Summary of the paper:

EXPLORATION BY RANDOM NETWORK DISTILLATION

Ignacio Monardes

January 26, 2026

Reference

Authors: Yuri Burda, Harrison Edwards, Amos Storkey, Oleg Klimov. OpenAI team.
Title: *EXPLORATION BY RANDOM NETWORK DISTILLATION*
Conference/Journal:
Link: `https://arxiv.org/pdf/1810.12894`

1 Motivation

Current problem

There insufficient exploration in sparse-reward environments with long horizons. The agent learns that every action leads to zero reward. Some algorithms try to use prediction error of the environment dynamics to encourage exploration, but they tend to exploit stochastic transitions.

Relevance of the problem

Agents cannot solve such tasks because they learn that everything they can do yields zero reward, so there is no incentive to explore new areas and they fail to obtain useful rewards.

Hole in the literature / previous limitations

There are methods that introduce novelty-based bonus measured with complex structures to avoid noise and getting stuck on stochastic transitions. These approaches are often difficult to implement and still fail on many Atari games such as Montezuma's Revege or Freeway.

2 Idea

Main Idea

The paper defines the reward as the sum of the extrinsic reward e_t (from the environment) and an intrinsic reward measured by the novelty of the state i_t.

$$r_t = e_t + i_t$$

To encourage i_t to be novelty one try to approximate count-based exploration methods. In non-tabular cases it is not straightforward to produce counts. An alternative is to define i_t as the prediction error for a problem related to the agent's transitions. Examples includes forward dynamics ($s_{t+1} = f(s_t, a_t)$) and inverse dynamics ($s_t = f(s_{t+1}, a_t)$).

Innovation

The proposal uses two networks: a fixed (random) target network $f : O \to \mathbb{R}^k$ that maps observations to embeddings, and a predictor network $\hat{f} : O \to \mathbb{R}^k$ that is trained to match the target. The predictor is trained with gradient descent to minimize the mean squared error:

$$\min_{\theta} \quad \text{MSE } \|\hat{f}(x;\theta) - f(x)\|^2$$

Sources of prediction error include:

- Limited training data (few examples seen by the predictor) - this is desirable as a reward signal.
- Stochasticity in the environment (Noisy TV).
- Model misspecification - missing important information or limited model capabilities.
- Learning dynamics - the predictor fails to approximates the target function.

The stochasticity and the model misspecification can be solved by using a deterministic (random but fixed) target network.

3 Methodology

Modeling

The reward is defined as the sum of the intrinsic reward and the extrinsic reward. We can fit two value heads V_E and V_I separately using their respective returns. The extrinsic reward is stationary and the intrinsic reward is non-stationary.

Assumptions

The intrinsic reward mus be normalized because it may be on a different scale than the extrinsic reward. In the paper the intrinsic reward is divided by a running estimate of its standard deviation. Observations are normalized by subtracting the running mean value and dividing by the running standard deviation to each dimension. Then this value is clipped to [-5, 5]. These running statistics are collected during a set of initial steps before training begins.

Architecture

4 Experiments

Baselines

The main baseline is PPO without intrinsic exploration. An useful comparison is PPO with some kind of intrinsic reward dervied from forward or backward dyanmics prediction error.

Environments and datasets

First, the authors evaluate RND without extrinsic reward. They define two measures of exploration: mean episodic return and the number of rooms the agent finds.

