Domain Exploration Through Artificial Curiosity

Matt Raymond
mattrmd@umich.edu

Ted Sender
tsender@umich.edu

Yabin Dong
dyabin@umich.edu

Aravind Mantravadi
amantrav@umich.edu

Abstract

Despite the many advancements in autonomy and artificial intelligence, current autonomous systems do not have the insight necessary for exploration. This paper attempts to address this deficiency by applying the established concept of artificial curiosity to a planetary rover. We propose a notion of *explorational value* as the average variance of an agent's sensory inputs over a given time interval, and a notion of *novelty* as the reconstruction loss of those sensory inputs through a compressor-decompressor. We simulate a 2D, curious, unsupervised agent on a surface albedo map of Mars, and test the hypothesis that an agent's movements governed by local novelty, in which the compressor-decompressor is a convolutional autoencoder, correlate with a greater explorational value than that of an agent moving in a purely linear or random manner.

1 Introduction

Curiosity is the driving force behind many of the world's greatest discoveries. It's what makes us want to explore and go places no person has ever gone before. It's the mysteriousness of nature that leads people to dedicate their lives to science. However, we've come to a point in human history where we see far more of the universe than we can ever explore. We can detect stars billions of light-years away, but can barely make it to our Moon. If the march scientific exploration is to continue, we need a surrogate scientist who can travel these distances and explore in our name.

In the past few decades, remote-controlled probes have been our best attempt, giving hundreds of scientists a first-hand glimpse of the surface of distant worlds. However, this is an imperfect solution. The vast distances of space cause massive communications delays, making exploration more difficult the farther we travel from earth [1]. Remote control is insufficient, and our current autonomous systems are ill-equipped for this kind of task [2, 3]. Instead, we need an autonomous explorer who can perform these tasks in our stead. Unfortunately, robots don't make very good scientists. They don't care about the differences between two rocks, whether they travel north or south, or whether they're seeing something for the first or thousandth time. But what if they did? This is the motivation behind artificial curiosity: creating a machine that thinks like a scientist.

2 Problem Definition

Our main goal is to develop an algorithm that follows the most explorationally advantageous path, or the path that will lead to the greatest scientific benefit. But what makes one path more explorationally advantageous than another? Although not a rule, it is often the case that discoveries are made when a scientist investigates something that is not well defined, or not well understood. For an agent, this could be a set of observations that fluctuate according to some unknown pattern. Thus, an agent should travel towards a region that it cannot easily predict. Due to the difficulty in computing a path's true explorational value, we propose an *explorational value heuristic (EVH)* as the average variance

of an agent's sensory inputs over a given time interval. We take as fact that a higher variance leads to a greater likelihood for scientific discovery. We further define *novelty* as the reconstruction loss of those sensory inputs through a fixed-length compressor-decompressor, in contrast to a variable-length compressor discussed in [4]. We hypothesize that a curious agent, an agent moving in the direction of the greatest (local) novelty, will also move in a path that has a high explorational value, similar to what is proposed by Graziano et al [4]. These definitions are adapted from concepts in [4].

2.1 Problem statement

The application of interest is a 2D rover (aka agent) moving on a 4K grayscale surface albedo map of Mars, see Figure 1, in which the rover can only move diagonally. Let a *curious agent* be an agent whose movements are governed by the local novelty of its camera data (it's only perception input). Let a *linear agent* be an agent that moves linearly in a mainly-deterministic manner (keeps moving in the same direction until it hits an edge, then it randomly changes direction). Let a *random agent* be an agent that moves randomly at each step. Let nov $\in \mathbb{R}_+$ denote the novelty value and let evh $\in \mathbb{R}_+$ denote the EVH, and $\mathbb{R}_+ = \{x | x \geq 0\}$.

The problem at hand is to:

1. Test the hypothesis that the curious agent's movements governed by local novelty correlate with a path of greater explorational value than that of two reference agents, the linear and random agents.

2. Determine the effectiveness of our proposed definitions for novelty and explorational value heuristic under the topic of artificial curiosity.

3 Related-work

Artificial curiosity is essentially a curiosity-driven learning that investigates the impacts of intrinsic motivations or rewards on agents in reinforcement learning [5, 6]. It has been used in applications as disparate as robotics control and video-game-playing agents [4–7]. Distinguished from conventional reinforcement learning, curiosity-driven learning defines an intrinsic reward function as the predicting loss of the next state given the current state [5]. For instance, in a study investigating agents playing SuperMario Bros, [5] intrinsic motivation or curiosity was represented by the prediction error of its own actions, and was learned from a self-supervised, inverse, dynamic model. Burda et al. [6] also studies curiosity-learning for play video games and they did not include any standard extrinsic reward.

For artificially-curious robotics specifically, Graziano et al. [4] review artificial curiosity for autonomous space exploration . To determine a useful curiosity measure, it is necessary for the agent to have a notion of "interestingness," which is formalized as instantaneous compression progress given new information [4, 8]. They show that compression progress measures the increase in compression ability based on revising previous history or making novel observations [4]. The "interestingness" of an observation then, is the degree to which it can improve compression progress.

As shown in related literature, prior state-of-the-art implementations usually include deep neural networks, autoencoders, or reinforcement learners to memorize the historical information, simulate curiosity-driven learning, make predictions, and translate those predictions to physical or virtual actions [4–7]. Particularly, convolutional autoencoders are one of the most prevalent methods for extracting features from image data [9], and have been used to perform image compression, segmentation, and medical image analysis [9–12]. Similar to the study of Schillaci et al. [7], we implement a convolutional autoencoder as a feature-aware, learning compressor. We test circular memory and priority queue memory, the results of which is presented in later sections. Additionally, to ensure that we have enough time for proper testing, we do not use a reinforcement learner. Instead, we simply use an argmax function to determine the direction of greatest novelty.

