Exploration by Random Network Distillation

Yuri Burda* OpenAI
Harrison Edwards* OpenAI
Amos Storkey Univ. of Edinburgh
Oleg Klimov OpenAI

arXiv 2018

Total cites : 393

2021.07.07

최영제

Index

1. Introduction

2. Method

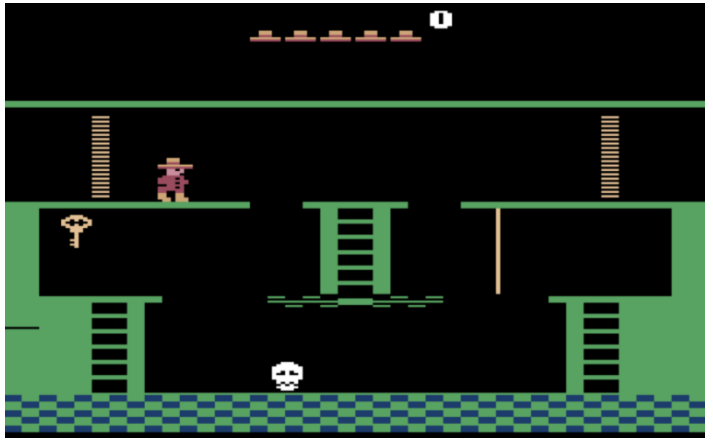
3. Experiments

1 Introduction

Dense/Sparse reinforcement learning

- Dense reward setting vs Sparse reward setting
- Reward가 드물게 발생하거나, 지연되어 발생하면 학습에 어려움이 존재
- 희소 보상 문제(sparse reward problem)로 인해 Atari – Montezuma’s revenge는 사람의 성능을 넘지 못함
- 실제 환경은 Montezuma’s revenge 보다 더욱 더 sparse reward 환경이 多

Atari – Demon attack (Dense)

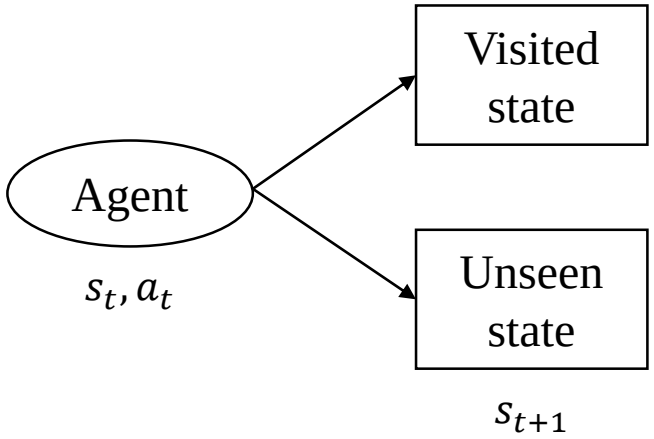


Atari – Montezuma's revenge (Sparse)

사다리, 해골 회피, … → Reward 0
열쇠 획득 → + Reward

Exploration bonus

- Sparse reward problem을 해결하는 방법은 exploration을 장려하여 reward를 획득한 episode를 충분히 경험하는 것
- Exploration을 장려하는 방법은 크게 두 가지가 존재함
    1) 환경을 parallel하게 구성하여 다양한 경험들을 얻고 이를 토대로 업데이트 하는 방법
    2) 단일 환경에서 새롭게 탐색된 state에 대해서 추가적인 reward를 부여하는 방법(exploration bonus)
- Count-based exploration bonus



