

ANALYZING THE EFFECTS OF EMULATING ON THE REINFORCEMENT LEARNING MANIFOLD

Anonymous authors

Paper under double-blind review

1 ARCHITECTURES, ALGORITHMS, METRICS, ENVIRONMENTS AND HYPERPARAMETER DETAILS

1.1 HYPERPARAMETER DETAILS AND ARCHITECTURES

All of the experiments are conducted in Arcade Learning Environment (ALE) Bellemare et al. (2013) with OpenAI wrappers Brockman et al. (2016). The state-of-the-art imitation learning policies are trained with the inverse Q -learning algorithm with the exact hyperparameter details provided in the original paper Garg et al. (2021) for Pong and Breakout MDPs. Thus,

- Replay memory size : 200000
- Initial memory : 5000
- Epsilon step : 1×10^6
- Epsilon window : 10
- Learning steps : 1×10^6
- Critic target update frequency : 1000
- Subsampling frequency : 1
- Batch size : 64
- Demo size : 20
- α : 0.5
- Mix coefficients : 1
- Initial temperature parameter : 1×10^{-3}

for Breakout. For Pong the replay memory size is 100000, initial memory is 5000, epsilon step is 1×10^6 , epsilon window is 10, learning steps is 1×10^6 with evaluation interval 5×10^3 , number of seeds 1000, critic target update is 1000 and the batch size is 64 with same demo size, mix coefficients and α . However, the authors of the inverse Q -learning paper did not share the hyperparameters for Seaquest. We tuned ourselves and we actually achieved slightly higher results than what was reported in the original paper.

For the straightforward vanilla trained deep reinforcement learning policy we use Deep Double Q-Network (DDQN) initially proposed by Hasselt et al. (2016). For completeness we will provide the exact hyperparameters used in here too. For yet more details please see Dhariwal et al. (2017).

- Buffer size : 50000
- Learning rate for Adam optimizer : 5×10^{-5}
- Value for action probability : 0.02
- Discount factor is 0.99
- Batch size is 32

1.2 ADVERSARIAL DIRECTIONS

Not that for consistency with prior study we used the exact same hyperparameters with Korkmaz (2022). In this paper, the authors use the ENR technique to optimize the adversarial directions.

$$\min_{s_{\text{adv}} \in S} c \cdot J(s_{\text{adv}}) + \lambda_1 \|s_{\text{adv}} - s\|_1 + \lambda_2 \|s_{\text{adv}} - s\|_2^2 \quad (1)$$

As has been explained in Section 4 in Definition 4.1 the algorithm and MDP independent adversarial direction $\mathcal{A}_{\text{alg}+\mathcal{M}}^{\text{random}}$ computes a direction from a randomly sampled state of a randomly sampled episode of the policy trained with an algorithm **A** in an MDP \mathcal{M} and introduces this direction to the observation of the policy trained with algorithm **B** in a completely different MDP \mathcal{M}' .

1.3 PEARSON CORRELATION COEFFICIENT

Let $\mathcal{R}_{\mathcal{M}}^i$ represent a sample from the distribution of the cumulative rewards obtained from the MDP by the policy and let $\mathcal{R}_{\text{IQ}}^i$ represent a sample from the distribution of the reward predictions of the inverse Q -learning algorithm. Thus the Pearson correlation coefficient is,

$$\rho_{\mathcal{R}_{\mathcal{M}}, \mathcal{R}_{\text{IQ}}} = \frac{\text{cov}(\mathcal{R}_{\text{IQ}}, \mathcal{R}_{\mathcal{M}})}{\sigma_{\mathcal{R}_{\mathcal{M}}} \sigma_{\mathcal{R}_{\text{IQ}}}} \quad (2)$$

where $\sigma_{\mathcal{R}_{\mathcal{M}}}$ represents the standard deviation of the distribution of $\mathcal{R}_{\mathcal{M}}$ and $\sigma_{\mathcal{R}_{\text{IQ}}}$ represents the standard deviation of the distribution of \mathcal{R}_{IQ} . Note that the covariance of $\mathcal{R}_{\mathcal{M}}$ and \mathcal{R}_{IQ} is

$$\text{cov}(\mathcal{R}_{\text{IQ}}, \mathcal{R}_{\mathcal{M}}) = \mathbb{E}[(\mathcal{R}_{\text{IQ}} - \mu_{\mathcal{R}_{\text{IQ}}})(\mathcal{R}_{\mathcal{M}} - \mu_{\mathcal{R}_{\mathcal{M}}})] \quad (3)$$

where $\mu_{\mathcal{R}_{\text{IQ}}}$ represents the mean of the distribution of \mathcal{R}_{IQ} and $\mu_{\mathcal{R}_{\mathcal{M}}}$ represents the mean of the distribution of $\mathcal{R}_{\mathcal{M}}$.