4 Basic approach

4.1 Simulation environment and base agent model

To keep matters simple, the environment the rover will operate in is an albedo map of Mars, shown in Figure 1, with a single measurable quantity (surface radiosity) represented by pixel intensity. The origin is at the top left of the image, as is standard in image processing. We choose this dataset because it has a low-dimensionality, is easy to visualize, and is unusually granular for open-access NASA data. Further, each agent is represented as a single pixel within the albedo map; although planets are spherical and the agent should be capable of circumnavigation, it will be confined to the boundaries of the image for simplicity (i.e. it cannot cross the right edge and re-appear on the left).

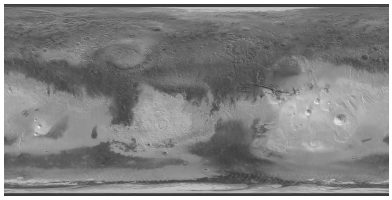

Figure 1: Thermal emission spectrometer (TES) global bolometric albedo map of Mars [13]

Perception model. We also provide the agent with a perception system by means of a simulated birds-eye view camera. The simulated camera will be represented by a fixed-size window centered at the agent's location. The size of this window will be twice the agent's field-of-view (FOV), which is defined as the distance the agent can see in a standard Cartesian direction (horizontally or vertically). We then take this full view and split it into four FOV $\times$ FOV regions (top left, top right, bottom left, bottom right) and denote these images as $I^{(1)}, I^{(2)}, I^{(3)}$, and $I^{(4)}$ respectively. We also denote the fullview image representing the entire patch of pixels the agent is able to see as

$$I^F = \begin{bmatrix} I^{(1)} & I^{(2)} \\ I^{(3)} & I^{(4)} \end{bmatrix}.$$

Agent motion. Agents can move along any of the diagonal directions (towards a camera region) with a pre-determined step size, and must be at least FOV pixels away from all edges of the image. These rules are based on how the curious agent operates, and for consistency all agents must adhere to the same set of rules.

4.2 Reference agents

The reference agents are built on simple rules, following opposite extremes (moving in a straight line or randomly). These reference agents act as a baseline from which we will compare our proposed curious agent.

Linear agent. The linear agent is a (mainly) deterministic agent. At time $t = 0$ the agent begins moving along a straight path in a random initial direction. When the agent's next step will bring it within FOV pixels from an edge, it randomly chooses a new direction (including reverse) to move it away from the edge and then continues its route. This process repeats until the simulation is ended.

Random agent. The random agent is a completely stochastic agent. $\forall t \in \mathbb{N}$, the direction at time t is a random choice from among the set of valid directions to keep the agent at least FOV pixels away from an edge. This process repeats until the simulation is ended.

4.3 Calculating the explorational value heuristic

The average variance of the agent's camera data from a given path is our proposed measure of the path's EVH. Given an agent's set of fullview images $\{I_1^F, I_2^F, \ldots, I_T^F\}$ over T time steps, we compute the EVH as follows. Denote $\bar{I}_{ij}^F = \sum_{t=1}^{T} I_{tij}^F$ as the mean value for the (i, j) pixel in the path.

Denote $\text{var}(I^F_{tij}) = \frac{1}{T}\sum_{t=1}^{T}(I^F_{tij} - \bar{I}^F_{ij})^2$ as the variance of the (i, j) pixel in I^F_t. Finally, compute $\text{evh} = \frac{1}{T \times \text{FOV}^2}\sum_{t=1}^{T}\sum_{i=1}^{\text{FOV}}\sum_{j=1}^{\text{FOV}}\text{var}(I^F_{tij})$.

5 Curious agent

The curious agent is the most involved agent because it is composed of a perception system and a brain. The perception system allows the agent to view its local surroundings and the brain performs all of the decision-making tasks based on the perceived inputs. The agent's perception system has already been detailed in Section 4.1.

5.1 The brain

The brain is most complex component because it contains the compressor-decompressor and further requires the existence of a memory module to select the information the agent should learn. The brain is also responsible for calculating the local novelty from the camera input and for determining which direction the agent should move in at each time step.

Compressor-decompressor. The compressor-decompressor, is represented by a custom-built convolutional autoencoder. We initially consider using VGG16, a pretrained convolutional neural network, to extract features for a simple autoencoder. However, preliminary experimentation shows that a generalized pretrained network is not appropriate for our specific domain. Instead, we implement a custom convolutional autoencoder that is symmetric with respect to the encoder and decoder, as is commonly done in practice [4, 14, 15]. Both the encoder and decoder contain 3 convolutional layers and 3 dense (fully-connected) layers. Because the agent's FOV is set at 64, the bottleneck layer in the autoencoder is set to a size of 256, which corresponds to 6.25% of the total image dimension. The full model definition is detailed in Appendix B, Table 5. The convolutional autoencoder is trained using the Adam optimizer, with the learning rate being a configurable hyperparameter. In addition, at each time step we train the network for a configurable number of epochs based on all the data currently in the memory module.

