Honey, I Shrunk The Actor: A Case Study on Preserving Performance with Smaller Actors in Actor-Critic RL – Additional Results Data –

Siddharth Mysore
Department of Computer Science
Boston University
Boston, U.S.A.
sidmys@bu.edu

Bassel El Mabsout
Department of Computer Science
Boston University
Boston, U.S.A.
bmabsout@bu.edu

Renato Mancuso
Department of Computer Science
Boston University
Boston, U.S.A.
rmancuso@bu.edu

Kate Saenko
Department of Computer Science, Boston University
Co-affiliated with MIT-IBM Watson AI Lab
Boston, U.S.A.
saenko@bu.edu

Benchmarking Asymmetric Actors

Table I of the main paper (available at http://ai.bu.edu/littleActor/) summarized experimental results of exploring the actor-size reduction benefits of actor-critic asymmetry. In this section, we provide data on the training rewards and losses for each of the algorithms tested, which include DDPG [1], TD3 [2], SAC [3] and PPO [4], on OpenAI Gym [5] benchmarks as well as the Pygame Learning Environment (PLE) [6]. 7 environments were tested: Pendulum-v0, Reacher-v2, Ant-v2, HalfCheetah-v2, Acrobot-v1, PLE's Pong and PLE's Pixelcopter. As with the toy problem, implementations for DDPG, TD3 and SAC are based on the OpenAI Spinning Up [7] code-base. As discussed in the main paper, to preserve parity with published material, our PPO implementation is based on OpenAI's Baselines [8] code as it includes optimizations that are not included in the Spinning Up code-base (we note here that a similar note on the implementation specifics is made in the Spinning Up documentation).

Experiments were run on a desktop computer running Ubuntu 18.04.5 LTS with an Intel Core i7-6850K CPU, Nvidia GeForce GTX 1080 Ti GPU, and 64GB RAM.

Code and corresponding README for running experiments with asymmetric actors and critics on registered OpenAI Gym environments is provided on our project website: http://ai.bu.edu/littleActor/.

A. Training Rewards

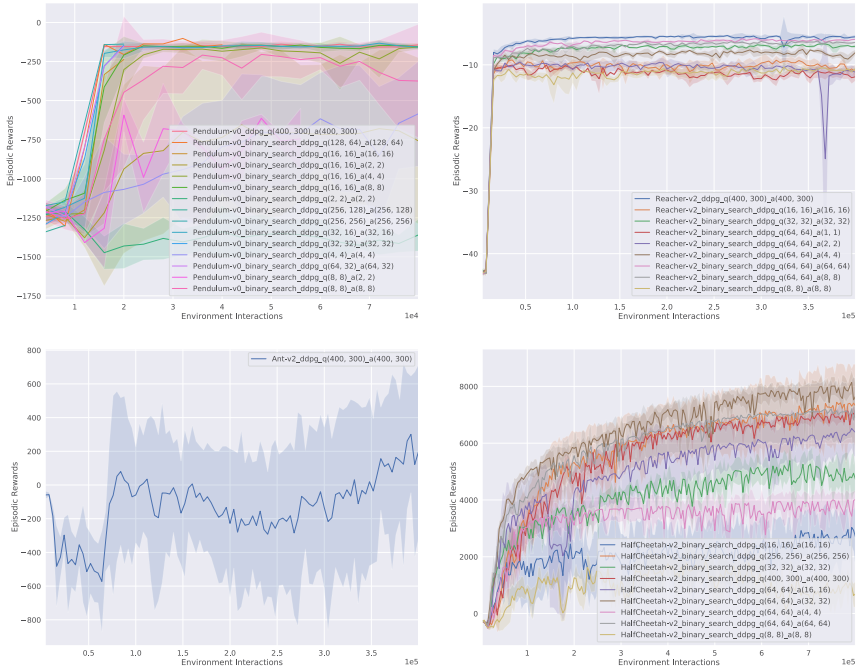

Fig. 1. Progress of training rewards for DDPG on Pendulum-v0, Reacher-v2, Ant-v2 and HalfCheetah-v2. Note DDPG's failure to learn on the Ant-v2 environment

