Neuro-Inspired Fragmentation and Recall to Overcome Catastrophic Forgetting in Curiosity

Jaedong Hwang, Zhang-Wei Hong, Eric Chen, Akhilan Boopathy, Pulkit Agrawal, Ila Fiete
Massachusetts Institute of Technology
{jdhwang, zwhong, ericrc, akhilan, pulkitag, fiete}@mit.edu

Abstract

Intrinsic reward functions are widely used to improve exploration in reinforcement learning. We first examine the conditions and causes of catastrophic forgetting of the intrinsic reward function, and propose a new method, *FARCuriosity*, inspired by how humans and non-human animals learn. The method depends on fragmentation and recall: an agent fragments an environment based on surprisal signals, and uses different local curiosity modules (prediction-based intrinsic reward functions) for each fragment so that modules are not trained on the entire environment. At each fragmentation event, the agent stores the current module in long-term memory (LTM) and either initializes a new module or recalls a previously stored module based on its match with the current state. With fragmentation and recall, FARCuriosity achieves less forgetting and better overall performance in games with varied and heterogeneous environments in the Atari benchmark suite of tasks. Thus, this work highlights the problem of catastrophic forgetting in prediction-based curiosity methods and proposes a first solution[1].

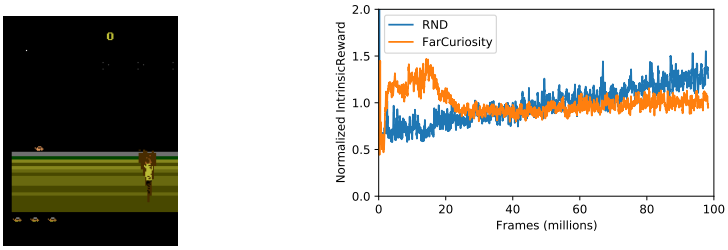
1 Introduction

Many real-world reinforcement learning tasks do not supply extrinsic rewards at every time-step, making it difficult for the agent to assess the effectiveness of different actions: rewards might only be available after a long sequence of observations and actions. This *sparse reward* setting makes learning a policy challenging, and it can be helpful to add a dense and internally generated auxiliary reward (called an *intrinsic reward*) that promotes exploration. Many works demonstrate the utility of intrinsic rewards, and there are a variety of popular methods for generating such rewards [6, 8, 31]. Most count-based intrinsic reward functions [1, 6, 14, 41, 44] define the intrinsic reward of a state proportional to the inverse square of the number of visits to that state. Other models use prediction errors as a measure of novelty and thus intrinsic reward: Intrinsic Curiosity Module (ICM) [31] employs a forward and an inverse model to predict future states and action consequences, and high intrinsic rewards are given to transitions that result in high prediction error [38]. As the forward model improves its predictions, the intrinsic reward decreases. RND (Random Network Distillation) [8] learns to predict the next state via a random feature space. Count-and prediction-also based models have been combined [33], and goal-based methods have been recently studied [9].

Models that depend on learning to generate intrinsic reward (e.g. prediction-based models including RND and ICM) rely on the model not forgetting information about previously visited states (catastrophic forgetting [17, 36]). The issue of forgetting in prediction-based intrinsic reward models is usually overlooked, possibly because of the standard reinforcement learning paradigm in which each episode is reset to the same starting observation. However, as we show in Figure 1, the intrinsic reward for the same starting observation on each episode computed by RND throughout training on

[1] https://github.com/FieteLab/FARCuriosity

37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) Starting observation (b) Intrinsic reward of (a)

Figure 1: Intrinsic reward generated by RND [8] and our proposed method during training at the first observation in each episode of Jamesbond, which is the same across episodes. Rewards are normalized separately using the average intrinsic reward across training, in order to directly compare the magnitude of the change in intrinsic reward. Intrinsic reward from RND continues to increase at the start location across training and repeated observations, rather than decreasing.

Jamesbond continues to increase. This suggests that the starting observation is gradually forgotten over the course of learning within the episode, an indication of catastrophic forgetting.

We analyze when and why catastrophic forgetting happens even in the standard RL paradigm with environment resets at each episode, and suggest one possible relaxation inspired by how natural agents store information. The episodic memory of natural agents divides their continuous experience of the world into multiple episodes (fragments) based on large changes in context [5, 16, 28, 35, 42, 50]. It plays an important role in learning and cognition [12, 15, 18, 19, 40]. Hence, we propose a fragmentation-and-recall-based curiosity (*FARCuriosity*) to tackle catastrophic forgetting in the prediction-based intrinsic reward functions. While an agent explores an environment, a curiosity module generates an intrinsic reward (prediction error). Based on this, the agent decides to store the current module in long-term memory (LTM) and initialize a new one (*fragmentation*). Alternatively, the agent refers to LTM and *recalls* an existing curiosity module if the current observation is similar to one when the fragmentation happened before. Consequently, each curiosity module does not train from the entire space of an environment but trains from a similar subspace, and it reduces the hazard of catastrophic forgetting. Experimental results show that FARCuriosity achieves better performance than the baseline fragmentation-less curiosity module [8] on heterogeneous environments that have diverse states, leading huge distribution shifts among standard Atari game benchmarks [7].

2 Catastrophic Forgetting in Prediction-Based Intrinsic Reward Function

2.1 Catastrophic Forgetting

Catastrophic forgetting denotes the phenomenon where neural networks lose previously learned information while learning new information [17, 36]. This problem has surfaced as an important issue in 'continual learning' task settings. In continual learning, the network is trained sequentially on a set of tasks. Each task is typically composed of disjoint subsets (e.g. classes) of datasets which may consist of a set of classes as in class-incremental learning [13, 30, 34]. On the other hand, Wołczyk *et al.* [48] proposed a continual reinforcement learning (CRL) benchmark: an agent is sequentially trained on multiple robot manipulation tasks in Meta-World [49] and catastrophic forgetting across tasks is calculated. Catastrophic forgetting is measured as negative 'backward transfer (BWT)', which quantifies the performance that drops on earlier tasks after training on later tasks [27].

2.2 Prediction-Based Intrinsic Reward Function

In reinforcement learning, the intrinsic reward is usually designed to encourage the agent to explore novel states. It is especially important in a sparse reward environment that the agent is hard to get any reward during training. There is a key assumption in intrinsic reward functions: The intrinsic reward function is a monotonically decreasing function for a given state s:

$$r^i_{t_1}(s) \leq r^i_{t_2}(s) \iff t_1 > t_2, \tag{1}$$

where $r^i_t(s)$ denotes the intrinsic reward for a state s at time t. This leads the agent to explore novel (unvisited) states instead of already visited states.

