A Decision Metric for the Use of a Deep Reinforcement Learning Policy

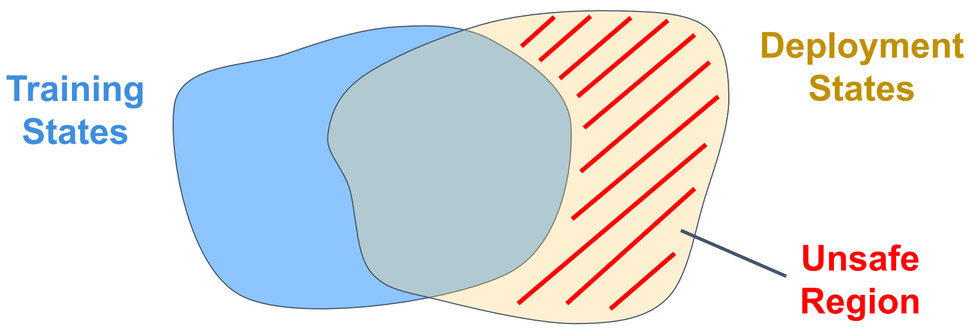
ICML Workshop on Responsible Decision Making in Dynamic Environments
July 23, 2022

Christina Selby
Edward W. Staley
Johns Hopkins University
Applied Physics Lab (APL)

When can we trust a DRL policy?

Policies learned with deep reinforcement learning (DRL) can be extremely effective in well-defined environments.

However, a deployed policy may experience distribution shift, in which the encountered states are out-of-distribution (OOD) from the training environment.

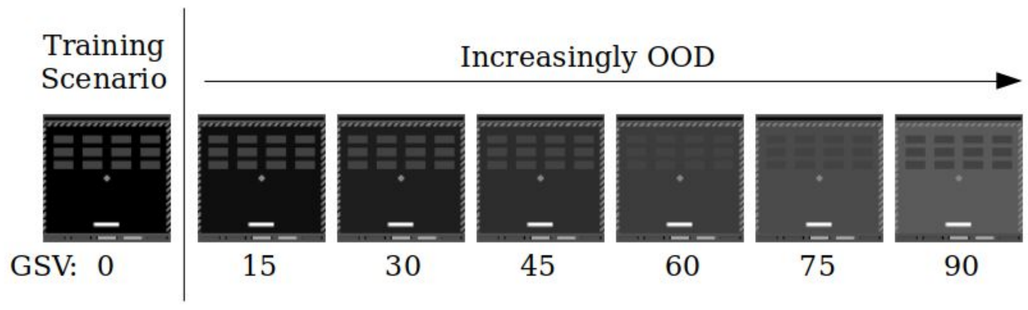


To help assure safety, we define OOD states as being **unsafe**, and wish to create a statistical metric which can be used to switch off the DRL policy when these are encountered.

Experiment Setup

We train a PPO policy on Meta Arcade breakout, a simple video game environment which exposes access to the underlying game configuration.

We train on a black background, and test our policy under various shifts away from black.



We seek to define a decision metric that can be used to halt a policy in OOD regions, with minimal impact to operating in known states.

Modified RND for Measuring State Uncertainty

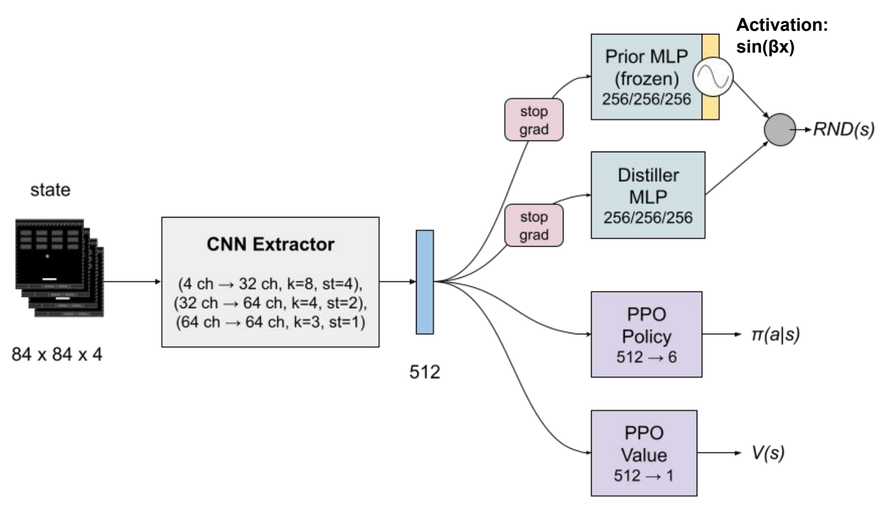
Random Network Distillation (RND) is a method to measure unfamiliarity of encountered states during RL training, originally developed as a method of assigning exploration bonuses to a learning agent.