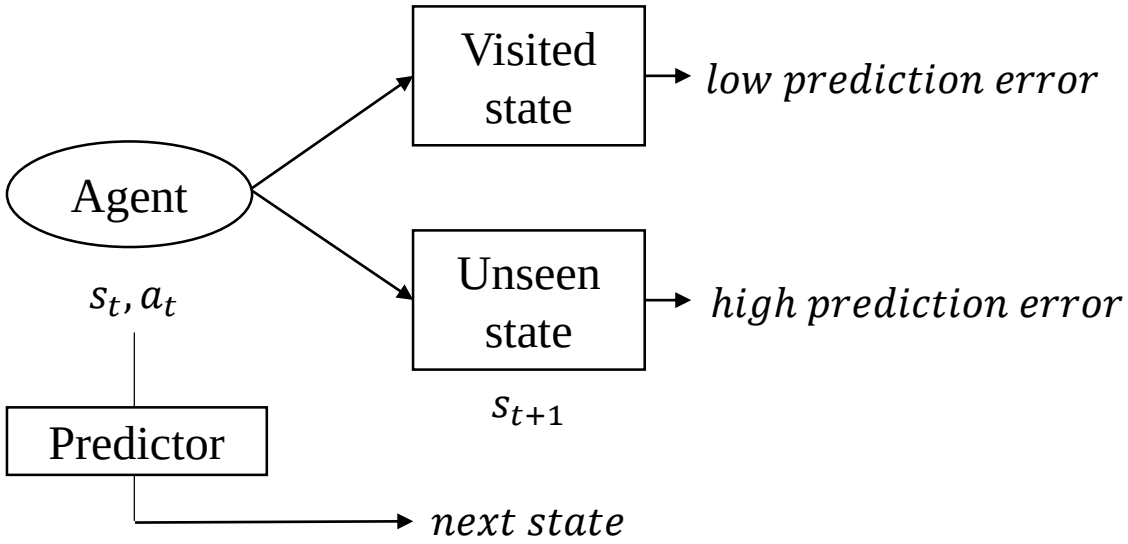
* 모델 내부에서 발생하는 reward: intrinsic reward
* 모델 외부에서 발생하는 reward: extrinsic reward

$$\text{Intrinsic reward} = \frac{1}{n_t(s)}$$

Introduction

Curiosity

- Memory를 사용하는 것은 비효율적이며 continuous 환경에는 적용이 까다로움
- Curiosity는 현재 state와 action을 바탕으로 next state를 예측 후 발생하는 오차를 사용하여 exploration을 강제하는 방법
- 기존에 많이 방문한 state의 경우 낮은 예측 오차를, 새롭게 방문한 state는 높은 예측 오차를 갖게 됨
- 이를 intrinsic reward로 활용하여 exploration을 장려함



Intrinsic reward = prediction error

Curiosity

- Curiosity는 intrinsic reward만으로도 Montezuma’s revenge의 room을 절반 이상 clear함
- 단 curiosity는 next state를 잘 예측해야하는 문제를 갖고 있음
- 만약 state 안에 noise가 발생할 경우 exploration을 전혀 하지 않더라도 높은 prediction error를 갖는 경우가 발생함

Noisy TV problem

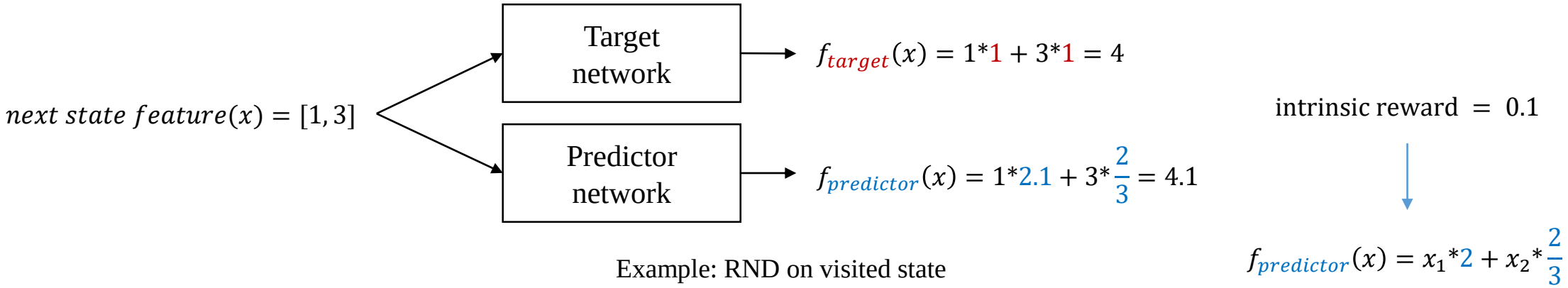
2 SIL Method

Random Network Distillation

- RND는 intrinsic reward를 구축함에 있어 next state를 예측하지 않고 feature를 추출하는 방법으로 위 문제를 해결
- RND는 두 개의 network를 사용
  1) Target network: fixed and randomly initialized network (trainable = False)
  2) Predictor network: target network가 추출한 feature와 동일한 feature를 갖도록 학습함 Minimize: $\left\| f_{target}(x) - f_{predictor}(x) \right\|^2$