In count-based exploration methods [6, 41, 44], the intrinsic reward function is defined as the inverse of the square root of the number of visits (or pseudo-count). However, prediction-based intrinsic reward functions [8, 31, 32] define the prediction error of the neural network for the state (or state-action pair) as an intrinsic reward. As the training goes on, the function generates lower rewards for states that are visited multiple times. Consequently, it seems that prediction-based intrinsic reward methods follow the assumption mentioned above. However, this is true only if the network does not suffer from catastrophic forgetting that makes the network lose information about already visited states. Standard reinforcement learning benchmarks [7, 24, 26, 43, 45, 46] usually allow resetting the environment multiple times during training so that the agent can explore different subsets of the observation space. This seems to prevent catastrophic forgetting (increment of intrinsic reward for the previously visited states) since the agent has a chance to visit the same state (*e.g.*, the starting state) during training.

Nevertheless, catastrophic forgetting can occur especially in multi-stage environments such as Jamesbond in Atari benchmark. In the games, the agent moves in one direction and cannot go back during the episode. As the agent is trained, it visits more states and lives longer in each episode. This can lead to an increase in prediction errors since the network is updated more times using other observations before observing the same state. In other words, the non-i.i.d property of the reinforcement learning environment causes catastrophic forgetting.

3 FARCuriosity

3.1 Overview

Intrinsic reward functions [8, 9, 31–33] are used for exploring novel spaces of an environment by generating intrinsic rewards. The intrinsic reward can serve as the primary reward for training policies in realistic scenarios where external rewards are sparse or absent. Since a curiosity module is trained concurrently with the policy, it can suffer from catastrophic forgetting [17, 36] over large and heterogeneous environments where the observation space has high variance such as having multiple stages with different backgrounds as we mentioned in Appendix 2. Hence, generating intrinsic rewards in these environments demands a larger model capacity to memorize what the model has seen. The catastrophic forgetting of a curiosity module leads to high prediction errors (high intrinsic rewards) for already explored states, only encouraging to visit those states.

We tackle this problem based on the notion of fragmentation and recall, the interaction between short-term memory (STM) inside of an agent (*e.g.*, RAM, VRAM) and long-term memory (LTM) outside of an agent (*e.g.*, external memory; SSD, flash memory). STM stores the current local curiosity module that generates intrinsic rewards and is updated while an agent explores the environment. The intrinsic reward (prediction error) is also used as a surprisal which is a criterion of fragmentation. On the other hand, LTM stores unused curiosity modules.

When the surprisal (intrinsic reward) exceeds some threshold, this corresponds to a *fragmentation* event. At a fragmentation event, the local model is written to long-term memory (LTM) to preserve the local information without forgetting (Figure 2a). The current (abstracted) observation is also stored and used as a key to recall the corresponding model, and the agent initializes an entirely new local model. On the other hand, if the current observation is sufficiently similar to a stored observation (key) in LTM, the agent *recalls* the corresponding model fragment (local model) (Figure 2b). Hence, the agent can preserve and reuse previously acquired information. We name this framework fragmentation-and-recall curiosity (*FARCuriosity*). The goal of *FARCuriosity* is to overcome catastrophic forgetting issues in heterogeneous environments to better explore states. We expect this improved exploration to support improved performance on many tasks, above the performance improvements provided by using the existing prediction-based intrinsic-reward approaches that suffer from catastrophic forgetting [8].

3.2 Fragmentation

Fragmentation divides the input space into multiple subspaces and allocates a curiosity module to each subspace. A fragmentation event occurs if the ratio of the current surprisal and the running average of the surprisal within the local region is greater than a threshold, ρ. However, z-score can be used to incorporate the variance of the surprisal in the subset of the environment. When the fragmentation

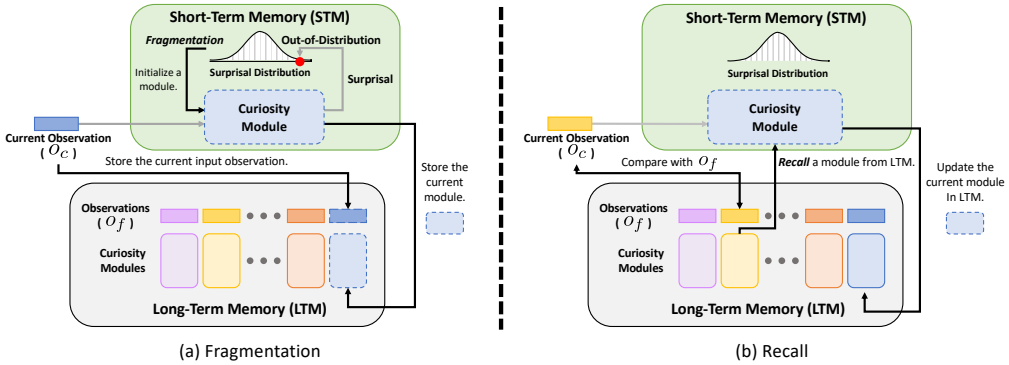


Figure 2: Illustration of the proposed FARCuriosity framework. (a) *Fragmentation*. If the current surprisal is out-of-distribution with a high value (gray arrows), the current curiosity module and its input are stored in LTM and a new module is initialized. (b) *Recall*. If the current observation is similar to an observation in LTM, the corresponding curiosity module is recalled and we update the current module in LTM.

occurs, we store the current curiosity module and its input (usually the next observation) in LTM and initialize a new module as shown in Figure 2a. Consequently, the previous module can preserve the information of the local region without forgetting caused by learning from other spaces.

3.3 Recall

The recall is designed to load the previous curiosity module from LTM when an agent re-enters a local space corresponding to the module. At every time step, the agent can search LTM to select a curiosity module. The curiosity module's input observation (or its encoded feature) is the key used to select a module: Given the current input observation, o_c, and the stored observation in LTM, of which an element is o_f, and a fixed feature extractor $\phi(\cdot)$ (*e.g.*, target network in RND [8]), we define a recall score s_f^{rec} as a cosine similarity between two (encoded) observations. If s_f^{rec} is equal to or greater than a recall threshold ψ for any element o_f, we store the current module in LTM and recall the module that corresponds to o_f.

4 Experiments

We analyze the catastrophic forgetting of RND and its cause in prediction-based intrinsic reward approaches with a toy example and show performance degradation due to catastrophic forgetting in the appendix.

