add a mechanism to further enable sample-efficient learning. In addition to the state reached at the end of the episode, we artificially generate new goals by randomly selecting states visited during the episode. The new goals are stored in the replay buffer that is used to train an RL algorithm.

#### 2.2.2 Reward Calculation

At every time step, the deep predictor neural network outputs the probability that the active goals are mastered by the agent $g_{active} = \langle p(g_1, ..., g_K) \rangle$. Given these predictions we define the goal-based curiosity reward:

$$r^{gc} = \delta \times g(\langle \alpha \rangle - \langle p(g_1, ..., g_K \rangle) \tag{3}$$

with $g$ the function mapping the probabilities to the reward, the parameters $\delta$ the scale of the new reward, and $\langle \alpha \rangle$ the sign of the new reward. In the current implementation, we use $\alpha = \langle 1.0 \rangle$ a uniform vector of 1 and $g = \max()$.

In other words, predicting that the agent doesn't master a goal will result in a higher curiosity-based reward. One issue with the combination of extrinsic reward and curiosity-based reward is the scaling of the reward which may vary between the tasks. In order to mitigate this scaling problem, we normalize the curiosity-based reward:

$$r_t^{gc} = \frac{r_t^{gc}}{\sigma(R_{gc})} \tag{4}$$

with $\sigma(R_e)$ the standard deviations of the curiosity-based reward returns.

## 3. Conclusion

This paper introduces a new mechanism for generating curiosity based rewards on the idea of predicting the capabilities of the agent. This allows our agent to learn policies in sparse reward environments without human engineering. Our model works in the latent space to generate a compact representation of the states and goals which are used by a deep neural network to predict the capabilities of the agents. By acquiring knowledge about itself, the agent can use its curiosity to explore unseen states of the environment. As a result, we can expect to solve a large set of tasks requiring more supervision than the extrinsic reward of the environment.

We are interested in testing our method on a set of tasks such as *MuJoCo*, or *Super Mario Bros*, two sparse reward environments. In the future, we are willing to introduce human feedback during the choice of the goals and improve the goal sampling method.

## References

[Abbeel et al., 2007] Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. In *Proceedings of Advances in Neural Information Processing Systems*, pages 1–8.

[Bellemare et al., 2015] Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2015). The arcade learning environment: an evaluation platform for general agents. In *Proceedings of the International Conference on Artificial Intelligence*, pages 4148–4152.

[Burda et al., 2018] Burda, Y., Edwards, H., Storkey, A. J., and Klimov, O. (2018). Exploration by random network distillation. *CoRR*, abs/1810.12894.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *International Conference on Learning Representations*.

[Levine et al., 2016] Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373.

[Mnih et al., 2013] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[Pathak et al., 2017] Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the International Conference on International Conference on Machine Learning*.

[Racanière et al., 2017] Racanière, S., Weber, T., Reichert, D., Buesing, L., Guez, A., Jimenez Rezende, D., Puigdomènech Badia, A., Vinyals, O., Heess, N., Li, Y., Pascanu, R., Battaglia, P., Hassabis, D., Silver, D., and Wierstra, D. (2017). Imagination-augmented agents for deep reinforcement learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Proceedings of Advances in Neural Information Processing Systems*, pages 5690–5701. Curran Associates, Inc.

[Silver et al., 2014] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *Proceedings of the International Conference on International Conference on Machine Learning*, pages I–387–I–395.