

1706.01905 - Parameter Space Noise for Exploration

Parameter Space Noise for Exploration

Matthias Plappert^{†‡}, Rein Houthooft[†], Prafulla Dhariwal[†], Szymon Sidor[†],
Richard Y. Chen[†], Xi Chen[†], Tamim Asfour[‡], Pieter Abbeel[†], and Marcin Andrychowicz[†]

[†] OpenAI

[‡] Karlsruhe Institute of Technology (KIT)

- **Yunqiu Xu**
- OpenAI's NoisyNet
 - [OpenAI's blog](#)
 - [Talk on Vimeo](#)
- Further readings which can be seen in my notes:
 - 1703.09327 - DART- Noise Injection for Robust Imitation Learning
 - Add noise to off-policy imitation learning
 - **1706.10295 - Noisy Networks for Exploration**
 - Very similar work by DeepMind
 - It seems that their work is more robust for that it's not restricted to Gaussian noise
 - 1710.02298 - Rainbow: Combining Improvements in Deep Reinforcement Learning
 - A combination of different DQN including NoisyNet

1. Introduction

- Challenges: effective and efficient exploration
- This work:

- Investigate how parameter space noise can be combined with DRL to improve exploration behavior
- Can be applied on both off-policy (DQN, DPPG) and on-policy (TRPO) methods

2. Parameter Space Noise for Exploration

- Overview:
 - Change current policy with Gaussian noise

$$\hat{\theta} = \theta + N(0, \sigma^2 I)$$
 - The perturbed policy is sampled at the beginning of each episode
 - During an episode, the parameters of perturbed policy keep fixed
- State-dependent exploration
 - Action space noise : independent with current state
 - Parameter space noise :
 - Perturbed at the beginning of each episode and keep fixed within episode
 - **Ensure consistency in actions** → **dependence between state and action**
- Use layer normalization between perturbed layers
- Adapt the scale of the parameter space over time and relate it the variance in action space
- This paper also shows parameter space noise for off-policy and on-policy methods, which I think is similar to DeepMind's work but is more abstract and complex

3. Experiments

- Questions:
 - (i) Do existing state-of-the-art RL algorithms benefit from incorporating parameter space noise?
 - (ii) Does parameter space noise aid in exploring sparse reward environments more effectively?
 - (iii) How does parameter space noise exploration compare against evolution strategies with respect to sample efficiency?