4.1 Comparison with a Baseline

If the agent only explores a small portion of the environment, fragmentation is not needed although the environment has hundreds of distinctive spaces. Therefore, we define *heterogeneity* of each Atari environment as the average across all pixels of the standard deviation of each pixel in generated trajectories by a trained policy:

$$(Heterogenity) = \frac{1}{H \times W} \sum_{(h,w)\in(H,W)} \sigma(I_{1\leq t\leq T,h,w}), \tag{2}$$

where $\sigma(\cdot)$ denotes standard deviation and $I \in \mathbb{R}^{T\times H\times W}$ denotes a generated trajectory composed of T frames with size $H \times W$. We use vanilla PPO [39] instead of RND to give a guideline for choosing RND and FARCuriosirty in the environment.

Figure 3 measures the relative performance improvement on various Atari games which is defined as below:

$$(Relative\ Improvement) = \frac{R_{\text{FARCuriosity}} - R_{\text{RND}}}{R_{\text{RND}}}, \tag{3}$$

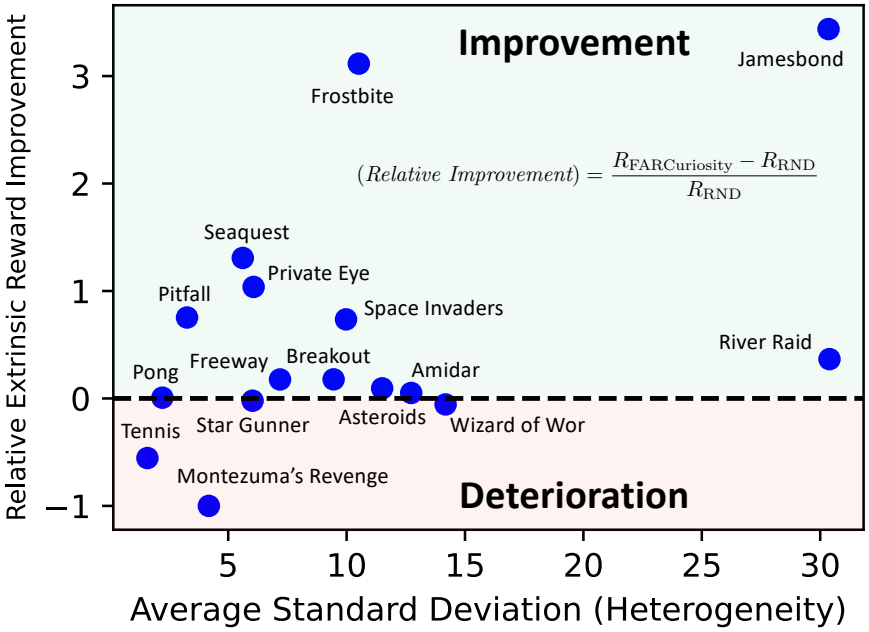


Figure 3: The relative performance improvement over RND performance (R_{RND}) in various Atari games. The average standard deviation is measured from each pixel in a trajectory generated by trained RND. FARCuriosity is effective in visually high-variance (heterogeneous) games.

where $R_{\text{FARCuriosity}}$ and R_{RND} denote the average extrinsic reward of FARCuriosity and RND, respectively. Since both $R_{\text{FARCuriosity}}$ and R_{RND} are negative in Tennis and Pitfall, we multiply the improvement by -1 here. The negative improvement denotes that performance is worsened by FARCuriosity. FARCuriosity generally improves performance in heterogeneous environments while achieving similar or even worse performance in homogeneous games. FARCuriosity achieves worse performance in Montezuma's Revenge which might be due to its low heterogeneity since PPO could not explore many rooms. In the least heterogeneous environment, Tennis where only players and a ball are moving with a fixed background, FARCuriosity deteriorates compared to RND. We analyze the relationship between the number of fragments and the performance in Appendix D and present the average game score for each model in each environment in Table 2 and Figure 10 in the appendix.

5 Discussion

We shed light on the problem of catastrophic forgetting when using prediction-based intrinsic rewards in reinforcement learning. Catastrophic forgetting happens not only in reset-free environments but also in resetting environments as an agent explores more observation spaces and lives longer. The forgetting leads to a collapse of the assumption in the intrinsic reward function: the function is a monotonically decreasing function on the same observation over time. Consequently, it can potentially yield a problem: the agent is prone to visit already-visited states. We also demonstrate the hazard of catastrophic forgetting using count-based exploration with exponential decay in a minigrid environment.

To alleviate the catastrophic forgetting problem, we propose a simple method inspired by how natural agents explore space. Our FARCuriosity *fragments* the exploration space (*i.e.*, observation space) based on the surprisal, which is generated by a curiosity module [8] in short-term memory (STM), in real-time. During fragmentation, an agent stores the current curiosity module in long-term memory (LTM) and the stored module *recalled* when the agent returns to the state where the fragmentation happened so that the agent can reuse the local information. Consequently, FARCuriosity achieves better performance in heterogeneous environments while it has drawbacks in homogeneous environments. However, we did not assess fragmentation quality and leave it for future study. Our paper aims to highlight the catastrophic forgetting issue in prediction-based intrinsic reward, which is potentially hazardous to RL agents by misguiding behavior.

References

[1] Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

[2] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *NeurIPS*, 34, 2021.

[3] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *JMLR*, 3:397–422, 2003.

[4] Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *ICML*, 2020.

[5] Christopher Baldassano, Janice Chen, Asieh Zadbood, Jonathan W Pillow, Uri Hasson, and Kenneth A Norman. Discovering event structure in continuous narrative perception and memory. *Neuron*, 95(3):709–721, 2017.

[6] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *NeurIPS*, 2016.

[7] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *JAIR*, 47:253–279, 2013.

[8] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *ICLR*, 2019.

[9] Andres Campero, Roberta Raileanu, Heinrich Kuttler, Joshua B. Tenenbaum, Tim Rocktäschel, and Edward Grefenstette. Learning with amigo: Adversarially motivated intrinsic goals. In *ICLR*, 2021.

[10] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. `https://github.com/maximecb/gym-minigrid`, 2018.

[11] Leshem Choshen, Lior Fox, and Yonatan Loewenstein. Dora the explorer: Directed outreaching reinforcement action-selection. In *ICLR*, 2018.

[12] Adrianus Dingeman De Groot. Het denken van den schaker, een experimenteelpsychologie studie. 1946.

[13] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020.

[14] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

[15] Dennis E Egan and Barry J Schwartz. Chunking in recall of symbolic drawings. *Memory & cognition*, 7(2):149–158, 1979.

[16] Youssef Ezzyat and Lila Davachi. What constitutes an episode in episodic memory? *Psychological science*, 22(2):243–252, 2011.

[17] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[18] Fernand Gobet, Peter CR Lane, Steve Croker, Peter CH Cheng, Gary Jones, Iain Oliver, and Julian M Pine. Chunking mechanisms in human learning. *Trends in cognitive sciences*, 5(6):236–243, 2001.