Memory module. The memory module is responsible for storing specific instances of past camera data which are then used to train the convolutional autoencoder. To more closely resemble a planetary rover, the memory module has a fixed length to simulate limited storage capacity, and this length is set as a configurable parameter. This begs the question, what is the most explorationally-advantageous method for allocating such a precious resource? We propose two types of memory modules, *priority memory* and *circular memory*, to compare the affect on what sensor data is most relevant for training the convolutional autoencoder. At time step t, each of the four regions from the camera input is evaluated for its novelty resulting in a pair called an *experience*, denoted $E^{(k)}_t = (\text{nov}^{(k)}_t, I^{(k)}_t)$ where $k \in \{1, 2, 3, 4\}$. The memory modules differ in what experiences they choose to keep or discard. Both modules also allow for random access so we can create datasets of all the stored images for training.

Priority memory works like a priority queue, where the priorities of the elements in the queue are inversely related to the novelty in each experience. This is intentional because the element with the highest priority, i.e. lowest novelty, can then be easily "popped" (removed). At time step t, if the priority queue is not full, then an experience $E^{(k)}_t$ is pushed to the priority queue. If the priority queue is full, then the experience is only pushed if its novelty is greater than that of the highest priority element. In this case, the highest priority element is popped and the new experience is pushed. We believe that regions with the highest novelty may compose a maximally-expressive subset of the visited domain, allowing for maximum information gain with minimal storage costs. One potential disadvantage of this approach is that some experiences in the priority queue may never be removed due to large initial novelties predicated by the initially-randomized compressor-decompressor. This could potentially bias the agent towards storing regions that are no longer relevant, or place undue emphasis on perfectly average regions.

Circular memory works like a circular buffer with a first-in-first-out (FIFO) approach. At time step t, if the buffer is not full, then an experience $E_t^{(k)}$ is pushed to the buffer. If the buffer is full, then the oldest experience is removed and the newest experience is pushed. This memory module poses as an alternative option for memory storage in that the agent may prefer to learn the most recent regions it visited instead of those with the largest novelty from the beginning of its adventure. It is possible that this will decrease the probability that the agent will reverse course and head back the way it came, which we hypothesize will increase its EVH.

Computing the novelty For our application, the novelty is the reconstruction loss of the agent's sensory inputs, in this case a camera region, through the convolutional autoencoder. We test two common loss functions to compute the novelty: Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Choosing the direction to move in. Initially, we propose that the agent should move in the direction of greatest local novelty at each time step. Since the problem of artificial curiosity is closely related to reinforcement learning, this approach may not result in a good balance between exploration and learning to facilitate a larger overall novelty, and a common solution is to introduce a small amount of randomness [16]. Accordingly, we propose an alternative method in which the agent occasionally moves in the *second* most novel direction. Let p be the probability that the agent moves in the direction of greatest local novelty and $1-p$ be the probability the agent moves in the direction of second greatest local novelty. It is expected that p will be relatively high, such as $p \geq 0.8$.

6 Experiments

Multiple experiments are conducted to help us assess our curious agent's performance in relation to that of the reference agents. In all experiments, a few high-level parameters remain constant. All agents are run for $T = 1000$ time steps with $FOV = 64$ and a movement step size of 8 pixels. A set of $N = 10$ starting positions is chosen randomly, and all agents are simulated at each starting position. For the curious agents, the memory module capacity is set to some multiple of 4 because there are 4 new camera regions at each time step. When training the curious agent's convolutional autoencoder, a new dataset is created at each step t by shuffling and batching the stored images in the memory module into groups of size 4. In all calculations involving images (training, computing novelty or EVH, etc.) all images are normalized to the range $[-1, 1]$. Further, the Adam optimizer for the convolutional autoencoder uses the default values from Tensorflow for the decay rates, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. All code is written in Python, and the neural network is developed and trained using Tensorflow. All experiments are executed in a docker environment on a computer with a 16GB NVIDIA Quadro RTX 5000 GPU.

During each run for an agent, several pieces of data are recorded and saved for later analysis:

- The agent's path (x_t, y_t) $\forall t \in \{1, \ldots, 1000\}$.
- The camera regions $(I_t^{(1)}, I_t^{(2)}, I_t^{(3)}, \text{and } I_t^{(4)})$ and the reconstructed regions $(\tilde{I}_t^{(1)}, \tilde{I}_t^{(2)}, \tilde{I}_t^{(3)}, \text{and } \tilde{I}_t^{(4)})$ $\forall t \in \{1, \ldots, 1000\}$.
- The novelty for each of the 4 camera regions $(\text{nov}_t^{(1)}, \text{nov}_t^{(2)}, \text{nov}_t^{(3)}, \text{nov}_t^{(4)})$ $\forall t \in \{1, \ldots, 1000\}$.
- The EVH for the agent's path.

6.1 Parameter study

There are a total of 5 different parameters that can be configured when simulating the curious agents: memory type, memory length, novelty function, number of training epochs per iteration, and learning rate. A parameter study is needed to determine if there is an optimal set of values that results in a higher EVH. Table 1 shows the different values for each parameter that are used in the study. We create a curious agent for every possible combination of parameters (one from each category), which in this case is 48 different curious agents. Note, in this parameter study we are only concerned with a curious agent that always moves in the direction of greatest local novelty (i.e. $p = 1$), which is the first proposed method of movement described in Section 5.1.

Table 1: Parameters used in the curious agent parameter study

Parameter	Value(s)
Memory Type	Priority, Circular
Memory Length	32, 64
Novelty Function	MSE, MAE
Num. Train Epochs Per Step	1, 2, 3
Learning Rate	0.0002, 0.0004

We perform two evaluation methods to determine what parameters are optimal.