3.1 Performance on Games

- DQN results: discrete action space

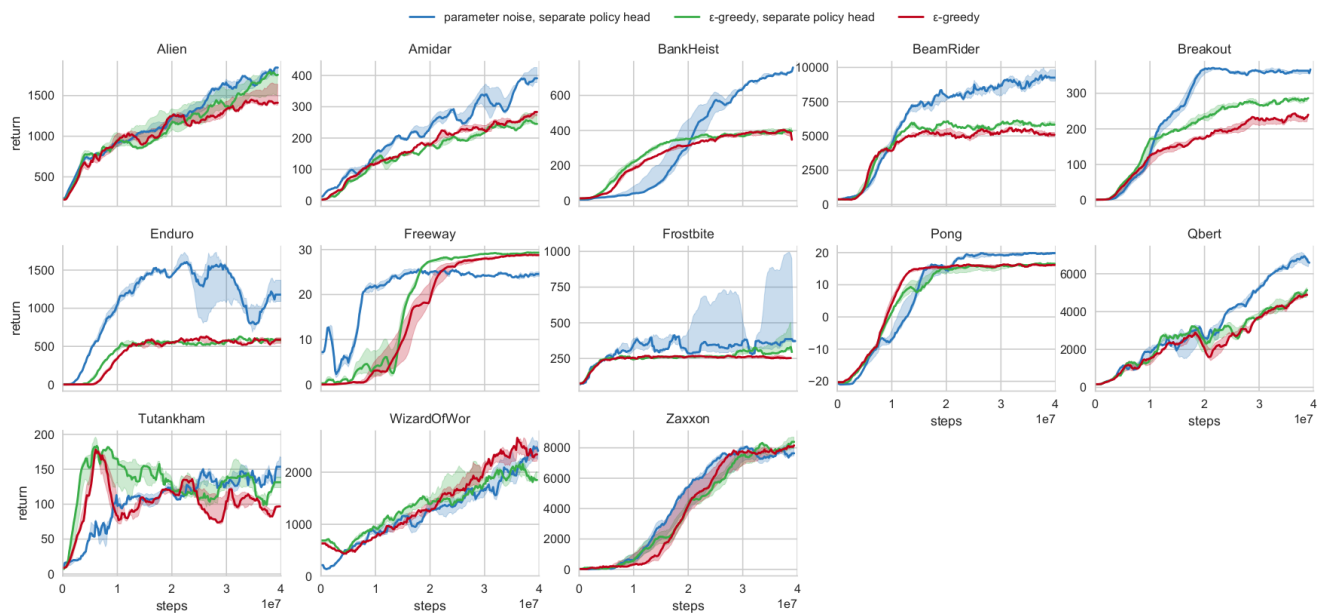


Figure 1: Median DQN returns for several ALE environment plotted over training steps.

- DDPG results: continuous action space

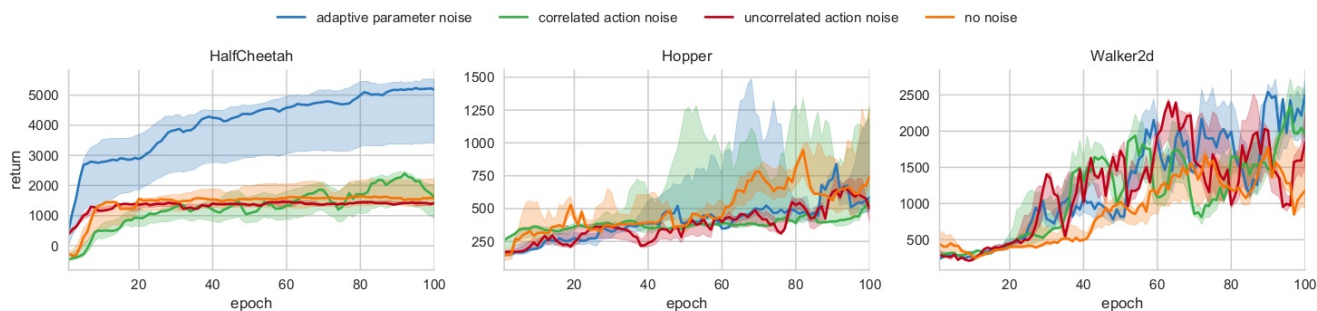


Figure 2: Median DDPG returns for continuous control environments plotted over epochs.

- TRPO results: continuous, on-policy

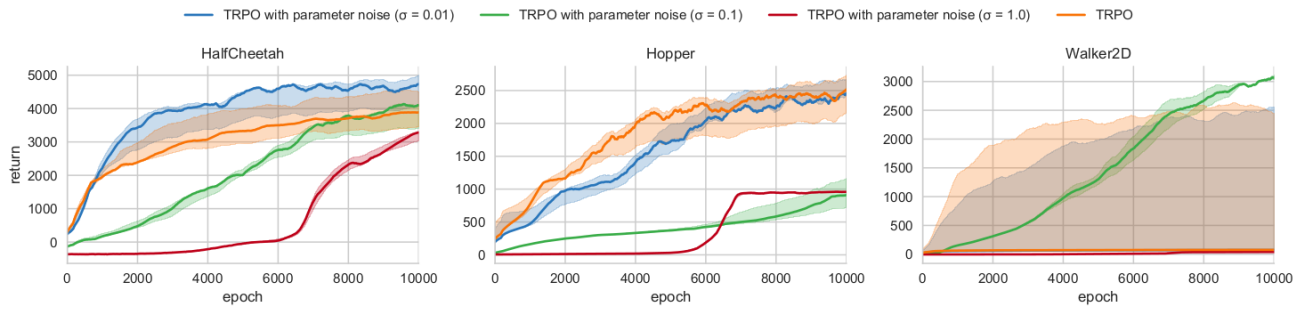


Figure 3: Median TRPO returns for continuous control environments plotted over epochs.