[19] Fernand Gobet and Herbert A Simon. Expert chess memory: Revisiting the chunking hypothesis. *Memory*, 6(3):225–255, 1998.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration, 2017.

[22] Chia-Chun Hung, Timothy Lillicrap, Josh Abramson, Yan Wu, Mehdi Mirza, Federico Carnevale, Arun Ahuja, and Greg Wayne. Optimizing agent behavior over long time scales by transporting value. *Nature communications*, 10(1):1–12, 2019.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

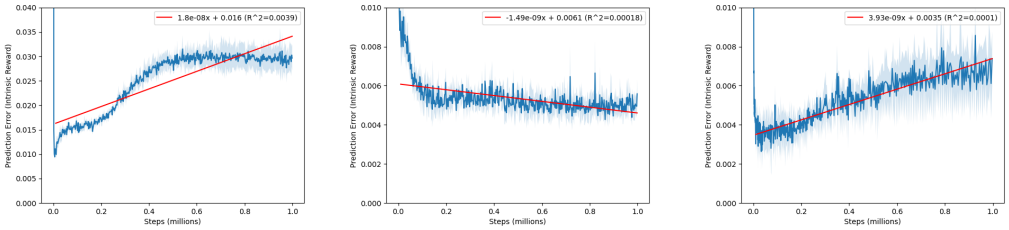
[24] Heinrich Küttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The NetHack Learning Environment. In *NeurIPS*, 2020.

[25] Andrew Lampinen, Stephanie Chan, Andrea Banino, and Felix Hill. Towards mental time travel: a hierarchical memory for reinforcement learning agents. In *NeurIPS*, 2021.

[26] Youngwoon Lee, Edward S Hu, and Joseph J Lim. IKEA furniture assembly environment for long-horizon complex manipulation tasks. In *ICRA*, 2021.

[27] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017.
[28] Darren Newtson and Gretchen Engquist. The perceptual organization of ongoing behavior. *Journal of Experimental Social Psychology*, 12(5):436–450, 1976.
[29] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V. Hafner. Intrinsic motivation systems for autonomous mental development. *TEVC*, 11(2):265–286, 2007.
[30] Jaeyoo Park, Minsoo Kang, and Bohyung Han. Class-incremental learning for action recognition in videos. In *ICCV*, 2021.
[31] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.
[32] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *ICML*, 2019.
[33] Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. In *ICLR*, 2020.
[34] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.
[35] Lauren L Richmond and Jeffrey M Zacks. Constructing experience: Event models from perception to action. *Trends in cognitive sciences*, 21(12):962–980, 2017.
[36] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.
[37] Nikolay Savinov, Anton Raichuk, Damien Vincent, Raphael Marinier, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. In *ICLR*, 2019.
[38] Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pages 222–227, 1991.
[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
[40] Herbert A Simon. How big is a chunk? by combining data from several experiments, a basic human memory unit can be identified and measured. *Science*, 183(4124):482–488, 1974.
[41] Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.
[42] Khena M Swallow, Jeffrey M Zacks, and Richard A Abrams. Event boundaries in perception affect memory encoding and updating. *Journal of Experimental Psychology: General*, 138(2):236, 2009.
[43] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *NeurIPS*, 2021.
[44] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In *NeurIPS*, 2017.
[45] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018.
[46] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, 2012.
[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
[48] Maciej Wołczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. In *NeurIPS*, 2021.
[49] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2020.
[50] Jeffrey M Zacks and Khena M Swallow. Event segmentation. *Current directions in psychological science*, 16(2):80–84, 2007.
[51] Shaowu Zhang, Fiola Bock, Aung Si, Juergen Tautz, and Mandyam V Srinivasan. Visual working memory in decision making by honey bees. *PNAS*, 102(14):5250–5255, 2005.

Appendix



(a) reset-free (b) fixed episode length (c) gradually increased episode length

Figure 4: Prediction error (intrinsic reward) of the starting observation in a 10×10 grid environment with (a) reset-free, (b) fixed episode length, and (c) gradually increased episode length. The prediction error in reset-free environments is computed every 200 steps with the starting observation while others are measured when the agent visits there. The error is increased in reset-free environments and when the episode length is gradually increased, which means catastrophic forgetting happens.

A Related Work

A.1 Curiosity-Driven Reinforcement Learning

Motivated by the insufficiency of rewards alone to guide exploration in sparsely rewarded RL environments, several works construct *intrinsic rewards* encouraging agents to explore unfamiliar parts of an environment. Count-based methods provide intrinsic rewards for reaching infrequently visited states. These methods have strong theoretical guarantees for multi-arm bandits [3] and can be effective for reinforcement learning [6, 14, 41, 44]. Prediction-based methods estimate where the agent cannot predict some aspect of the environment, and provide intrinsic rewards for visiting those states [1, 8, 11, 21, 31, 33]. ICM [31], for instance, learns a forward model to predict the next state given the current state-action pair and provides rewards scaling with the state's prediction error. On the other hand, Intelligent Adaptive Curiosity (IAC) [29] divides sensorimotor space (state and action pairs) into multiple regions and allocates each expert module to the regions. Although IAC requires memorizing all exemplars and directly using the similarity of exemplars as a fragmentation criterion, FARCuriosity does not need to memorize them and use prediction error (surprisal) as the criterion.

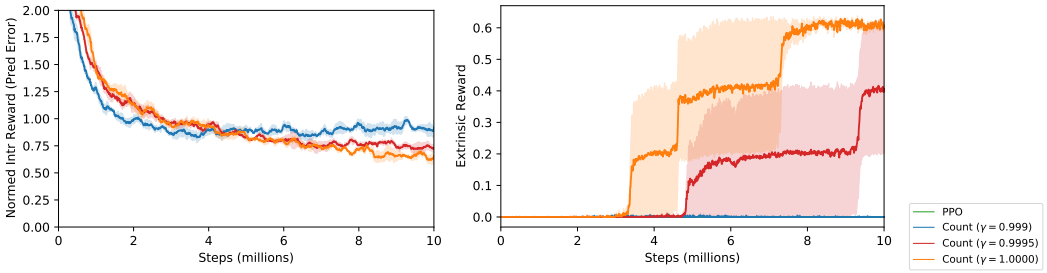
A.2 Memory-Based Reinforcement Learning

Memory-based reinforcement learning aims to solve the long-term credit assignment problem. Hung *et al.* [22] combine LSTM [20] with external memory along with an encoder and decoder for the memory. HCAM [25] uses a hierarchical LTM with chunks and attention for long-term recall inspired by Transformers [47]; however, its chunks are formed periodically rather than based on content and are not used as intrinsic options for exploration. Memory is also used for improving exploration: Go-Explore [14] remembers promising states so that it can directly visit those states, where the definition of promising can include rewarded states. Episodic Curiosity [37] memorizes visited states for calculating reachability to penalize an action that leads to reachable future states. While earlier memory-based exploration methods use memory as a guide for evaluating the intrinsic reward of a state, we augment existing exploration methods with memory, in combination with map or model fragmentation, to more efficiently explore.