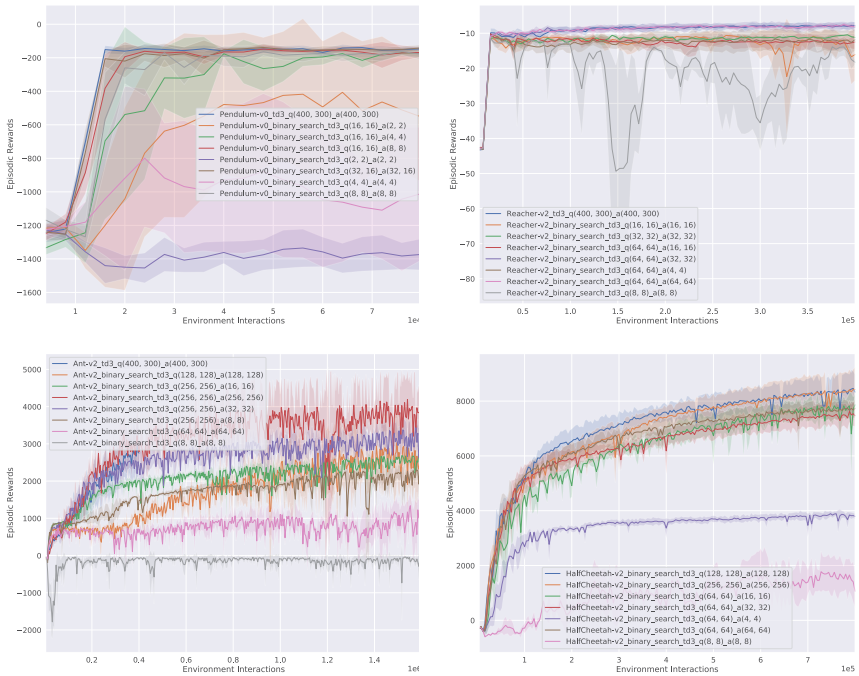

Fig. 2. Progress of training rewards for TD3 on Pendulum-v0, Reacher-v2, Ant-v2 and HalfCheetah-v2

