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

"Let the Noise Flo"

- Flo Rida

Background – Reinforcement Learning

- Formalize as Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \rho, r)$ with
  - Set of states $\mathcal{S}$
  - Set of actions $\mathcal{A}$
  - Reward function r: $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
  - Initial state distribution ρ: $\mathcal{S} \rightarrow [0, 1]$
  - State transition distribution $\mathcal{P}$: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$
- Agent uses a policy to select actions:

$$\pi\colon \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$$

- We wish to find a policy π that maximizes the expected discounted return:

$$\eta(\pi) := \mathbb{E}_\tau\left[\sum_t \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t)\right], \text{with } \gamma \in [0, 1)$$

- τ denotes a trajectory with $\boldsymbol{s}_0 \sim \rho$, $\boldsymbol{a}_t \sim \pi(\cdot \mid \boldsymbol{s}_t)$, $\boldsymbol{s}_{t+1} \sim \mathcal{P}(\cdot \mid \boldsymbol{s}_t, \boldsymbol{a}_t)$
- Agent has to explore to discover information about r, ρ, $\mathcal{P}$

Parameter Space Noise – Motivation

- Typically, exploration is realized in the action space:
$$\hat{\pi}(\boldsymbol{s}) \coloneqq \pi_{\boldsymbol{\theta}}(\boldsymbol{s}) + \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$$

- However, this leads to inconsistent exploration since the noise is not conditioned on the state

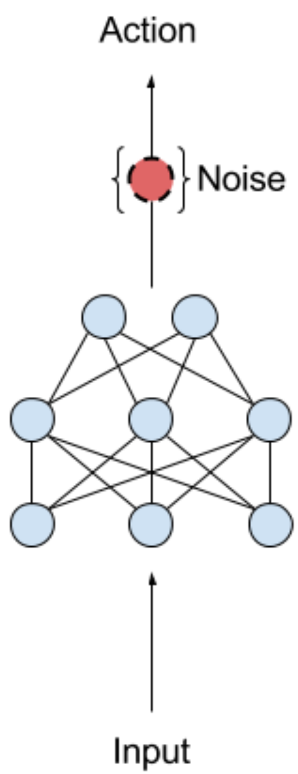
Parameter Space Noise - Formulation

- What if we apply noise to the parameters of the policy instead?

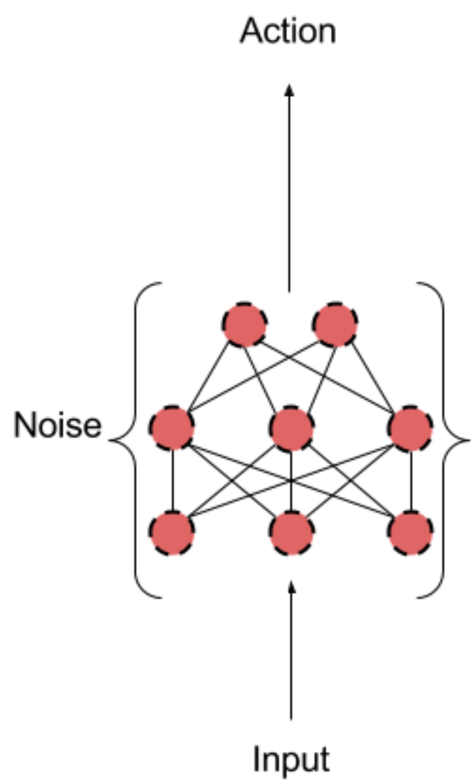
  Define $\hat{\pi}(\boldsymbol{s}) := \pi_{\widehat{\boldsymbol{\theta}}}(\boldsymbol{s})$ with $\widehat{\boldsymbol{\theta}} := \boldsymbol{\theta} + \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$

- We sample the noise at the beginning of each rollout, and keep it fixed for the duration of the rollout.