B Analysis of Catastrophic Forgetting

B.1 Cause of Catastrophic Forgetting

We illustrate when and why catastrophic forgetting can happen with a toy experiment. We generate a 10×10 grid environment where the observation of each location is a randomly generated 32-dimensional vector. The agent starts from the center of the environment and randomly moves up,



(a) Intrinsic reward (b) Extrinsic reward

Figure 5: (a) Intrinsic of the first observation in every episode, and (b) extrinsic reward in MultiRoom-N6 in MiniGrid benchmark during training generated by four different models: PPO and count-based exploration with various exponential decay γ. Even small forgetting (exponential decay) dramatically degrades the performance.

down, left, and right. However, the agent is prohibited from visiting the start location. When the environment is reset, the agent starts from the center of the environment again. We train RND [8] module with random policy in this environment for one million steps sequentially. Please refer to Section C.2 for the details.

Figure 4 presents the intrinsic reward (prediction error) changes of the starting point across the training with various scenarios: reset-free, fixed episode length, and gradually increased episode length. In the reset-free setting (Figure 4a), we store an observation of the starting position and keep track of the corresponding intrinsic reward during training to compare with other settings. The intrinsic reward in this setting increases since the model is not trained on the start observation throughout training except for the first time step. On the other hand, the reward keeps decreasing in fixed episode length (200) in Figure 4b, but if we increase the episode length, it will be the same as the reset-free setting. Although the environment is reset, if the episode length is gradually increased, the intrinsic reward increases as shown in Figure 4c. This is because as the episode length increases, the network is updated using different observations, which leads to the network forgetting the observation in the start position. Although this example only shows the starting observation for simplification, this scenario can be generalizable to all observations that are hard to reach during episodes. Furthermore, in most real-world environments, it is difficult to go back to the previously visited states; although we can reset the agent, or it is impossible to reset the environment. Hence, catastrophic forgetting is an issue that may need to be considered when training an agent using a prediction-based intrinsic reward function.

B.2 Hazard of Catastrophic Forgetting

To investigate the risks of catastrophic forgetting, we study the relationship between forgetting and performance drops. To manipulate the degree of catastrophic forgetting and reduce the variance from the environment, we use count-based exploration rather than a prediction-based model in MultiRoom-N6 of MiniGrid [10]. The intrinsic reward of count-based exploration is defined as $1/\sqrt{N_{\text{visit}}(o)}$, where $N_{\text{visit}}(o)$ denotes the number of visits to the observation, o [41]. Here, we add exponential decay to the number of visits with a factor of γ on every iteration:

$$N_{\text{visit}}(o) \leftarrow \gamma \cdot N_{\text{visit}}(o) + b, \tag{4}$$

where b is 1 if an agent visits o at that time, otherwise 0. This exponential decay presents a forgetting effect that models the memory retention of short-term memory of natural agents [51]. Please refer to Appendix C.3 for experimental details.

Figure 5 shows the mean cumulative extrinsic rewards of PPO with count-based exploration in MultiRoom-N6. Count-based exploration without exponential decay achieves 0.6 within 10 million steps while vanilla PPO cannot learn this task. Count-based exploration with γ as 0.9995 has already worse performance than the model without forgetting. while an agent's performance is almost 0 when γ is 0.999. This means that even a mere 0.1% of forgetting in every iteration dramatically affects the performance.

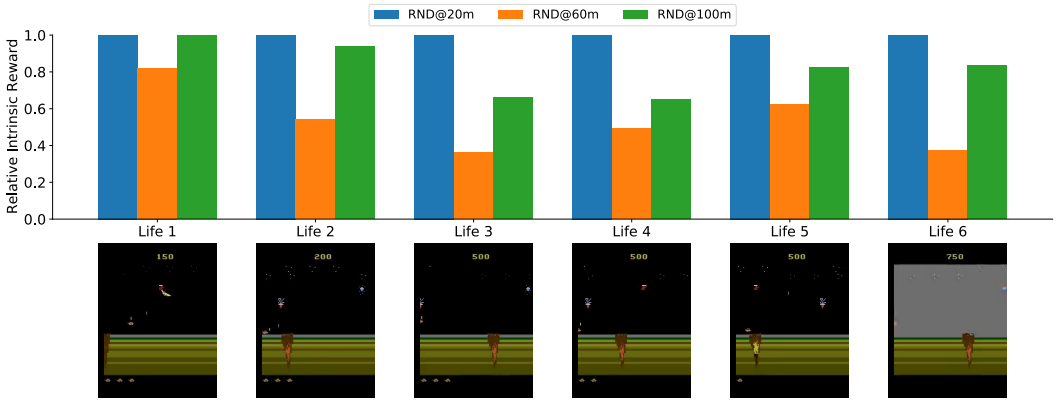


Figure 6: The intrinsic rewards of RND in three different checkpoints during training (training on 20, 60, and 100 million frames, respectively) when an agent is about to die in a video generated by RND trained on 20 million frames (RND@20m). Intrinsic reward for each frame is normalized by the original intrinsic rewards generated by RND@20m. High intrinsic reward denotes that the agent is misguided to die. In every moment, intrinsic rewards from RND training from 60 million frames (RND@60m) are lower than RND@20m. However, the rewards are increased again at the end of training (RND@100m).

B.3 Catastrophic Forgetting of RND

we conduct further analysis of RND [8] in Jamesbond where an agent starts from the same position every episode and moves in one direction without return. We first generate a trajectory of an episode from a policy after training 20 million frames and then, measure intrinsic rewards of RND at three different checkpoints; after training 20, 60, and 100 million frames, when the agent is dead by missiles, enemies, or obstacles. As shown in Figure 6, although the intrinsic rewards are decreased in the middle of training, they subsequently increase This implies that training decreases the reward but catastrophic forgetting happens between 60 million frames and 100 million frames. This might be related to the agent more frequently going to the ocean which has a different background and new enemies (*e.g.*, frogman, mini-submarines) in the later stage of training. It leads to catastrophic forgetting about the previous scene.

Moreover, we measure the intrinsic reward of the first observation in every episode generated by RND during training in Jamesbond in Figure 1b. The intrinsic reward generated by RND increases throughout training, implying that this method suffers from catastrophic forgetting. On the other hand, the intrinsic reward of the FARCuriosity agent at the observation remains almost the same after a short initial increase, regardless of how much subsequent learning the agent performs elsewhere in the environment.