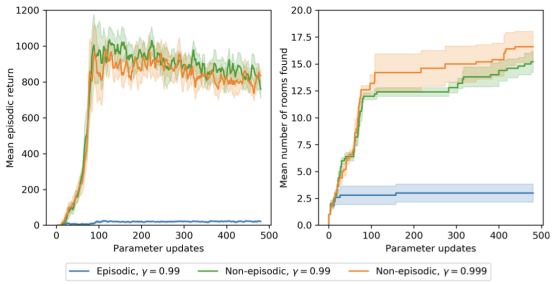
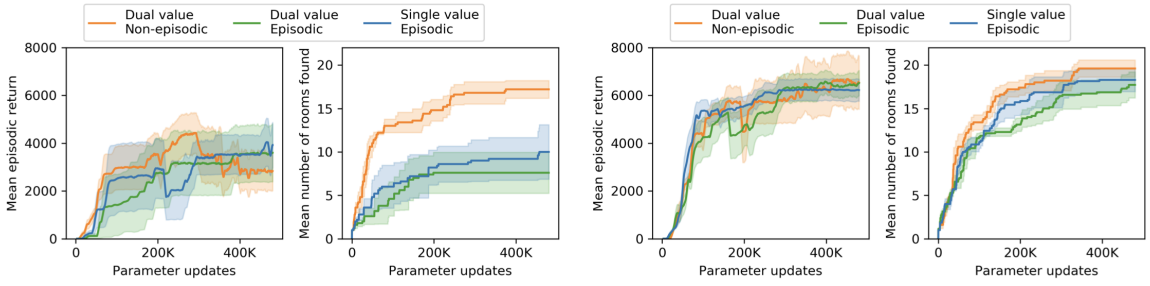


Figure 3: Mean episodic return and number of rooms found by pure exploration agents on Montezuma's Revenge trained without access to the extrinsic reward. The agents explores more in the non-episodic setting (see also Section 2.3)

Next they combine intrinsic reward with the extrinsic reward, and evaluate both episodic and non-episodic intrinsci reward formulations together with episodic extrinsic rewards. intrinsic reward combinaed with episodic extrinsic reward.



(a) RNN policies (b) CNN policies

Figure 4: Different ways of combining intrinsic and extrinsic rewards. Combining non-episodic stream of intrinsic rewards with the episodic stream of extrinsic rewards outperforms combining episodic versions of both steams in terms of number of explored rooms, but performs similarly in terms of mean return. Single value estimate of the combined stream of episodic returns performs a little better than the dual value estimate. The differences are more pronounced with RNN policies. CNN runs are more stable than the RNN counterparts.

Settings

The main settings is to use:

- Intrinsic discount $\gamma_I = 0.99$ for non-episodic rewards
- Extrinsic discount $\gamma_E = 0.999$ with episodic rewards.
- An RNN policy.
- 1024 parallel environments.

5 Results

	Gravitar	Montezuma's Revenge	Pitfall!	PrivateEye	Solaris	Venture
RND	**3,906**	**8,152**	-3	8,666	3,282	**1,859**
PPO	3,426	2,497	0	105	3,387	0
Dynamics	3,371	400	0	33	3,246	1,712
SOTA	2,209[1]	3,700[2]	**0**	**15,806**[2]	**12,380**[1]	**1,813**[3]
Avg. Human	3,351	4,753	6,464	69,571	12,327	1,188

Table 1: Comparison to baselines results. Final mean performance for various methods. State of the art results taken from: [1] (Fortunato et al., 2017) [2] (Bellemare et al., 2016) [3] (Horgan et al., 2018)