  * This process distills a randomly initialized network into a trained one

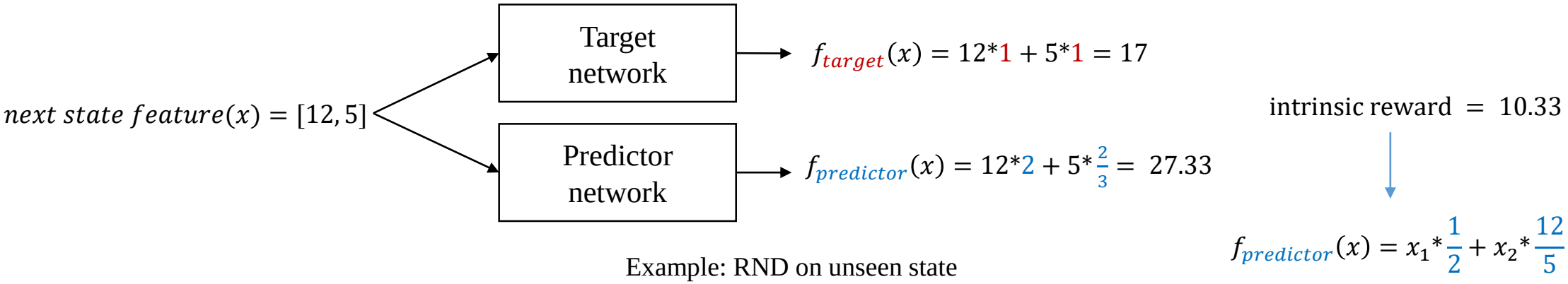
- Visited state에 대해서 최적화된 predictor network의 예시는 다음과 같음



Example: RND on visited state

Random Network Distillation

- Unseen state에 대한 intrinsic reward 예시는 다음과 같음



Example: RND on unseen state

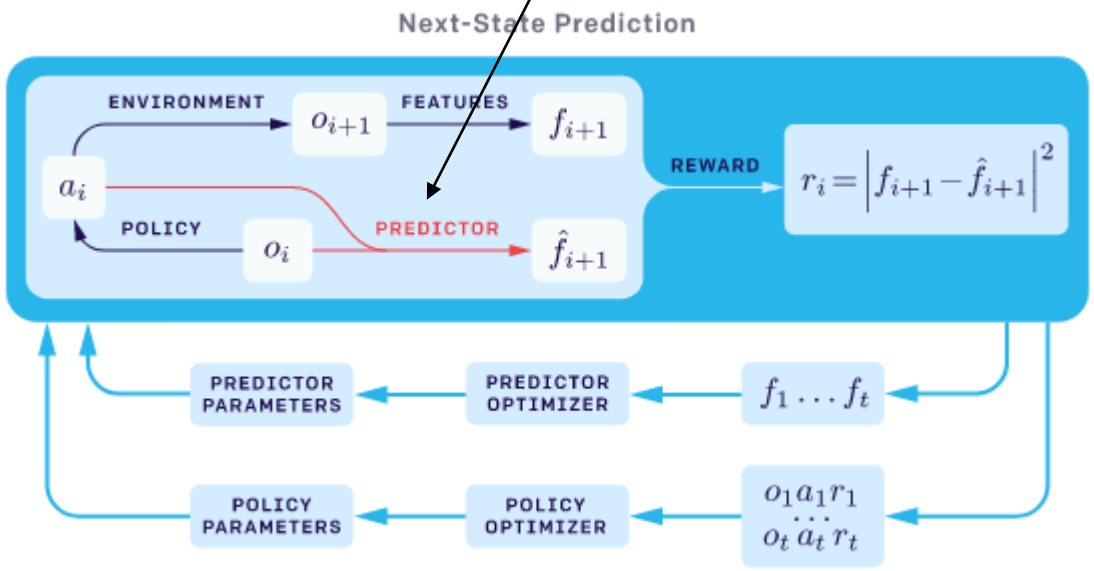
- Target network와의 출력값이 동일하도록 학습하면 두 network의 가중치가 100% 동일해지지 않나?

MNIST 데이터를 학습시키는 CNN 모델만 생각해도 완전히 Overfitting되는 열 개의 모델을 만들더라도 각기 Neural Network의 weight는 다르다는 것을 생각해보면 직관적으로 절대 같아질 수 없다는 것을 알 수 있습니다.