Parameter-wise comparison. There are $M = 5$ parameters, let R_m denote the set of values for the mth parameter, and let $r \in R_m$ denote a parameter value. For a given $R_m \in \{R_1, \ldots, R_5\}$ and $\forall r \in R_m$, find all curious agents over all positions (a total of 480 agents) created with the parameter value r and compute the average EVH among those agents. We use the maximum average EVH for a given parameter as one method for finding the parameter's optimal value.

Agent-wise comparison. Another method is to compare the average EVH for each of the 48 curious agents. That is, for each of the 48 agents, compute its average EVH over the N positions. We posit that the agents with the best average EVH will coincide with the optimal parameters found in the parameter-wise evaluation.

6.2 Improvement study

Using the optimal parameters from Section 6.1, determine if the second method of movement for the curious agents (with $0 < p < 1$) can lead to higher EVH. The values of p we test are: 1.0, 0.95, 0.9, 0.85, and 0.8.

7 Results

7.1 Parameter study

Table 2 ranks agents based on their average EVH from the parameter-wise evaluation, and Table 6 in Appendix B ranks agents using the average EVH from the agent-wise evaluation (including the linear and random agents for comparison). We see that from the parameter-wise evaluation that the best parameters are: circular memory, memory length of 64, MSE novelty function, 3 training epochs per step, and a learning rate of 0.0004. From the agent-wise evaluation, the best agent we actually tested had the same parameters, except with a learning rate of 0.0002. Denote this best curious agent from the agent-wise evaluation as agent $C*$. From these results we can clearly conclude that the two parameters with the greatest effect were the memory type and number of epochs to train the convolutional autoencoder at each time step. The circular memory outperformed the priority memory, and having 3 training epochs per time step appeared to work the best. It is interesting to note that the linear agent came in second to agent $C*$, if ranking all types of agents together.

The paths from the linear, random, and curious agent $C*$ at the starting position $(574, 952)$ is shown in Figure 2. Agent $C*$'s path appears rather similar to the random agent's path. However, the curious agent moves in the direction of greatest local novelty (as indicated by its high EVH) while the random agent does not.

7.2 Improvement Study

In this study we actually test two curious agents with the parameters shown in Table 3. We observe that most agents preferred larger memory lengths and possibly the higher learning rate. So, in addition to testing agent $C*$, we test another agent with the same parameters, but increase the memory length to 100 and the learning rate to 0.0005. We also introduce the randomness into the agent's movement, as described in Section 5.1 by testing the values of p described in Section 6.2.

Table 2: Parameter-wise average EVH for parameter study

Parameter	Parameter Value	Average EVH
Memory Type	Priority	0.008237
	Circular	0.012328
Memory Length	32	0.009686
	64	0.010879
Novelty Function	MSE	0.020847
	MAE	0.020282
Training Epochs Per Step	1	0.013010
	2	0.013929
	3	0.014190
Learning Rate	0.0002	0.019740
	0.0004	0.021389

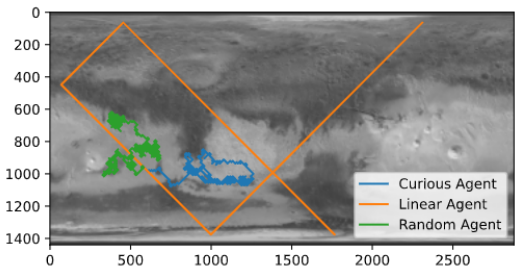

Figure 2: Comparison between an example of the $C*$, Linear, and Random agents for the parameter study, all starting at $x = 574, y = 952$.

Table 4 in Appendix B shows the average EVH for the agents. Increasing the memory length to 100 did not seem to have a marked improvement. The top two agents in this study had $p = 0.95$, one of which was agent $C*$ and the other agent had a memory length of 100 with learning rate of 0.0005. We see that yet again, agent $C*$ performs quite well. This suggests that occasionally moving in the direction of second greatest local novelty can further increase the EVH.

Figure 3 shows the paths of the linear, random, and $C*$ agents. We see that the path from agent $C*$ branches out farther than before, suggesting a more aggressive search for novelty. Figure 4 shows a plot of each agent's local novelty as a function of the time step. All agents with a learning rate of 0.0005 show a large jump in novelty, and investigation shows that these agents spontaneously began producing all-black images, suggesting that 0.0005 is too high for this application.

8 Conclusions

In this paper, we propose a novel method for quantifying explorational value based on previous works on novelty and interestingness, as well an unsupervised, curious agent to seek it out. We then perform a parameter study and an improvement study, showing that our curious agent can determine paths of greater explorational value than linear or random agents. We have also shown that this agent can be successfully implemented using a convolutional autoencoder and simple circular memory.

Table 3: The parameters for the two agents tested in the improvement study

Memory Type	Memory Length	Novelty Function	Train Epochs Per Step	Learning Rate
Circular	64	MSE	3	0.0002
Circular	100	MSE	3	0.0005

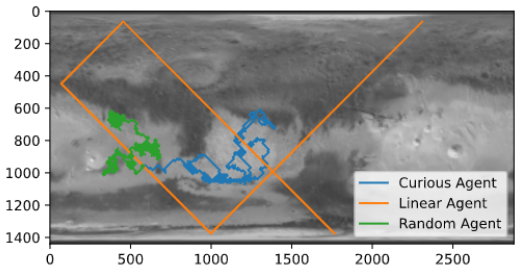

Figure 3: Comparison between an example of the $C*$ with $p = 0.95$, Linear, and Random agents in the the improvement study, all starting at $x = 574, y = 952$.