C Experimental Details

Our model is implemented on PyTorch and the experiments are conducted on Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz and a single NVIDIA Tesla V100 GPU for reinforcement learning experiments.

C.1 Main Experiments

We implement our FARCuriosity method based on RND [8] and compare it against the vanilla RND and Proximal Policy Optimization (PPO) [39]. Both FARCuriosity and the vanilla RND use PPO for training a policy, as [8] suggested. The learning rate is 0.0001, the reward discount factor is 0.99 and the number of epochs is 4. For other parameters, we use the same values mentioned in PPO and RND: we set the GAE parameter λ as 0.95, value loss coefficient as 1.0, entropy loss coefficient as 0.001, and clip ratio (ϵ in Eq. (7) in [39]) as 0.1.

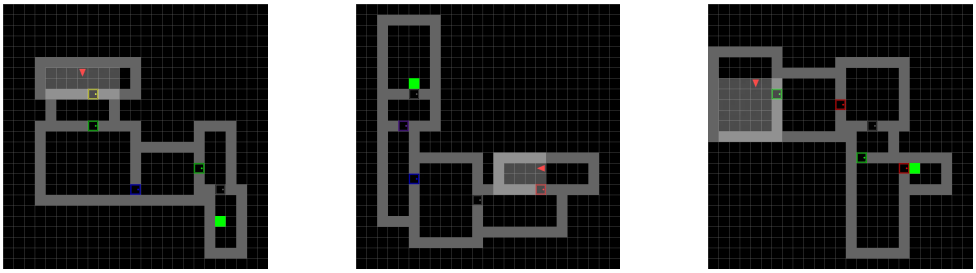


Figure 7: Examples of MultiRoom-N6 in MiniGrid benchmark. The red triangle is an agent and the green rectangular is the goal location. A rectangular with a dot is a door the agent needs to open.

We set the fragmentation parameter ρ to 10, and the recall parameter ψ is 0.99. The maximum size of the LTM is 800, and the least recently used fragment (curiosity module) is substituted by a new fragment if LTM is fully used. For fragmentation, we also add a cosine similarity constraint: if the cosine similarity between features of the current input and any o_f in LTM is higher than 0.75, we skip the fragmentation to avoid excessive fragmentation in high dimensional space. We use a fixed random network in RND as a feature extractor $\phi(\cdot)$ in Section 3.3 and all curiosity modules share the same fixed network.

We run Atari experiments for 100 million frames with three different random trials, and we optimize in batches of 128 frames per environment across 32 parallel environments with minibatch size, 4. We use 16 Atari games: Amidar, Asteroids, Breakout, Freeway, Frostbite, Jamesbond, Montezuma's Revenge, River Raid, Pitfall, Pong, Private Eye, Seaquest, Space Invaders, Star Gunner, Tennis, and Wizard of Wor as shown in Figure 11.

C.2 Toy Grid World

We conducted simple experiments on a toy grid environment to understand why catastrophic forgetting happens even in an environment that is reset multiple times during training. The size of the grid environment is 10×10 and an observation of each location is a 32-dimensional vector sampled from a uniform distribution, $\mathcal{U}(0, 1)$. In every episode, the agent starts from the center of the environment $(5, 5)$ and randomly explores the environment except the start position. The episode length is 200 in a fixed length setting (Figure 4b) and $10n$ at n-th episode in the gradually increasing setting (Figure 4c). Both fixed and training networks in RND are implemented as two fully connected layers with Rectified Linear Unit (ReLU).

We use Adam [23] with a learning rate of 0.001 and RND is sequentially trained on the current observation (the number of the parallel environments is 1 and the batch size is 1) for one million steps. We used three different random seeds and calculate means and standard errors in Figure 4.

C.3 Count-Based Exploration in MiniGrid benchmark

MiniGrid [10] is a procedurally-generated grid world environment with various tasks. The observation is 7×7 egocentric view with three channels (cell types, cell colors, and door states). The number of actions is seven: turn left, turn right, move forward, pick up an object, drop an object, toggle (open/close a door), and done. MultiRoom-N6 is composed of six rooms. An agent needs to navigate from the first room to the goal location in the last room passing other rooms as shown in Figure 7. The agent needs to open (toggle) doors to enter the next room. The episode length $t_{\max}$ is 120 and the extrinsic reward is defined as $1 - 0.9t/t_{\max}$ at time t.

We use MD5 hash for counting the number of visits of each observation and keep updating it across multiple episodes during the training. Since the observation is 7×7 egocentric view following previous works [9, 33], multiple locations can share the same observation and the same intrinsic reward. We set the exponential decay factor, γ as 0.9990, 0.9995, and 1.0000, and run each model with three different trials for 10 million steps. Other hyperparameters are the same as the main experiments.

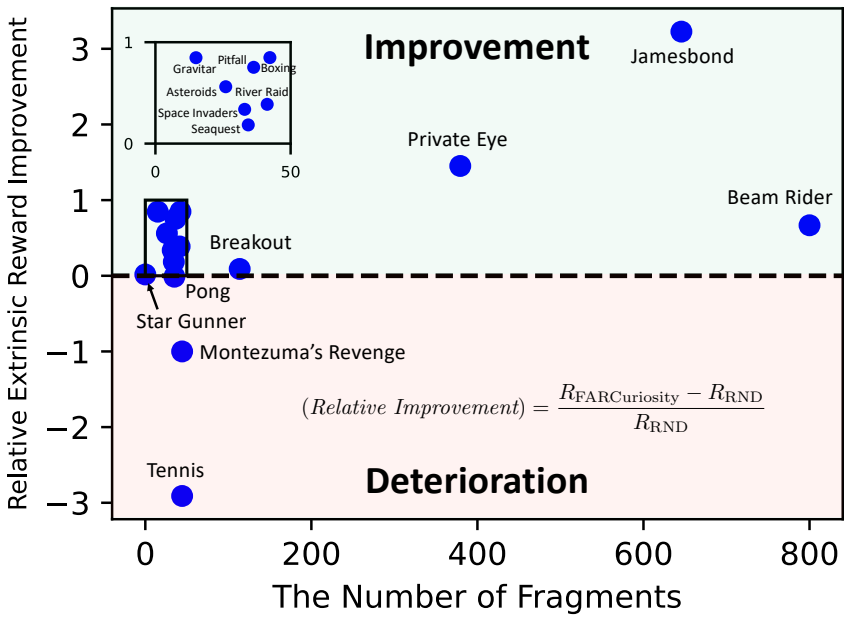


Figure 8: The relative performance improvement over RND performance (R_{RND}) in various Atari games.