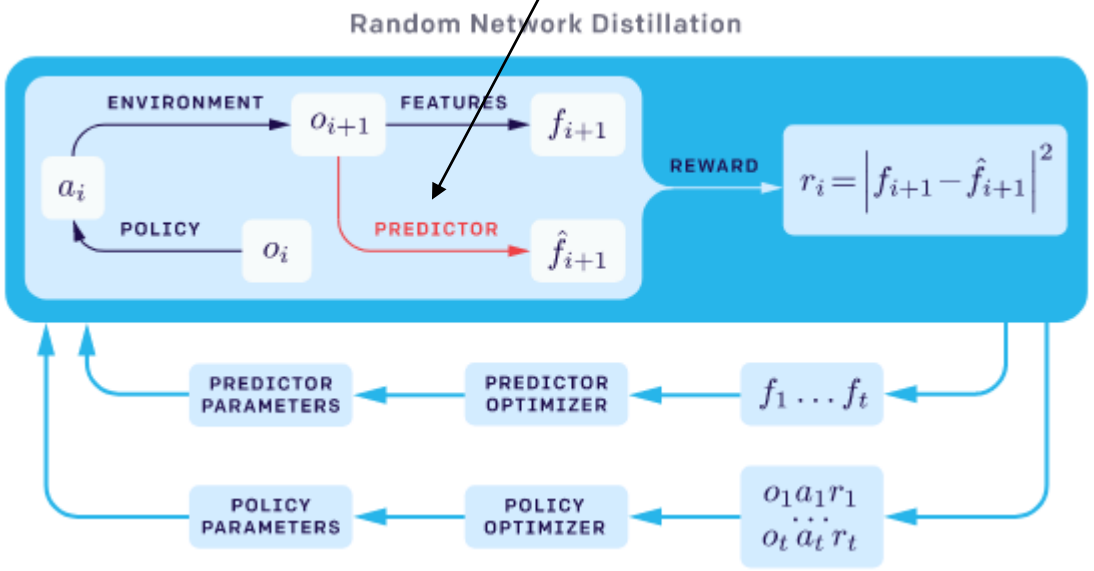
Method

Comparison of Next-State Prediction with RND

Observation과 action을 이용하여 next observation을 예측



Next observation으로부터 feature를 추출 후 비교



RND code

```
target_network = RND_network(state_idx)
prediction_network = RND_network(state_idx)
target_network.trainable = False

                                                                    …

for episode in range(num_episode) :

                                                                    …

    while True :
        state, reward, done, next_state = env.step(action)
        ## rnd networks update
        with tf.GradientTape() as tape:
            target_sf = target_network(next_state) # target feature 추출
            prediction_sf = prediction_network(next_state) # predictor feature 추출
            predictor_loss = 0.5 * tf.square(tf.stop_gradient(target_sf) - prediction_sf)
            predictor_loss = tf.reduce_mean(predictor_loss)
            grads = tape.gradient(predictor_loss, prediction_network.trainable_variables)
            optimizer.apply_gradients(zip(grads, prediction_network.trainable_variables)) # target network와 동일하도록 predictor network를 학습

        ## intrinstic reward
        int_reward = tf.norm(target_sf - prediction_sf)
        reward += int_reward/2 # extrinsic reward에 intrinsic reward를 더함

                                                                    …
                                                    PPO, TRPO, A2C, DQN, DDQN 등 학습
```

3 Experiments

Novelty detection on MNIST

- Feature를 추출하여 비교하는 것만으로 exploration을 장려할 수 있는가?

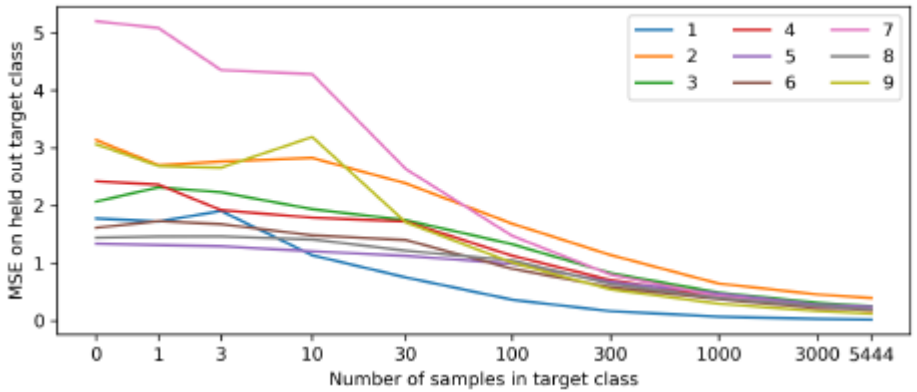


Figure 2: Novelty detection on MNIST: a predictor network mimics a randomly initialized target network. The training data consists of varying proportions of images from class "0" and a target class. Each curve shows the test MSE on held out target class examples plotted against the number of training examples of the target class (log scale).