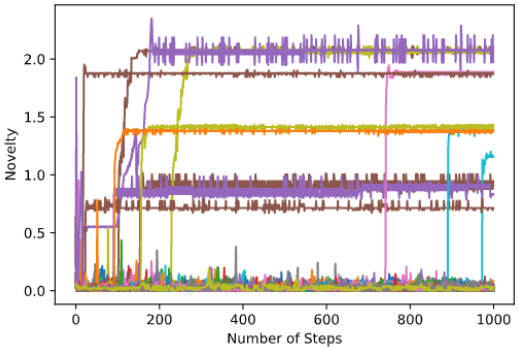

Figure 4: Graph of each agents' local novelty as a function of steps taken

We find that the priority memory does not function as well as expected, and that a longer memory results in a more accurate measurement of novelty, and therefore greater explorational value. We hypothesize this is because the priority-based memory tends to maintain old, non-useful memories at the expense of having a short-term memory. This may explain why agents with priority memory are more likely to retrace their steps in our experiments.

9 Future Work

Our agent assumes that novelty is inherently good; however, it is well established in the literature that unbounded novelty is dangerous [4]. In fact, simulated screens with random images have been shown to captivate curious agents, preventing them from exploring any further [6]. Schaul et al describe a method for balancing novelty with familiarity to achieve a form of artificial curiosity that is more similar to that of humans [15]. This measure, Coherence Progress, is the amount by which the compressibility of past observations increases when a new observation is added [15]. However, this is beyond the scope of this paper.

For future works, we propose a hybrid priority/circular memory that acts as a joint long-term/short-term memory system. Memories are initially added to the queue and entered into the priority queue when popped. We also propose a delayed-initialization method, where memories are not saved to the priority queue until after either a preset number of time steps have passed, or derivative of the loss has fallen below some threshold to allow for agent "equalization." We theorize that this may prevent the agent from placing undue emphasis on its first experiences, since the first few locations invariably receive a high novelty score. This is supported by Figure 5, which shows an initial spike in local novelty.

References

[1] Terry Fong, "Autonomous Systems: NASA Capability Overview," 8 2018.

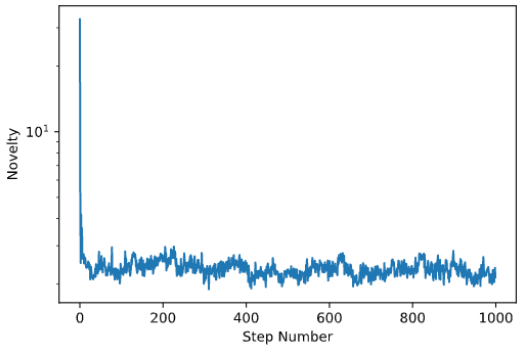

Figure 5: The step-wise average of all curious agents' local novelty, over time (logarithmic).

[2] R. C. Reinhart, J. S. Schier, D. J. Israel, W. Tai, P. E. Liebrecht, and S. A. Townes, "Enabling future science and human exploration with NASA's next generation near earth and deep space communications and navigation architecture," *Proceedings of the International Astronautical Congress, IAC*, vol. 7, pp. 4716–4725, 2017.

[3] K. Hambuchen, M. C. Roman, A. Sivak, A. Herblet, N. Koenig, D. Newmyer, and R. Ambrose, "NASA's space robotics challenge: Advancing robotics for future exploration missions," *AIAA SPACE and Astronautics Forum and Exposition, SPACE 2017*, no. 203999, pp. 1–6, 2017.

[4] V. Graziano, T. Glasmachers, T. Schaul, L. Pape, G. Cuccu, J. Leitner, and J. Schmidhuber, "Artificial curiosity for autonomous space exploration," *Acta Futura*, vol. 4, pp. 41–51, 2011.

[5] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," *34th International Conference on Machine Learning, ICML 2017*, vol. 6, pp. 4261–4270, 2017.

[6] Y. Burda, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[7] G. Schillaci, A. P. Villalpando, V. V. Hafner, P. Hanappe, D. Colliaux, and T. Wintz, "Intrinsic Motivation and Episodic Memories for Robot Exploration of High-Dimensional Sensory Spaces," *Journal of Vibration and Control*, no. X, p. 107754631982824, 2019.

[8] J. Schmidhuber, "Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5499 LNAI, no. April 2009, pp. 48–76, 2009.

[9] H. Huang, X. Hu, Q. Dong, S. Zhao, S. Zhang, Y. Zhao, L. Quo, and T. Liu, "Modeling task fMRI data via mixture of deep expert networks," *Proceedings - International Symposium on Biomedical Imaging*, vol. 2018-April, no. 7, pp. 82–86, 2018.

[10] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep convolutional autoencoder-based lossy image compression," *arXiv*, pp. 253–257, 2018.

[11] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep Features Learning for Medical Image Analysis with Convolutional Autoencoder Neural Network," *IEEE Transactions on Big Data*, vol. 7790, no. c, pp. 1–1, 2017.

[12] S. Karimpouli and P. Tahmasebi, "Segmentation of digital rock images using deep convolutional autoencoder networks," *Computers and Geosciences*, vol. 126, no. October 2018, pp. 142–150, 2019.

