The University of Texas at Austin
Cockrell School of Engineering

SEP 2022

CONTINUOUS CONTROL WITH DEEP REINFORCEMENT LEARNING

Lillicrap, Hunt, Pritzel, Heess, Erez, Tassa, Silver, Wierstra (all from Deepmind)
First published Sep 2015

JEFF BONYUN
ME Masters Student, The University of Texas at Austin

SET THE STAGE

Imagine…

It’s 2015.

Obama is President.

You’re minding your own business in middle school… when…

Deep-Q happens!

- DeepMind’s Atari paper is published in Nature.
- Played Atari games using pixel inputs and joystick actions.
- Was awesome.

BUT DEEP-Q ISN'T PERFECT

- Sure, it has a huge input space, from raw pixels.
- Sure, it's very capable at the task.

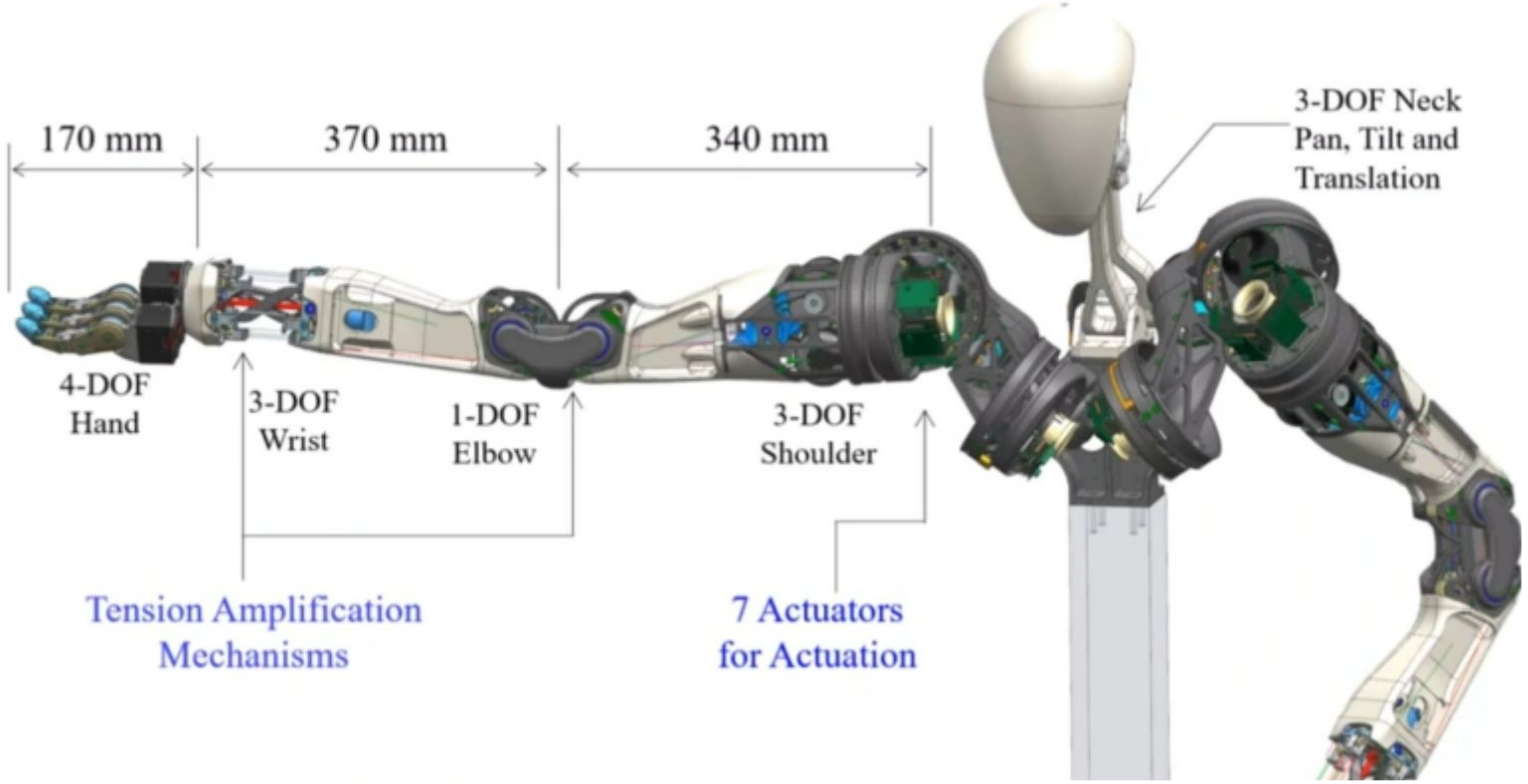
- Its actions were limited to an old-school Atari joystick:
  - 8 directions or no direction
  - with or without the button pressed
  - = 18 actions.

WHAT DDPG ACCOMPLISHES

- Starts with Deep Q's awesomeness
- Add High-Dimensional Actions
- Add Continuous Actions
- Can't we just divide into discrete steps?
  - You could, but if the space is *also* high-dimensional, it doesn't do any good.
  - Say you have 10 dimensions, and you discretize into 10 steps each. Now you have 10^10 = 10 billion actions = too many.

WHY SHOULD WE CARE?

Google: “robot with lots of joints”



- Robots tend to have lots of dimensions.
  - Every joint is another dimension.
- Robot dimensions tend to be continuous.
  - Every joint angle or joint velocity or joint torque is a continuous variable.
- We want to do “Robot Learning”, so we’d like to be able to learn actions that are relevant.