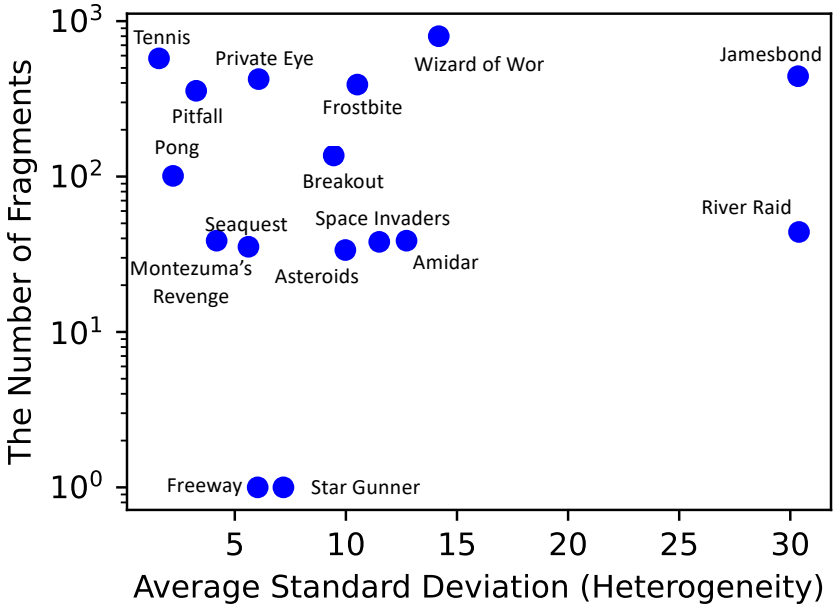


Figure 9: The number of fragments is not related to the heterogeneity of the game.

D Effect of the Number of Fragments

Figure 8 shows the relation between the relative performance improvement and the average number of fragments on multiple Atari games. In most games, the number of fragments is up to 200, but it reaches the maximum capacity of LTM in Wizard of Wor. There is no clear global tendency between performance improvement and the number of fragments. On the other hand, the number of fragments is not related to the heterogeneity of games as shown in Figure 9. This is because we use a dynamic threshold using running statistics of surprisal to avoid over-fragmentation in highly heterogeneous environments. However, it triggers excessive fragmentation in low heterogeneous environments as a trade-off.

Table 1: Sensitivity analysis about fragmentation threshold, ρ and recall threshold, ψ in Jamesbond.

Model	Average Score
PPO [39]	2022.9
RND [8]	1925.6
FARCuriosity ($\rho = 5$)	5339.8
FARCuriosity ($\rho = 5$, ours)	**11917.5**
FARCuriosity ($\rho = 15$)	10407.8
FARCuriosity ($\psi = 0.95$)	**12719.5**
FARCuriosity ($\psi = 0.975$)	11917.5
FARCuriosity ($\psi = 0.99$, ours)	10407.8

E Sensitivity Analysis

We conducted sensitivity analysis for fragmentation and recall thresholds for 50 million frames in Jamesbond as shown in Table 1. While changing one threshold, we fix another threshold. A small fragmentation threshold reduces the performance due to excessive fragmentations while FARCuriosity is robust to the change of recall threshold. However, the proposed method still achieves better performance than baselines.

F Individual Performance in Atari Games

Figure 10 presents the mean cumulative extrinsic rewards (also known as extrinsic return) and its standard errors at varying numbers of training data (frames) over three different random trials in sixteen games including three hard Atari games: Montezuma's Revenge, Pitfall, and Private Eye. No fragmentation occurs in Freeway and Star Gunner, which means that the difference between RND and FARCuriosity in those environments is because of the randomness of reinforcement learning and the environments. FARCuriosity outperforms RND in eight environments. These environments mainly have high heterogeneity except Seaquest. Especially, Jamesbond and River Raid have much more heterogeneity compared to other environments, and FARCuriosity achieves clearly better performance. In Breakout, the performance gap is much smaller. This is because the observations in those environments are more similar, and once the agent hits all bricks, the same bricks are revived. Hence, RND does not suffer from catastrophic forgetting and achieves similar results to FARCuriosity. FARCuriosity shows similar performance in Pitfall, Pong, Star Gunner, and Tennis. We believe this is because the observation space is less visually diverse in those games. In Pong, for example, the only visually dynamic features are the agent, the opponent, and the ball, all of which move over a fixed monotone background. On the other hand, FARCuriosity shows worse performance compared to RND in Montezuma's Revenge. Montezuma's Revenge is one of the most homogeneous games according to our measurement since an agent is difficult to explore many rooms. We hypothesize that this homogeneity can lead to wrong fragmentation, which degrades the performance. There are about 600 fragments in Tennis but it does not change the performance since the environment is easy while Montezuma's Revenge is not. We also calculate IQM [2] as aggregated human-normalized scores across environments. FARCuriosity outperforms its baselines.

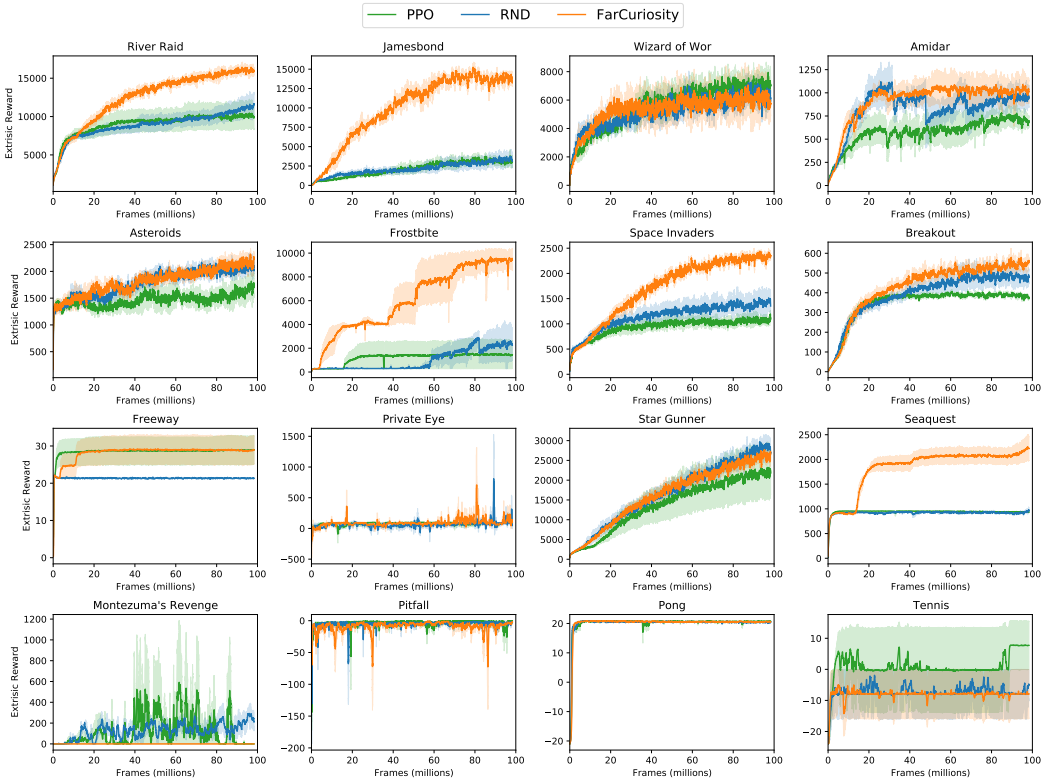
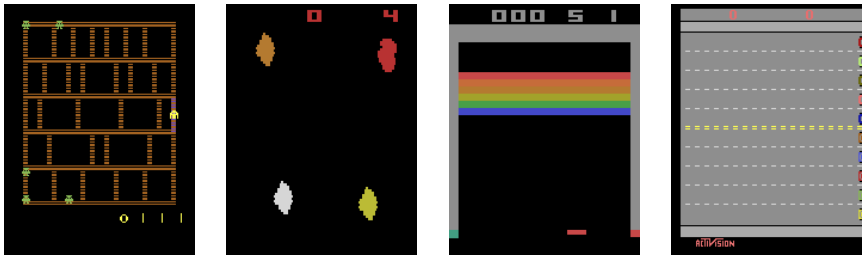