We adapt RND for use post-training, where it functions as an uncertainty metric for states during deployment. To increase the fidelity of this metric, we make a critical modification to RND: the final activation of the prior network is **sin(βx)**. Adjusting β allows us to adjust the complexity (frequency) of the prior over the state space, and therefore the sensitivity of the RND.

Training Details

- PPO implementation from Stable Baselines 3
- Trained for 20 million timesteps, 8 parallel workers, performing rollouts of 128 each
- Standard preprocessing for Atari (four frames stacked, grayscale)
- RND is trained on collected states from each iteration of PPO

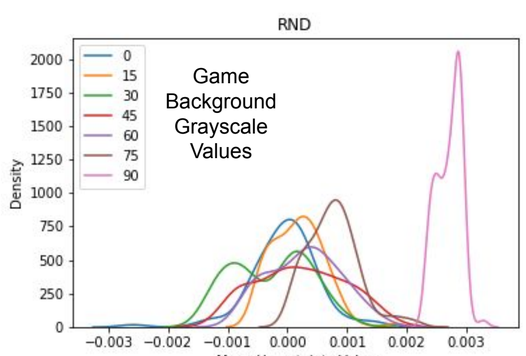
Network Architecture



Architecture for parallel training of PPO (purple) and RND (green), which itself consists of a frozen prior network and learned distiller. A feature extractor learned under PPO is also used as a feature extractor for the RND networks, but does not receive weight updates from the RND.

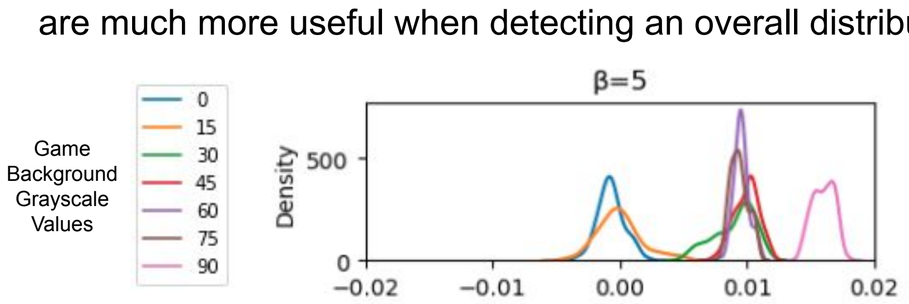
Detecting Distribution Shift

After training in our known environment (black background), we examine the distribution of uncertainty values from the RND in new environments:



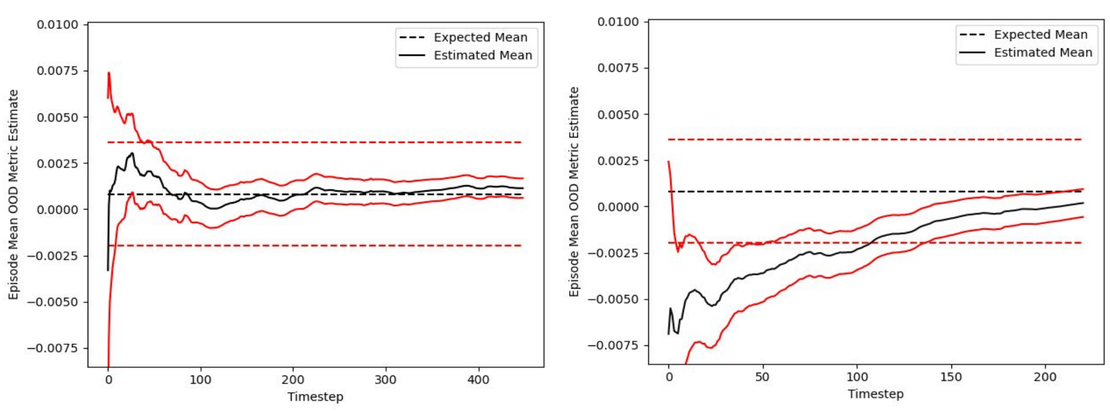
Without our added sinusoidal prior, the distributions do not reflect OOD until the distribution shift is severe (i.e. pink curve)

However, for properly tuned β, we can learn RND networks which are much more useful when detecting an overall distribution shift:



Building a Decision Metric

We can use the distributions on the left to calibrate a decision metric for a DRL agent in deployment. We estimate the mean uncertainty using a Kalman filter as the agent plays a game. If after some period of time we find that the observed RND output is not statistically consistent with those of the training environment, we can switch off the DRL in favor of a safer control scheme:



Left (grayscale=0): In the training environment, our mean uncertainty is within acceptable bounds.
Right (grayscale=15): At t=50, our RND statistics do not match the known environment, and we would turn off the policy.

Correspondence to: Christina Selby (christina.selby@jhuapl.edu), Edward W. Staley (edward.staley@jhuapl.edu)