[13] P. R. Christensen, J. L. Bandfield, V. E. Hamilton, S. W. Ruff, H. H. Kieffer, T. N. Titus, M. C. Malin, R. V. Morris, M. D. Lane, R. L. Clark, B. M. Jakosky, M. T. Mellon, J. C. Pearl, B. J. Conrath, M. D. Smith, R. T. Clancy, R. O. Kuzmin, T. Roush, G. L. Mehall, N. Gorelick, K. Bender, K. Murray, S. Dason, E. Greene, S. Silverman, and M. Greenfield, "Mars Global Surveyor Thermal Emission Spectrometer experiment: Investigation description and surface science results," *Journal of Geophysical Research: Planets*, vol. 106, 10 2001.

[14] L. Schaul Tom
}and Pape, G. Tobias, G. Vincent, and S. Jürgen, "Coherence Progress: A Measure of Interestingness Based on Fixed Compressors," in *Artificial General Intelligence* (K. R. Schmidhuber Jürgen
}and Thórisson and L. Moshe, eds.), (Berlin, Heidelberg), pp. 21–30, Springer Berlin Heidelberg, 2011.

[15] T. Schaul, L. Pape, T. Glasmachers, V. Graziano, and J. Schmidhuber, "Coherence progress: A measure of interestingness based on fixed compressors," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6830 LNAI, pp. 21–30, 2011.

[16] A. Aubret, L. Matignon, and S. Hassas, "A survey on intrinsic motivation in reinforcement learning," *arXiv*, no. Im, 2019.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 12 2015.

A Graphs

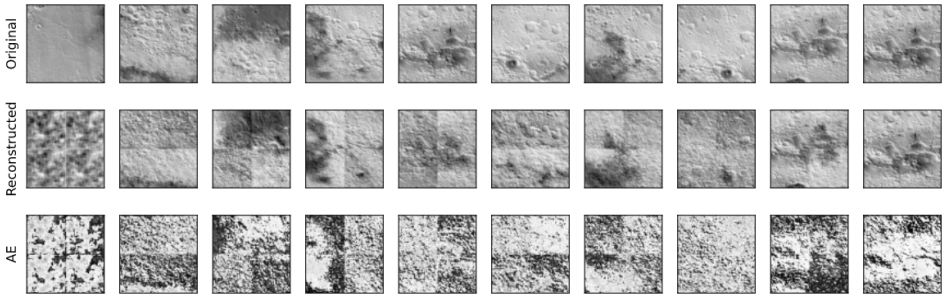

Figure 6: The progression of an agent's image reconstructions over time, including the original image, the agent's reconstruction, and the absolute error (AE) per pixel for the agent's guess. This is for agent $C*$ at position (728, 830) in the parameter study. Time increases to the right, with an interval of 50. Note the four separate camera regions in each image, indicating the four parts of the agent's field of view.

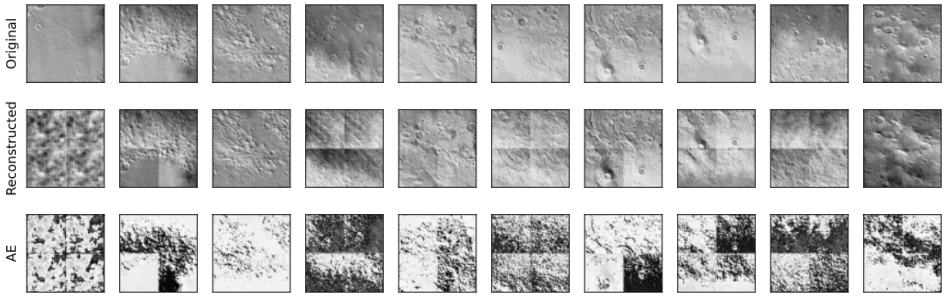

Figure 7: The progression of an agent's image reconstructions over time, including the original image, the agent's reconstruction, and the absolute error (AE) for the agent's guess. This is for agent $C*$ with $p = 0.95$ at position (728, 830) in the improvement study. Time increases to the right, with an interval of 50. Note the four separate camera regions in each image, indicating the four parts of the agent's field of view.