Y축: Target network – Predictor network

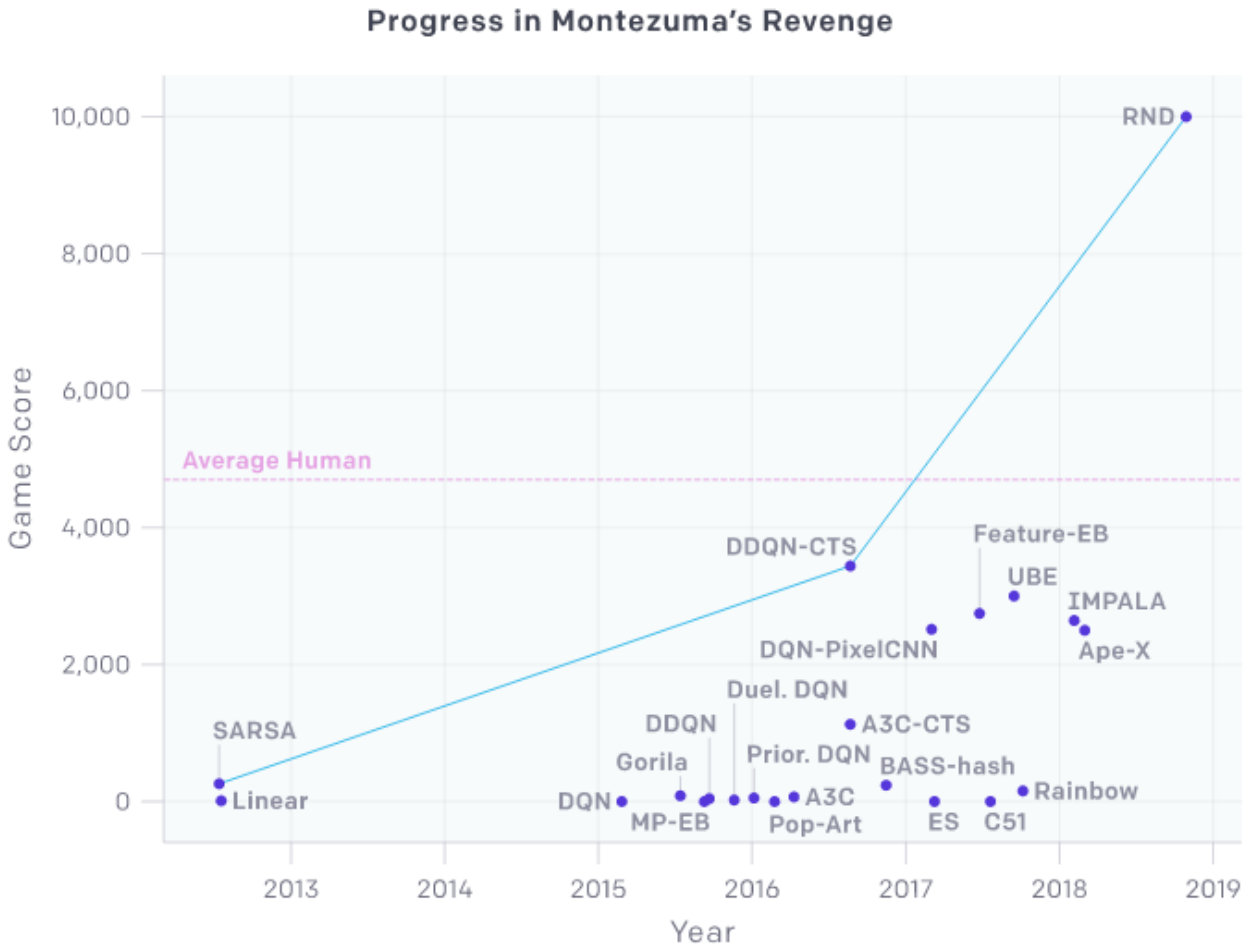
X축: Class 1-9 각각에 대한 sample 개수

Class 0: Visited state의 역할 (predictor network를 target network와 같은 featur를 추출하도록 학습)