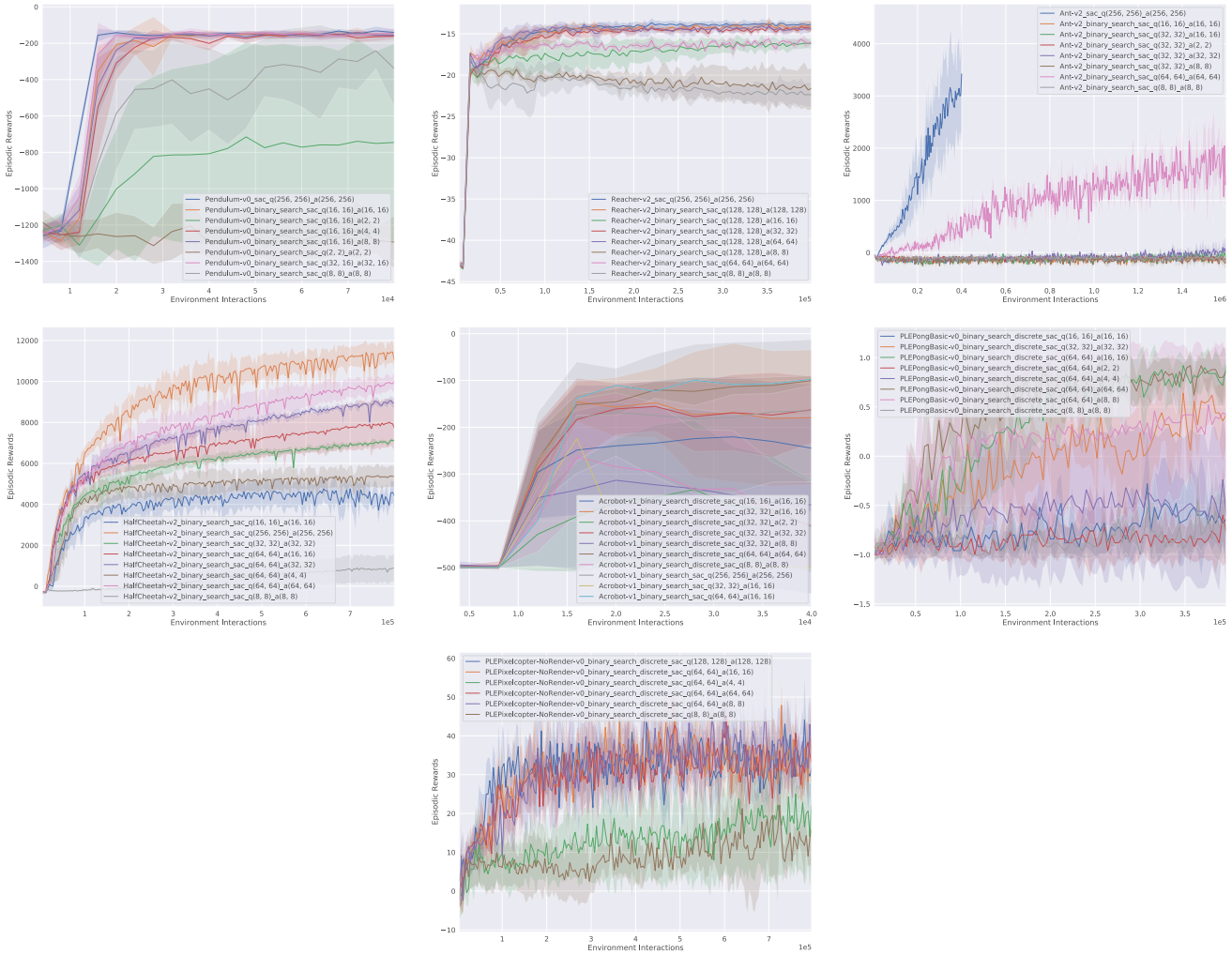

Fig. 3. Progress of training rewards for SAC on Pendulum-v0, Reacher-v2, Ant-v2, HalfCheetah-v2, Acrobot-v1, PLE Pong, and PLE Pixelcopter