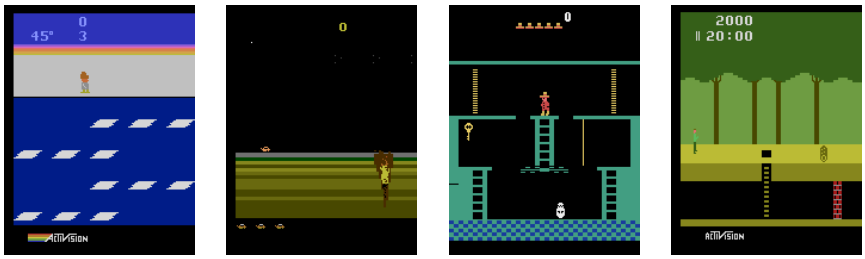
Figure 10: Mean extrinsic reward of RNN-based policies: PPO, RND, and FARCuriosity with extrinsic rewards on sixteen Atari games. The environments are sorted by the heterogeneity measured in Section 4: from highest to lowest.

Table 2: The average score of each method in Atari games with their heterogeneity. The numbers in parentheses are human-normalized scores where human performance is acquired by Badia *et al.* [4].

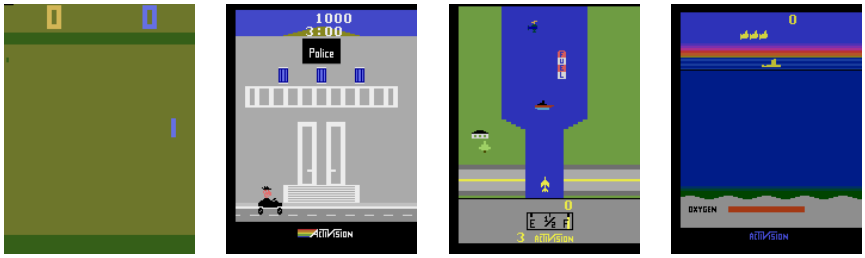
Game	Heterogeneity	PPO	RND	FARCuriosity
River Raid	30.4	9860.9 (54)	11659.3 (65.4)	**15923.6 (92.4)**
Jamesbond	30.4	3060.7 (1107.3)	2946.2 (1065.4)	**13073.0 (4764.1)**
Wizard of Wor	14.2	**7392.0 (162.9)**	6123.0 (132.6)	5782.3 (124.5)
Amidar	12.7	648.0 (37.5)	963.8 (55.9)	**1012.9 (58.8)**
Asteroids	11.5	1724.6 (2.2)	2028.5 (2.8)	**2219.7 (3.2)**
Frostbite	10.5	1444.7 (32.3)	2282.0 (51.9)	**9392.1 (218.5)**
Space Invaders	10.0	1088.7 (61.9)	1352.6 (79.2)	**2347.9 (144.7)**
Breakout	9.5	377.3 (1304.2)	478.5 (1655.6)	**563.5 (1950.7)**
Freeway	7.2	**29.0 (98.0)**	21.3 (72.0)	25.1 (84.8)
Private Eye	6.1	97.0 (0.1)	47.6 (0.0)	**97.0 (0.1)**
Star Gunner	6.0	25982.7 (264.1)	**27063.7 (275.4)**	26491.3 (269.4)
Seaquest	5.6	936.7 (2.1)	961.6 (2.1)	**2218.8 (5.1)**
Montezuma Revenge	4.2	0.0 (0.0)	**210.3 (4.4)**	0.0 (0.0)
Pitfall	3.3	**-0.7 (3.4)**	-4.1 (3.4)	-1.0 (3.4)
Pong	2.2	**20.8 (117.6)**	20.5 (116.7)	20.6 (117)
Tennis	1.6	**7.8 (203.9)**	-5.1 (120.6)	-8.0 (101.9)
IQM [2]		0.69	0.68	**0.91**



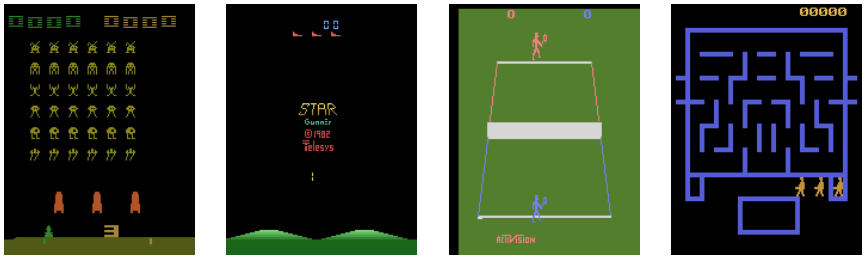
(a) Amidar (b) Asteroids (c) Breakout (d) Freeway



(e) Frostbite (f) Jamesbond (g) Montezuma's Revenge (h) Pitfall



(i) Pong (j) Private Eye (k) River Raid (l) Seaquest



(m) Space Invaders (n) Star Gunner (o) Tennis (p) Wizard of Wor

Figure 11: Various Atari environments used for comparing FARCuriosity and RND.