Parameter Space Noise – Formulation



$$\hat{\pi}(\boldsymbol{s}) \coloneqq \pi_{\boldsymbol{\theta}}(\boldsymbol{s}) + \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$$



$$\hat{\pi}(\boldsymbol{s}) \coloneqq \pi_{\widehat{\boldsymbol{\theta}}}(\boldsymbol{s}) \text{ with } \widehat{\boldsymbol{\theta}} \coloneqq \boldsymbol{\theta} + \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$$

Parameter Space Noise – Problems

- Recall that $\widehat{\boldsymbol{\theta}} \coloneqq \boldsymbol{\theta} + \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$

Parameter Space Noise – Problems

- Recall that $\widehat{\boldsymbol{\theta}} \coloneqq \boldsymbol{\theta} + \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$

- We use a scalar σ to perturb the weights of a deep network (Problem 1)
  - Such a network will likely have many layers
  - Each layer likely has different sensitivities to noise

Parameter Space Noise – Problems

- Recall that $\widehat{\boldsymbol{\theta}} \coloneqq \boldsymbol{\theta} + \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$

- We use a scalar σ to perturb the weights of a deep network (Problem 1)
  - Such a network will likely have many layers
  - Each layer likely has different sensitivities to noise

- We have to pick a suitable scalar σ (Problem 2)
  - In action space noise, the effect is intuitively understandable
  - In contrast, what does perturbing the weights of the policy mean?
  - Furthermore, the sensitivity of the policy to perturbations is likely changing as training progresses

Parameter Space Noise - Problem 1

- Use a similar re-parameterization as proposed in Salimans et al., 2017
- We use layer normalization (Ba et al., 2016)

$$\mathbf{n} = (\frac{\mathbf{a} - \mu}{\sigma})$$

$$\mathbf{h} = \mathbf{f}(\mathbf{g} \odot \mathbf{n} + \mathbf{b})$$

with $\boldsymbol{a} = \boldsymbol{W}\boldsymbol{x}$ and μ and σ are estimated over $\boldsymbol{a}$

- Adding noise to $\boldsymbol{W}$ now perturbs activations $\mathbf{n}$ which are normalized to zero mean and unit variance
- $\mathbf{n}$ more sensitivity to $\mathbf{0}$ mean noise
- Each layer would have similar sensitivity to $\sigma^{\mathbf{2}}$

Parameter Space Noise – Problem 2

- Reasoning about σ in parameter space is hard
- Idea: Think about the effect of a perturbation in action space:

$$d_k := \mathbb{E}_{\boldsymbol{s}}[d(\pi(\cdot \mid \boldsymbol{s}), \hat{\pi}(\cdot \mid \boldsymbol{s}))]$$

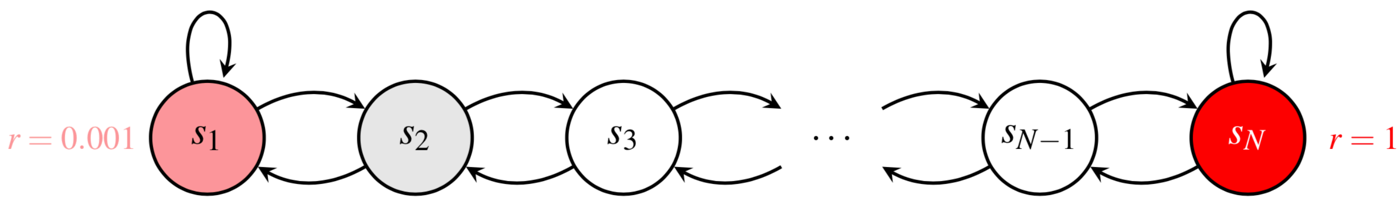
  using some distance / divergence measure $d(\cdot,\cdot)$

- Adaptively change σ:

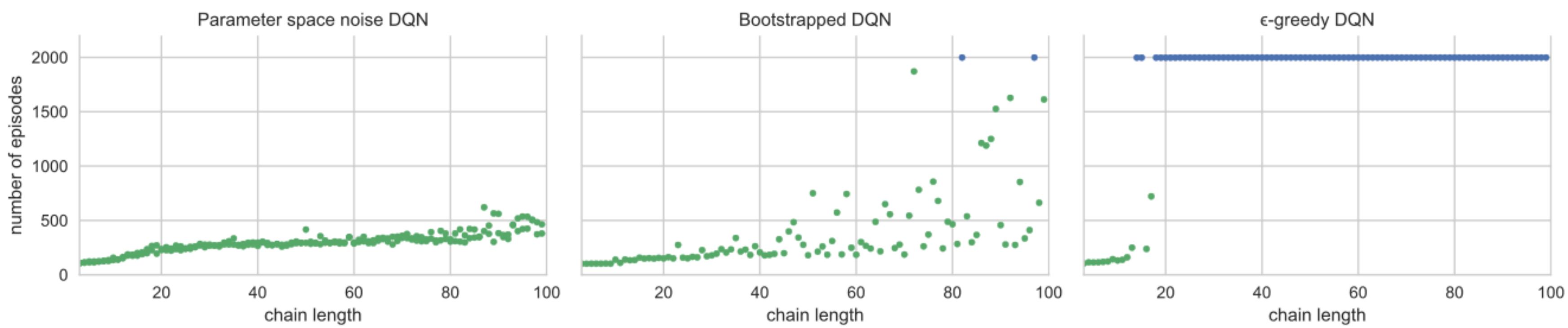
$$\sigma_{k+1} = \begin{cases} \alpha \sigma_k, & d_k \leq \delta \\ \frac{1}{\alpha} \sigma_k, & \text{otherwise} \end{cases}$$

Parameter Space Noise - Experiments (1)

- We test for exploration on a simple but scalable toy environment [1]
  - Chains of length N with initial state s_2. Each episode lasts N + 9 steps, algorithm successful if it can get the optimal reward of 10.



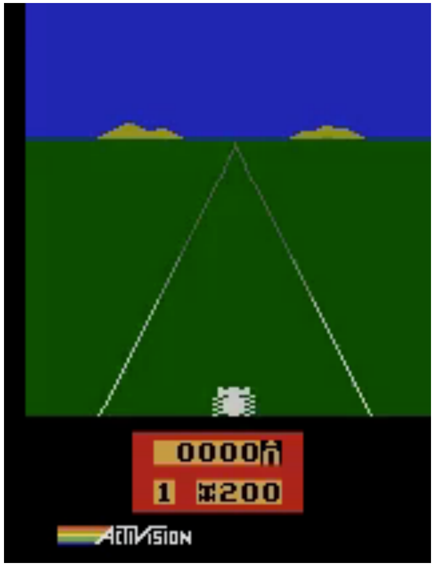
- Experiments on DQN with different exploration methods



[1] "Deep exploration via Bootstrapped DQN", Osband et al., 2016

Parameter Space Noise – Experiments (2)

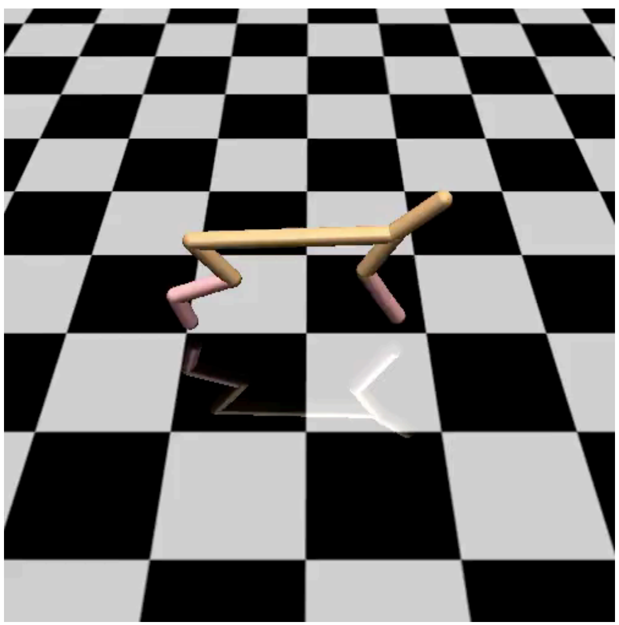
- Evaluation on 20 Atari games
- DQN with different exploration methods
- Exploration behavior of ε-greedy (left) vs. parameter space noise (right)