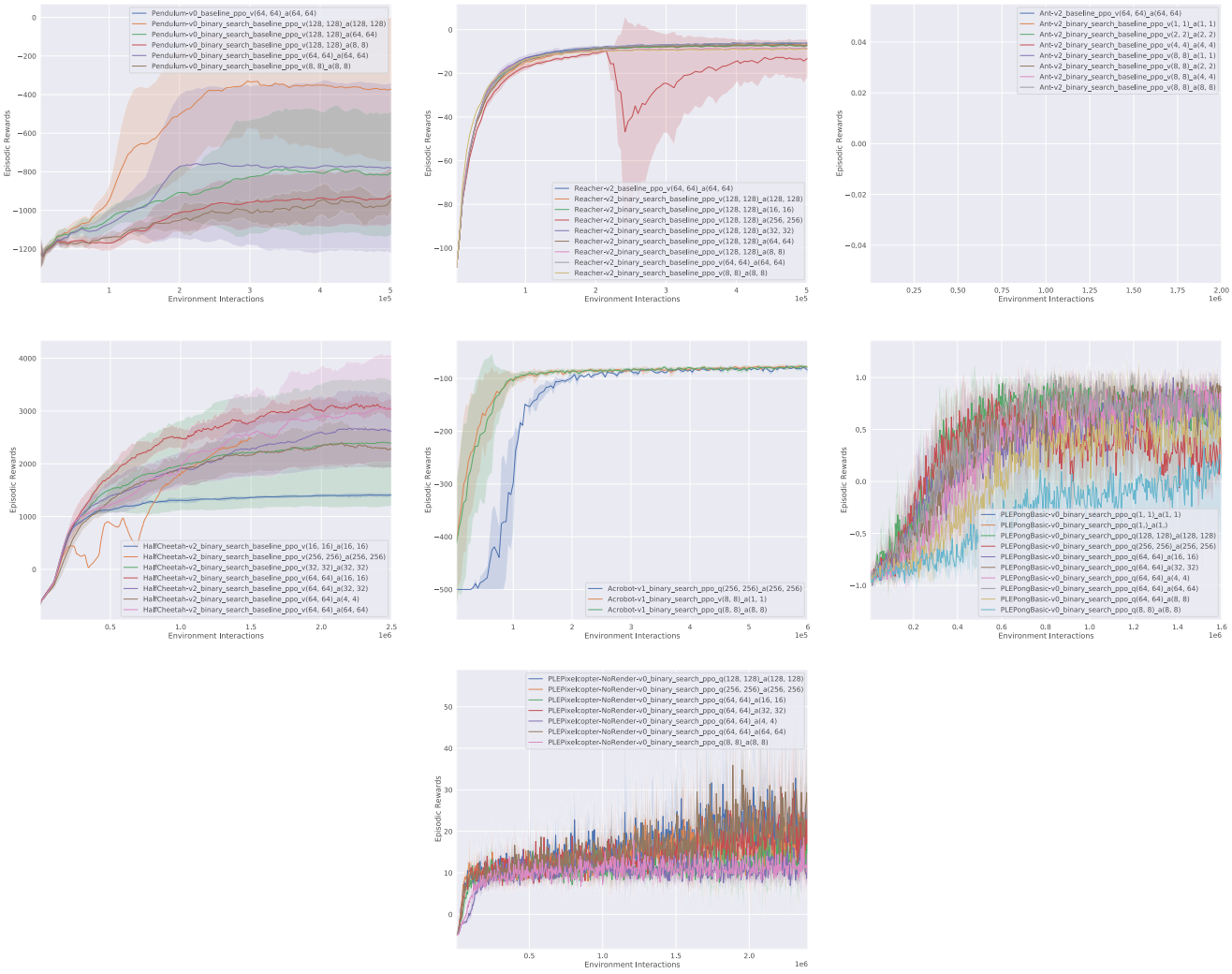

Fig. 4. Progress of training rewards for PPO on Pendulum-v0, Reacher-v2, Ant-v2, HalfCheetah-v2, Acrobot-v1, PLE Pong, and PLE Pixelcopter

B. Training Losses

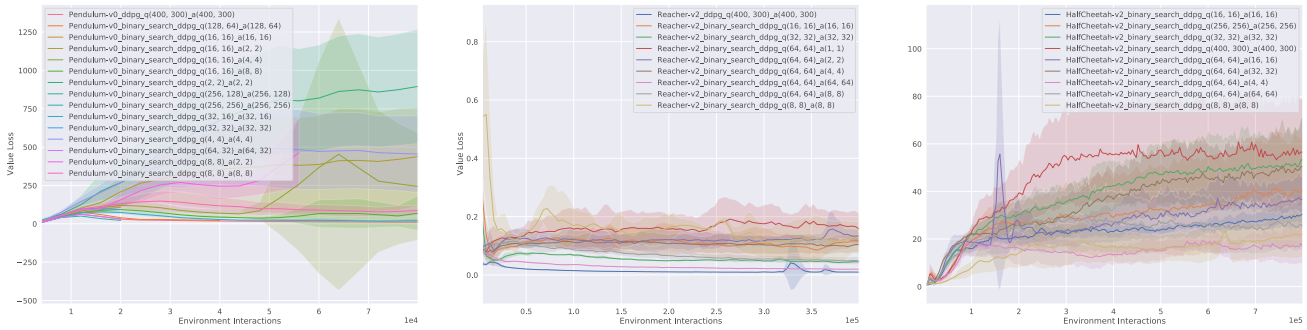

Fig. 5. Progress of value loss for the critic network for DDOG on Pendulum-v0, Reacher-v2 and HalfCheetah-v2. Noet here that Ant-v2 is omitted due to DDPG failing to learn on this environment

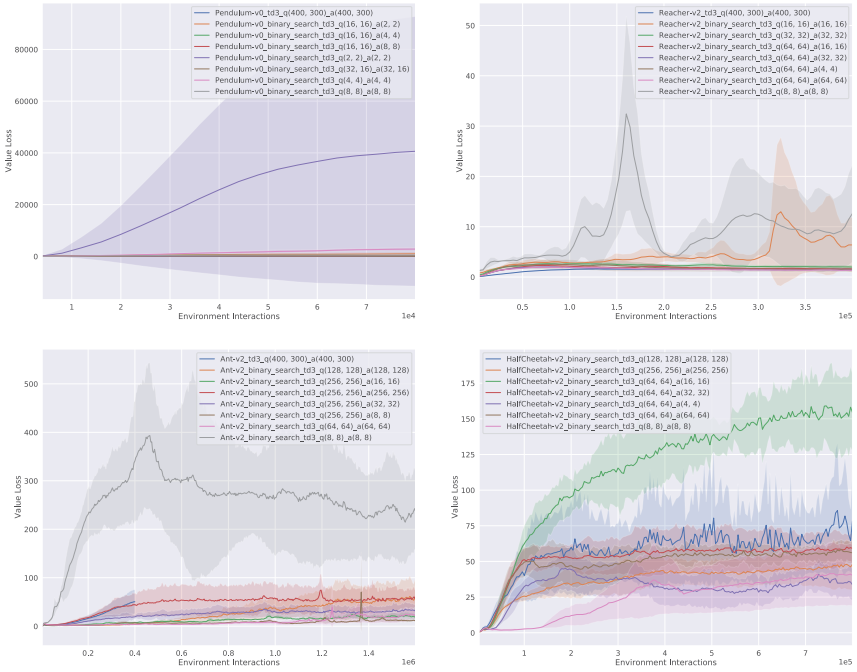

Fig. 6. Progress of average value loss for the critic networks for TD3 on Pendulum-v0, Reacher-v2, Ant-v2 and HalfCheetah-v2