3.2 Explore Efficiency

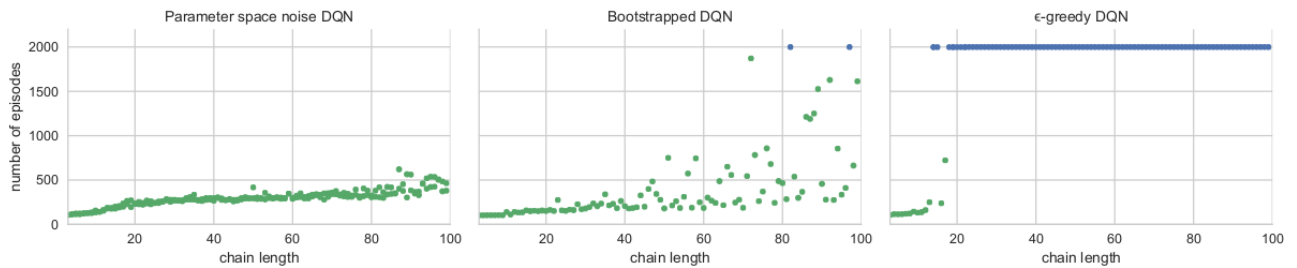


Figure 4: Median number of episodes before considered solved for DQN with different exploration strategies. Green indicates that the problem was solved whereas blue indicates that no solution was found within 2 K episodes. Note that less number of episodes before solved is better.

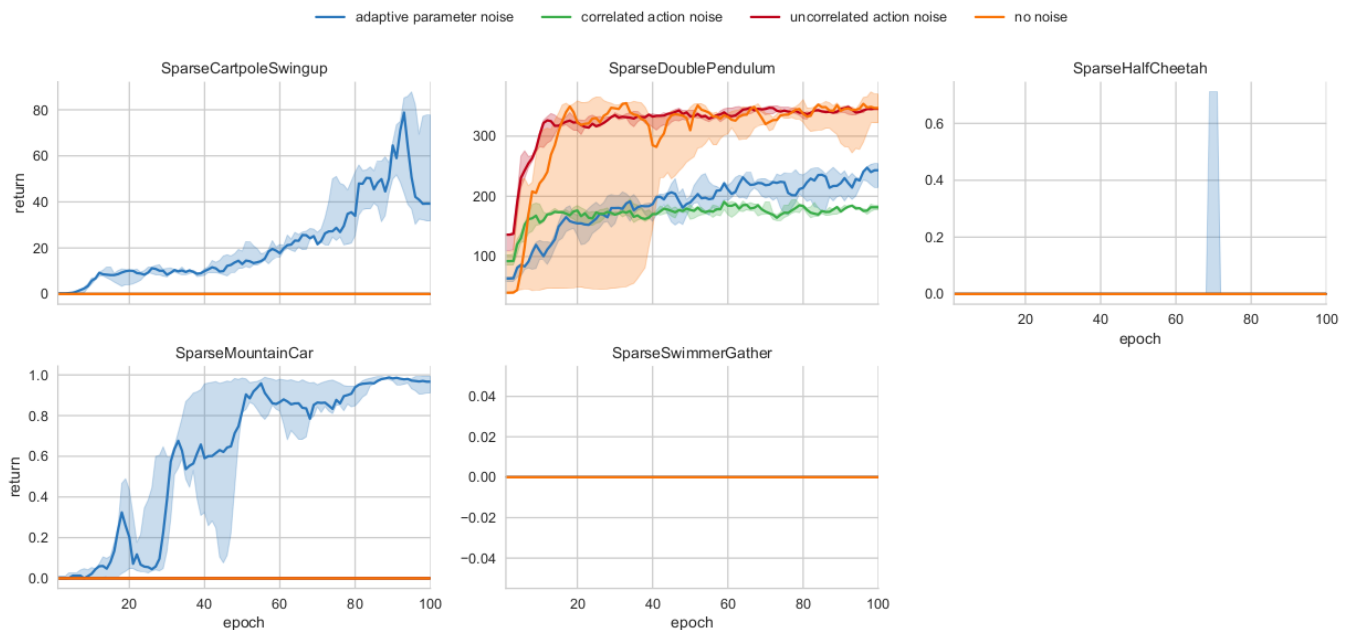


Figure 5: Median DDPG returns for environments with sparse rewards plotted over epochs.