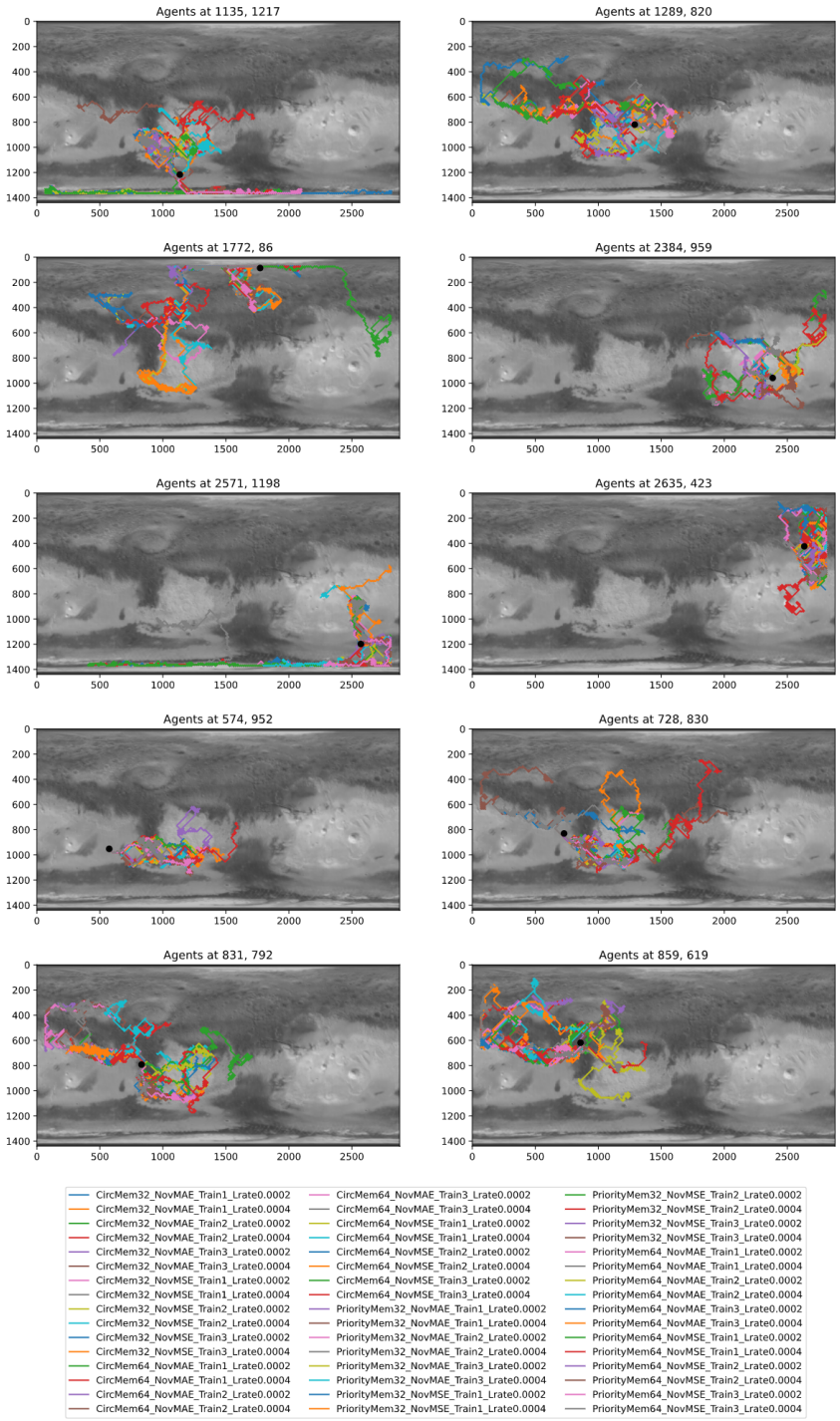

Figure 8: The paths traveled by curious agents in the parameter study, grouped by starting position (indicated by the black dot).

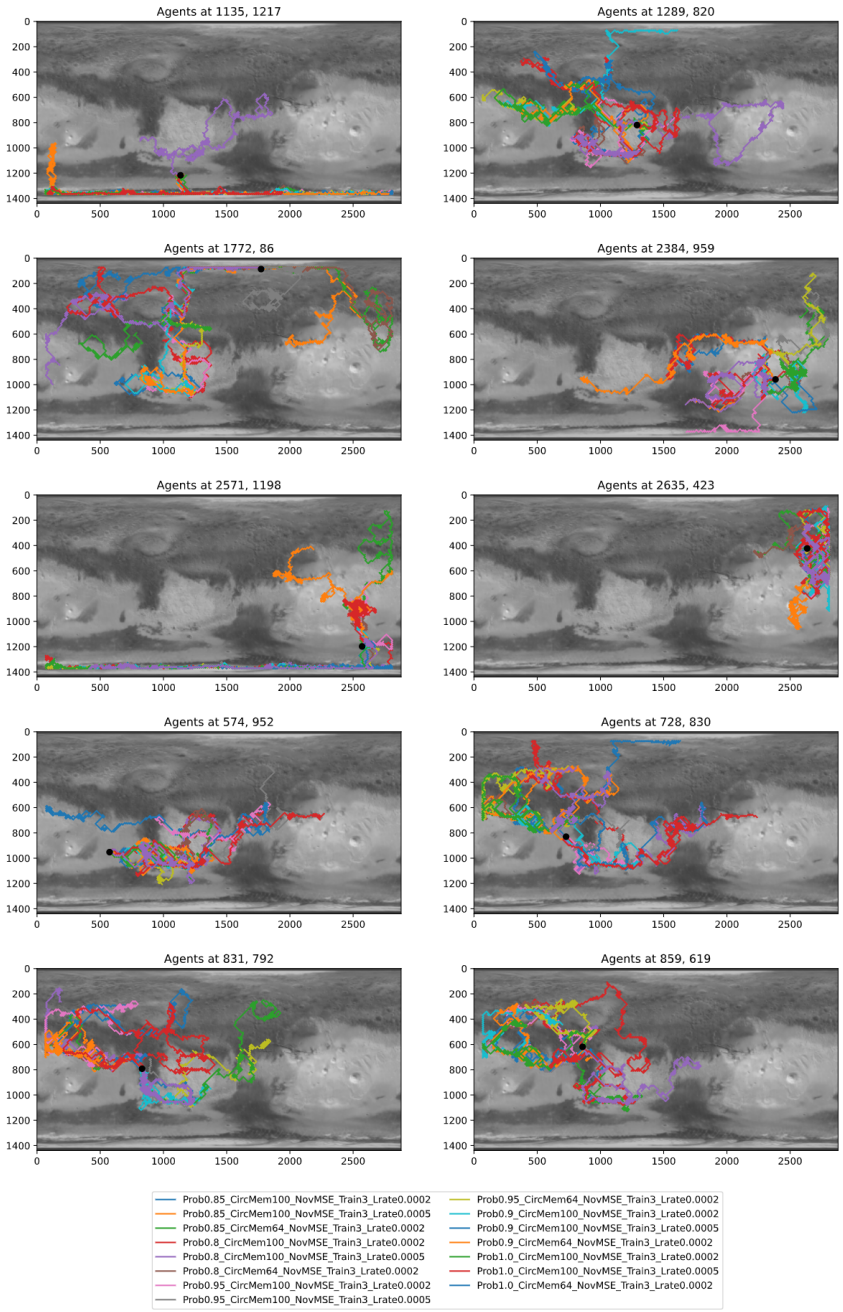

Figure 9: The paths traveled by curious agents in the improvement study, grouped by starting position (indicated by the black dot).