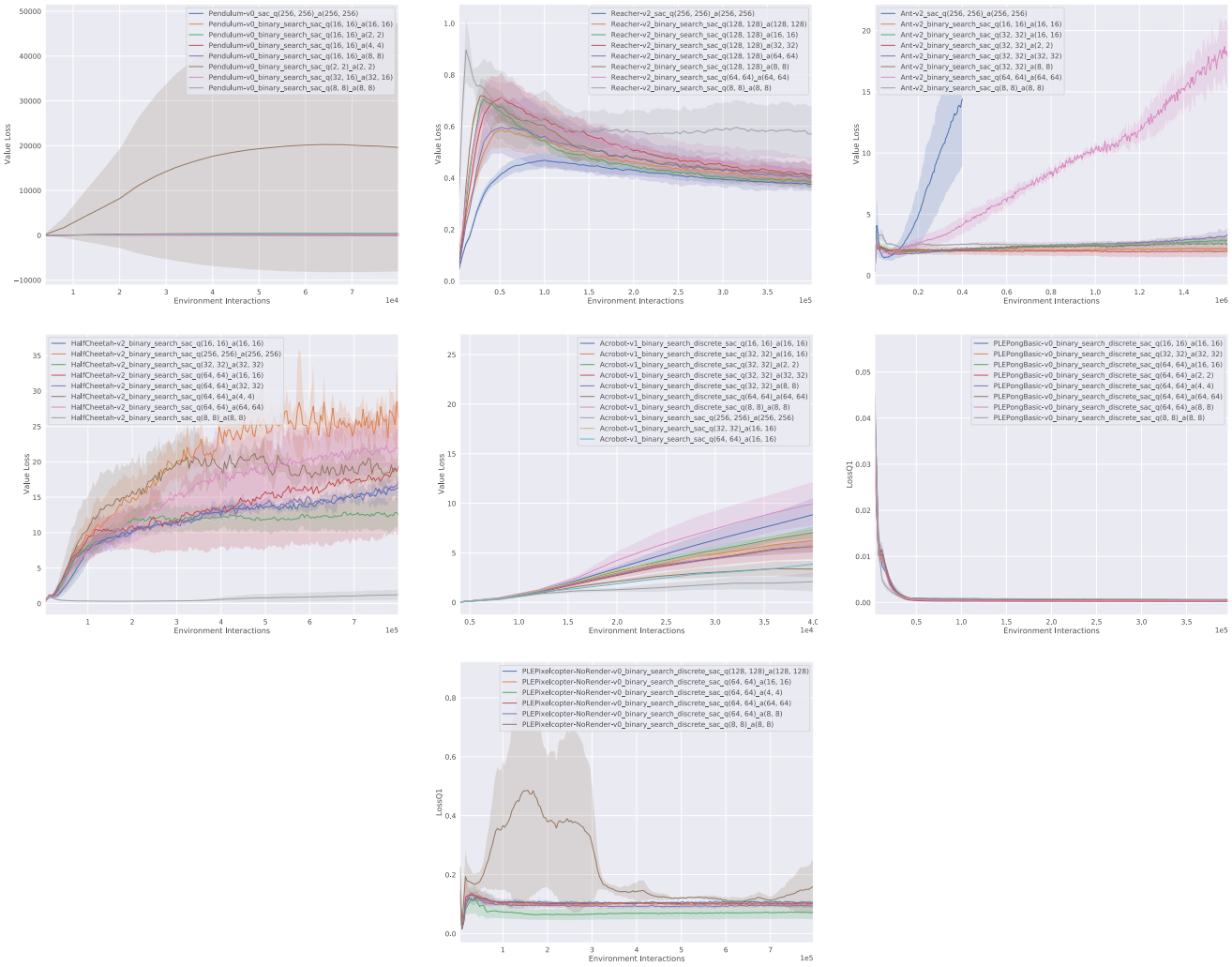

Fig. 7. Progress of average value loss for the critic networks for SAC on Pendulum-v0, Reacher-v2, Ant-v2, HalfCheetah-v2, Acrobot-v1, PLE Pong, and PLE Pixelcopter

REFERENCES

[1] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *International Conference on Learning Representations*, 2016.

[2] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*, 2018, pp. 1587–1596.

[3] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *International Conference on Machine Learning (ICML)*, 2018.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.

[5] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *CoRR*, vol. abs/1606.01540, 2016.

[6] N. Tasfi, "Pygame learning environment," https://github.com/ntasfi/PyGame-Learning-Environment, 2016.

[7] J. Achiam, "Spinning Up in Deep Reinforcement Learning," 2018.

[8] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov, "Openai baselines," https://github.com/openai/baselines, 2017.