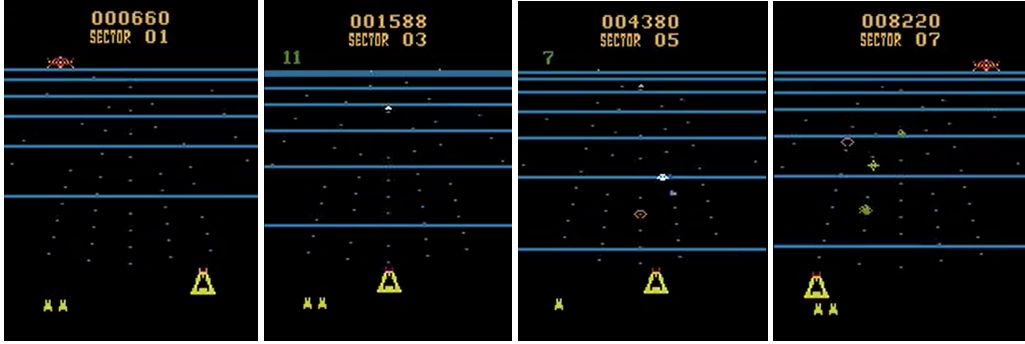


Figure 1: Markov Decision Processes from the Arcade Learning Environment proposed by Bellemare et al. (2013). Rows represents the rollout of the states from an episode in BeamRider.

1.4 SPEARMAN CORRELATION COEFFICIENT

Let $R(\mathcal{R}_{\mathcal{M}})$ represent the rank variables of $\mathcal{R}_{\mathcal{M}}^i$ and $R(\mathcal{R}_{\text{IQ}})$ represent the rank variables of $\mathcal{R}_{\text{IQ}}^i$. Thus, the Spearman correlation coefficient is defined as

$$\rho_{R(\mathcal{R}_{\mathcal{M}}), R(\mathcal{R}_{\text{IQ}})} = \frac{\text{cov}(R(\mathcal{R}_{\text{IQ}}), R(\mathcal{R}_{\mathcal{M}}))}{\sigma_{R(\mathcal{R}_{\mathcal{M}})} \sigma_{R(\mathcal{R}_{\text{IQ}})}} \quad (4)$$

Note that results reported on Pearson correlation coefficient and Spearman correlation coefficient in Section 5.2 in Table 3 are computed as described above.

1.5 ENVIRONMENTS

Note that to provide a fair assessment we used the exact same MDPs with the prior study that introduced the inverse Q -learning algorithm. Figure 1 and Figure 2 show the rollout of states from multiple Markov Decision Process from the Arcade Learning Environment (ALE). All of the MDPs considered have high-dimensional state representations.



Figure 2: Markov Decision Processes from the Arcade Learning Environment proposed by Bellemare et al. (2013). Rows represent the rollout of the states from an episode in Seaquest.

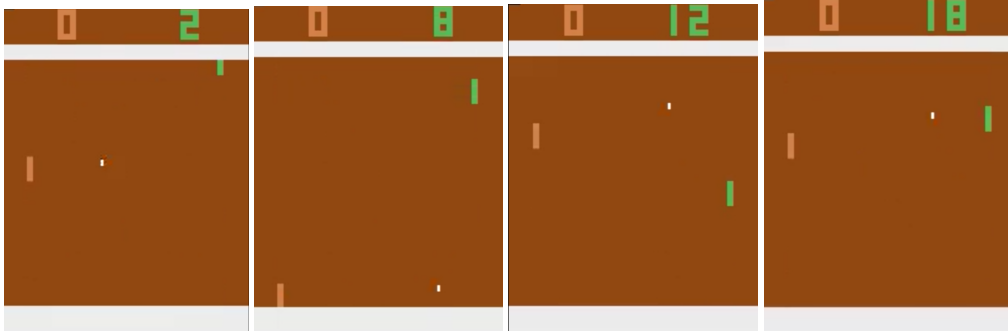


Figure 3: Markov Decision Processes from the Arcade Learning Environment proposed by Bellemare et al. (2013). Rows represent the rollout of the states from an episode in Pong.

REFERENCES

- Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research.*, pp. 253–279, 2013.
- Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv:1606.01540*, 2016.
- Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. *GitHub repository*, 2017.
- Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Neural Information Processing Systems (NeurIPS) [Spotlight Presentation]*, 2021.
- Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *In Thirtieth AAAI conference on artificial intelligence*, 2016.
- Ezgi Korkmaz. Deep reinforcement learning policies learn shared adversarial features across MDPs. *AAAI Conference on Artificial Intelligence*, 2022.