Unseen state (X축 전반부)에 대해서 높은 novelty
Visited state (X축 후반부)에 대해서 낮은 novelty

Experiments

Performance on Atari – Montezuma's revenge



3 Experiments

Performance Comparison of PPO,Next-State Prediction and RND

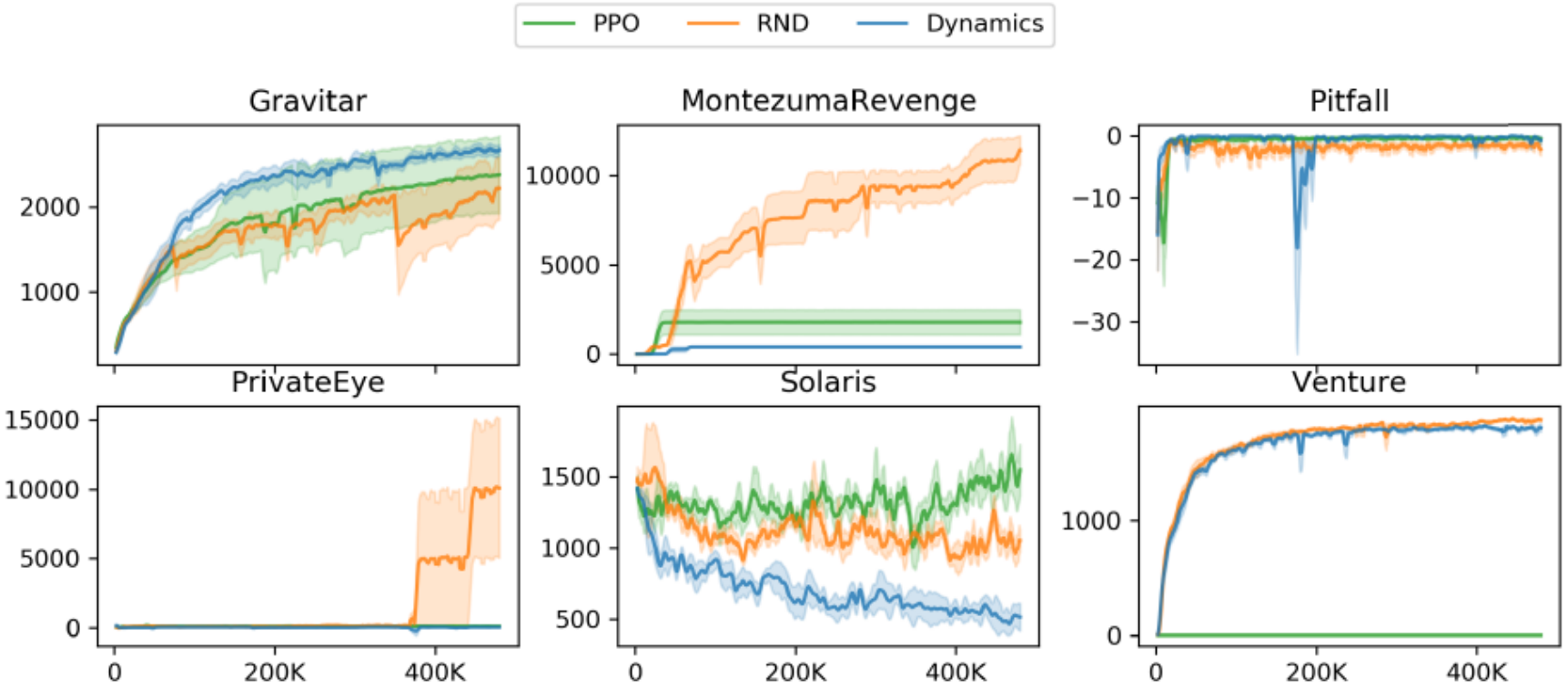


Figure 9: Mean episodic return of CNN-based policies: RND, dynamics-based exploration method, and PPO with extrinsic reward only on 6 hard exploration Atari games. RND significantly outperforms PPO on Montezuma's Revenge, Private Eye, and Venture.

Q&A