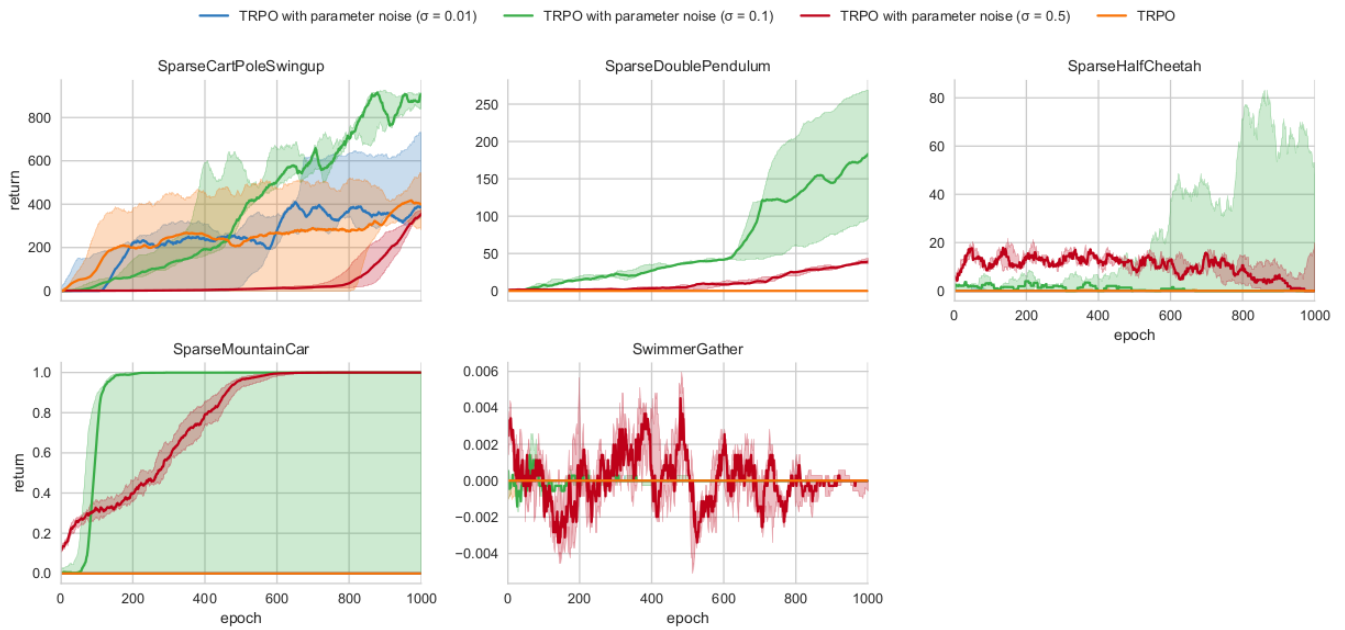


Figure 6: Median TRPO return for environments with sparse rewards plotted over epochs

3.3 Comparison with Evolution Strategies

- DQN with parameter space noise surpass ES on 15 out of 21 Atari games, while use 25 times less data

Table 1: Performance comparison between Evolution Strategies (ES) as reported in [9], DQN with ϵ -greedy, and DQN with parameter space noise (this paper). ES was trained on 1 000 M, while DQN was trained on only 40 M frames.

Game	ES	DQN w/ ϵ -greedy	DQN w/ param noise
Alien	994.0	1535.0	2070.0
Amidar	112.0	281.0	403.5
BankHeist	225.0	510.0	805.0
BeamRider	744.0	8184.0	7884.0
Breakout	9.5	406.0	390.5
Enduro	95.0	1094	1672.5
Freeway	31.0	32.0	31.5
Frostbite	370.0	250.0	1310.0
Gravitar	805.0	300.0	250.0
MontezumaRevenge	0.0	0.0	0.0
Pitfall	0.0	-73.0	-100.0
Pong	21.0	21.0	20.0
PrivateEye	100.0	133.0	100.0
Qbert	147.5	7625.0	7525.0
Seaquest	1390.0	8335.0	8920.0
Solaris	2090.0	720.0	400.0
SpaceInvaders	678.5	1000.0	1205.0
Tutankham	130.3	109.5	181.0
Venture	760.0	0	0
WizardOfWor	3480.0	2350.0	1850.0
Zaxxon	6380.0	8100.0	8050.0

4. Summary

- 和DeepMind同期发的工作很类似, 也是通过给参数加噪音来改善exploration
- DeepMind把他们的工作整合进Rainbow中, 可以看下