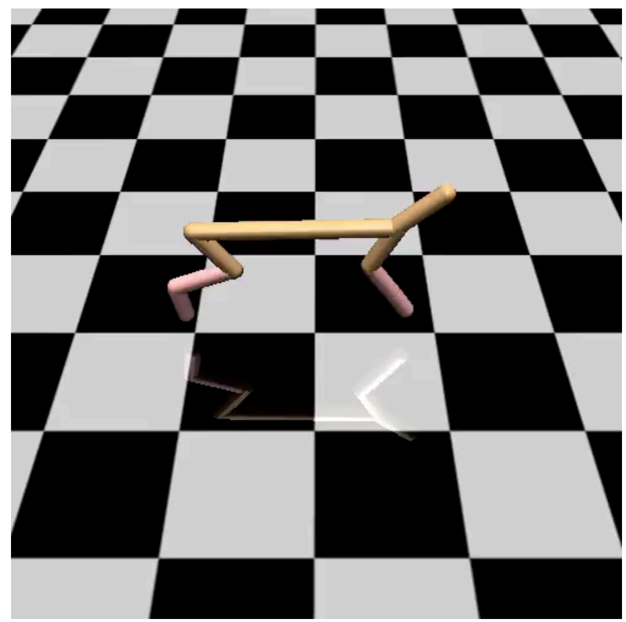


Parameter Space Noise – Experiments (3)

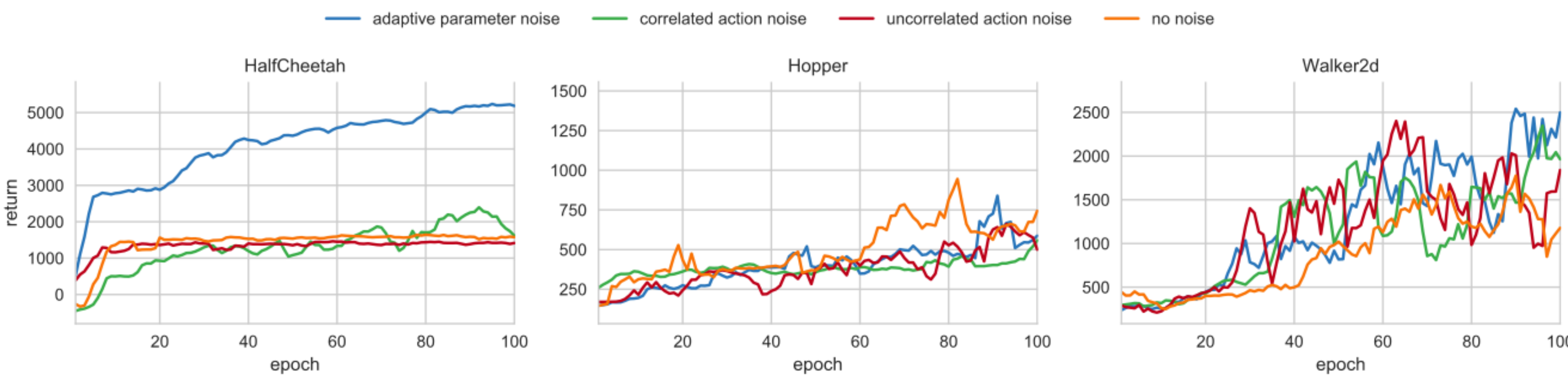
Parameter Space Noise – Experiments (4)

- Evaluation on 7 MuJoCo continuous control problems
- DDPG with different exploration methods
- Exploration of additive Gaussian noise (left) vs. parameter space noise (right)





Parameter Space Noise – Experiments (5)



Parameter Space Noise – Conclusion

- Conceptually simple concept designed as a drop-in replacement for action space noise (or as an addition)
- Often leads to better performance due to better exploration
- Especially helps when exploration is especially important (i.e. sparse rewards)
- Seems to escape local optima (e.g. HalfCheetah)
- Works for off- and on-policy algorithms for discrete and continuous action spaces

Parameter Space Noise – Related Work

- Concurrently to our work, DeepMind has proposed “Noisy Networks for Exploration”, Fortunato et al., 2017
- “Deep Exploration via Bootstrapped DQN”, Osband et al., 2016
- “Evolution strategies as a scalable alternative to reinforcement learning”, Salimans et al., 2017
- “State-dependent exploration for policy gradient methods”, Rückstieß et al., 2008
- And a lot of other papers on the general topic of exploration in RL

Thank you!