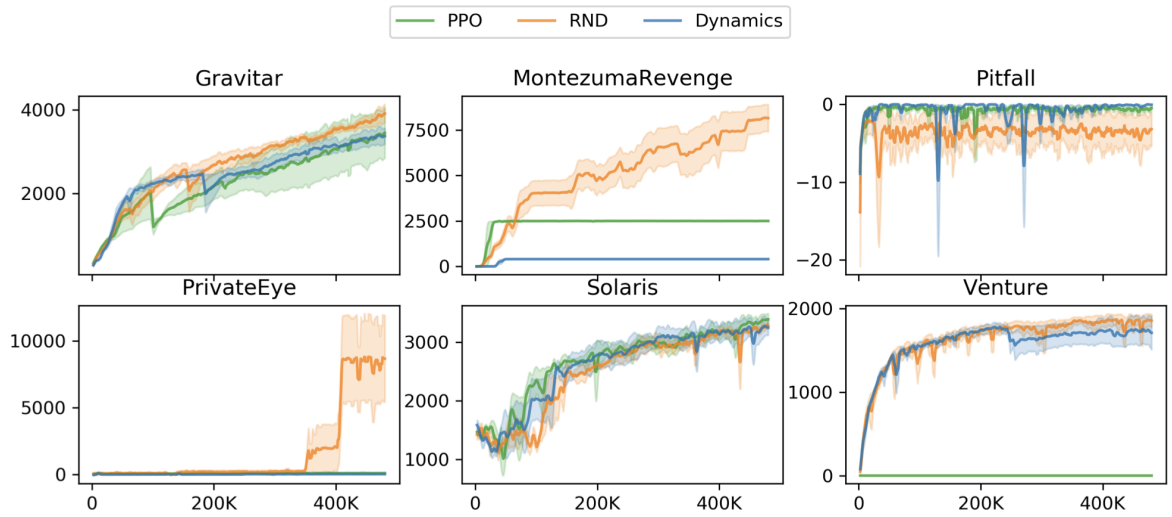


Figure 7: Mean episodic return of RNN-based policies: RND, dynamics-based exploration method, and PPO with extrinsic reward only on 6 hard exploration Atari games. RND achieves state of the art performance on Gravitar, Montezuma's Revenge, and Venture, significantly outperforming PPO on the latter two.

The agent performs better and achieves state of the art on Montezuma's Revenge, Private Eye and Gravitar. It is equally good in Gravitar, Solares and Venture. It is a little worse in Pifall because the extrinsic reward is very sparse.

An interesting thing of the Dynamics model is that the agent tends to move between two rooms because it is difficult to detect the border and makes a huge difference in the prediction of the next state.

Another interesting qualitative analysis is that when the agent achieves all the known extrinsic rewards the agent starts to jump/move around dangerous objects. This behaviour is related to the fact that dangerous states are difficult to achieve and rarely represented in the experiences.

Good and bad performance in different scenarios

The performance is much better

6 Critical Discussion

Strength

RND simple to implement and achieves state of the art on difficult environments. It is less susceptible to the Noisy TV because it doesn't attempt to directly to predict future or past observation.

Weakness

RND doesn't solve the first level of Montezuma's Revenge consistently. For unknown reasons it is worse on Pitfall than the others algorithms.

Questionable assumptions

Using intrinsic rewards not only increasesexploration, but also incentivizes searching for novel states. The deterministic target network might be to rigid for highly stochastic environments.

What it is not clear

- The environment of Montezuma's Revenge is deterministic, I don't know if the others environments used have some stochasticity to see how it performs.
- Why it is much worse on the Pitfall environment than the other algorithms if all the algorithms suffer from sparse rewards.

  The environment has extremely long horizons, with mostly penalty rewards. The states are visually similar. Exploration requires very precise sequences.
- I don't know if the mean episodic return is made by the extrinsic reward or the intrinsic reward. I think that it might be the extrinsic reward because that it is what we really try to improve.

  The reported mean episodic return refers to extrinsic reward only.
- Why the RNN captures information that the CNN does not captures.

  The RNN captures the history, the CNN only uses the current frame.
- Is the RNN applied as the predictor function or as an embedding of the state?

  The RNN is used in the policy and value networks. It is a memory.

- How are the environments of Gravitar, Pitfall, Private Eye, Solaris and Venture? Why are they difficult?

  Gravitar has sparse rewards, precise control, and deadly obstacles.

  Pitfall has almost zero rewards. Penalizes mistakes and requires memory.

  Solaris is a large map with sparse signals.

  Venture is a multiple rooms with enemies and sparse rewards.

7 Conclusions

Personal opinion

¿Is it a good contribution?

Absolutely. It achieves state of the art on Montezuma's Revenge and solves the first level some times. This game is very difficult, I had tried it and I couldn't get to many rooms.

¿Would I use it on my investigation?

Yes, it seems as a strong baseline against dreamerv3.

Useful link

- **OpenAI implementation:** `https://github.com/openai/random-network-distillation/blob/master/run_atari.py`
- **OpenAI RND web page:** `https://openai.com/index/reinforcement-learning-with-prediction`
- **Small implementation:** `https://github.com/wisnunugroho21/reinforcement_learning_ppo_rnd/blob/master/PPO_RND/pytorch/ppo_rnd_pytorch.py`
- **CleanRL implementation:** `https://github.com/vwxyzjn/cleanrl/blob/master/cleanrl/ppo_rnd_envpool.py`