B Tables

Table 4: Agent ranking based on average EVH from the improvement study, and includes the linear and random agents for comparison.

Agent Specifications			Average EVH
Prob p	Mem. Length	Learning Rate	
0.95	100	0.0005	0.065455
0.95	64	0.0002	0.062416
Linear Agent	–	–	0.061396
0.85	64	0.0002	0.059905
1.0	64	0.0002	0.059826
0.8	100	0.0005	0.059710
0.9	64	0.0002	0.056228
1.0	100	0.0005	0.054425
0.8	64	0.0002	0.052144
0.9	100	0.0005	0.047611
0.85	100	0.0005	0.046986
Random Agent	–	–	0.040246

Table 5: Architecture of our Convolutional Autoencoder. All convolutional and dense layers used the He initializer [17] and a weight regularizer with weight factor 0.01.

Layer Type	Output Shape	Number of Parameters
Input Layer	[(None, 64, 64, 1)]	0
Convolutional 2d	[(None, 64, 64, 32)]	320
Average Pooling 2d	[(None, 32, 32, 32)]	0
Convolutional 2d	[(None, 32, 32, 64)]	18496
Average Pooling 2d	[(None, 16, 16, 64)]	0
Convolutional 2d	[(None, 16, 16, 128)]	73856
Average Pooling 2d	[(None, 8, 8, 128)]	0
Flatten	[(None, 8192)]	0
Dense	[(None, 512)]	524800
Dense	[(None, 256)]	131328
Dense	[(None, 256)]	65792
Dense	[(None, 256)]	131328
Dense	[(None, 512)]	524800
Reshape	[(None, 8, 8, 128)]	0
UpSampling 2d	[(None, 16, 16, 128)]	0
Convolutional 2d	[(None, 16, 16, 128)]	147584
Upsampling 2d	[(None, 32, 32, 128)]	0
Convolutional 2d	[(None, 32, 32, 64)]	73792
UpSampling 2d	[(None, 64, 64, 64)]	0
Convolutional 2d	[(None, 64, 64, 32)]	18464
Convolutional 2d	[(None, 64, 64, 1)]	289

Table 6: Ranking of all agents in the parameter study based on average EVH, and includes the linear and random agents for comparison.

Agent Specifications					Average EVH
Mem. Type	Mem. Length	Novelty Func.	Train Epoch	Learning Rate	
Circular	64	MSE	3	0.0002	0.063059
Linear Agent	–	–	–	–	0.061396
Circular	64	MAE	3	0.0002	0.056734
Circular	64	MSE	3	0.0004	0.056535
Circular	64	MAE	2	0.0004	0.055693
Circular	64	MSE	2	0.0002	0.055447
Circular	32	MSE	2	0.0004	0.054584
Circular	64	MSE	1	0.0004	0.053942
Circular	64	MAE	1	0.0002	0.052104
Circular	32	MSE	2	0.0002	0.051726
Circular	32	MAE	3	0.0004	0.051714
Circular	32	MSE	1	0.0004	0.051628
Circular	32	MSE	3	0.0002	0.051272
Circular	64	MAE	3	0.0004	0.051129
Circular	32	MAE	1	0.0004	0.051014
Circular	32	MSE	3	0.0004	0.050281
Circular	64	MSE	2	0.0004	0.050122
Circular	64	MSE	1	0.0002	0.050024
Circular	32	MAE	2	0.0002	0.048858
Circular	64	MAE	1	0.0004	0.048574
Circular	64	MAE	2	0.0002	0.048422
Circular	32	MAE	2	0.0004	0.048365
Circular	32	MAE	3	0.0002	0.046087
Priority	64	MAE	2	0.0002	0.045693
Priority	64	MSE	1	0.0004	0.04358
Circular	32	MSE	1	0.0002	0.043454
Priority	64	MAE	1	0.0004	0.042469
Circular	32	MAE	1	0.0002	0.042032
Priority	64	MSE	3	0.0004	0.041196
Priority	64	MAE	3	0.0004	0.040405
Random Agent	–	–	–	–	0.040246
Priority	64	MAE	2	0.0004	0.039337
Priority	64	MSE	2	0.0004	0.038877
Priority	32	MSE	3	0.0004	0.038787
Priority	64	MSE	2	0.0002	0.036703
Priority	64	MSE	3	0.0002	0.035277
Priority	32	MAE	3	0.0002	0.034433
Priority	32	MAE	1	0.0004	0.033444
Priority	32	MAE	2	0.0004	0.033362
Priority	32	MAE	3	0.0004	0.03229
Priority	32	MSE	3	0.0002	0.032211
Priority	32	MSE	1	0.0004	0.031372
Priority	32	MSE	2	0.0004	0.030789
Priority	32	MSE	2	0.0002	0.029248
Priority	32	MAE	2	0.0002	0.029231
Priority	64	MAE	3	0.0002	0.028108
Priority	32	MAE	1	0.0002	0.027959
Priority	64	MSE	1	0.0002	0.027815
Priority	64	MAE	1	0.0002	0.026656
Priority	32	MSE	1	0.0002	0.024469