BUILDING BLOCKS

BUILDING BLOCK: Q LEARNING

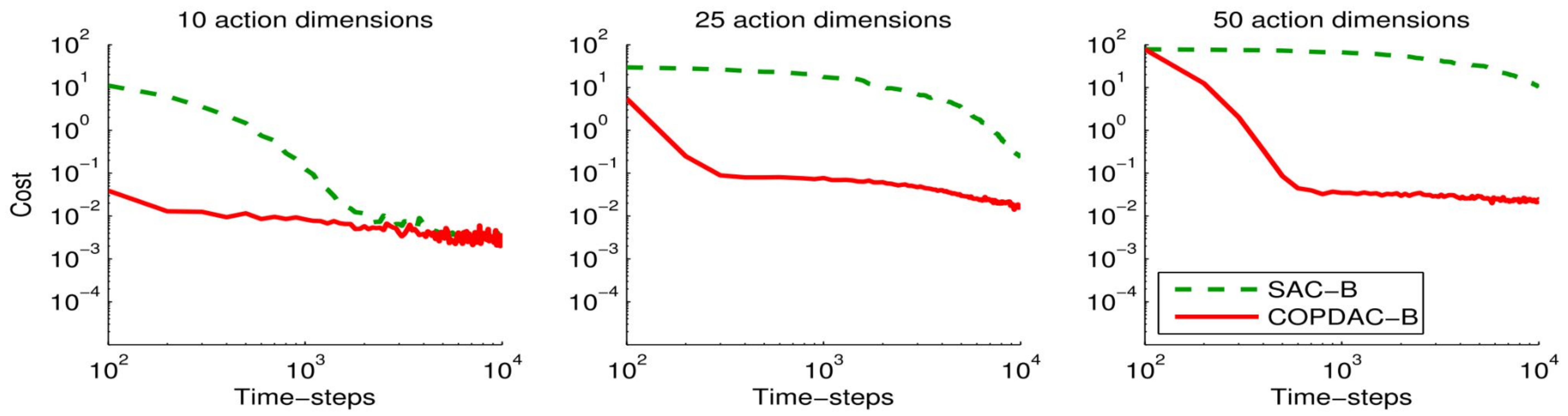
- 1989 (!): dissertation (!) by Watkins.
- Learns Q function = action-value function = value of taking an action from a given state and thereafter following a policy.
- Builds Q function through dynamic programming, recursively referring to itself in a different state.
- Mathematical proof of convergence.
- Discrete state; discrete actions.
    - Must visit every state and try every action an infinite number of times, if you want convergence.

BUILDING BLOCK: OFF-POLICY ACTOR-CRITIC

- 2012: Degris, White, Sutton (U of Alberta!)
- Combined Off-Policy with Actor-Critic for the first time.
- Off-Policy:
  - Learning the Q function while taking actions that aren't consistent with the policy you are learning.
  - Useful for: exploration and sample efficiency.
  - Previous works required argmax over actions; bad for continuous.
- Actor-Critic:
  - Actor: model for the policy that tells you what action to take.
  - Critic: model for the Q function.
  - Useful for: large/continuous action spaces, because the Actor takes care of finding the "best" action, instead of iterating through them all.

BUILDING BLOCK: DETERMINISTIC POLICY GRADIENT

- 2014: Silver et al (Deepmind)
- Uses Off-Policy Actor Critic
- Stochastic policy gradients require integration over state and actions.
  - You have to account for all the actions that you might take in that state.
- Deterministic policy gradients only require integration over state.
  - You know exactly which action you will take, so only consider that one.
- Much faster learning in large action spaces.



BUILDING BLOCK: DEEP-Q

- 2015 Atari paper: Minh et al (Deepmind again)
- Monochrome (preprocessed) images to joystick actions
- Models the Q function with a deep CNN, which was previously thought unstable
- Avoids instability of nonlinear functions for Q learning by:
  - Experience Replay: learning on random minibatches from a large buffer of past experiences, which breaks correlation between experiences.
  - Periodic target Q updates: keeps the Q target function stable as the new Q function is being learned, instead of fluctuating too quickly.
- Large (but discrete) state; discrete and small actions.

BUILDING BLOCK: DEEP-Q

- Evaluates every possible action at each step.
  - i.e. it is not using actor-critic, just “critic” in the form of the deep CNN Q function.

- Do you have lots of actions?
  - You’d have to iterate through each of them, calculating your Q value, to decide.
- Do you have continuous actions?
  - You’d have to run a non-convex optimization at every step.
- Do you have high-dimensional continuous actions?
  - Your non-convex optimization keeps getting harder (→ impossible) to solve.

DEEP DETERMINISTIC POLICY GRADIENT

TAXONOMY

Modeling	Model-Free
On/Off Policy	Off-policy
Input Space	Large; Fully Observed
Action Space	Large and continuous
Environment Response	Stochastic
Reward Response	Deterministic or Expected
Policy Response	Deterministic

CLEVER IDEA

- Add Deep-Q's clever ideas to DPG:
  - Deep Non-linear Critic
  - Experience Replay for stability (removing correlation)
  - Stable Target Network for stability (removing moving target)

- Add DPG's clever ideas to Deep-Q:
  - Actor Network for large continuous action space
  - Deterministic Policy for efficient learning

- Throw in some new stuff
  - Batch Normalization to work across domains (new in 2015, but not invented here).
  - Exponentially Updating Target instead of periodic copy (seems minor, but more graceful).
  - Exploration Noise so that behavior policy will cover state/action space.

CLEVER IDEA?

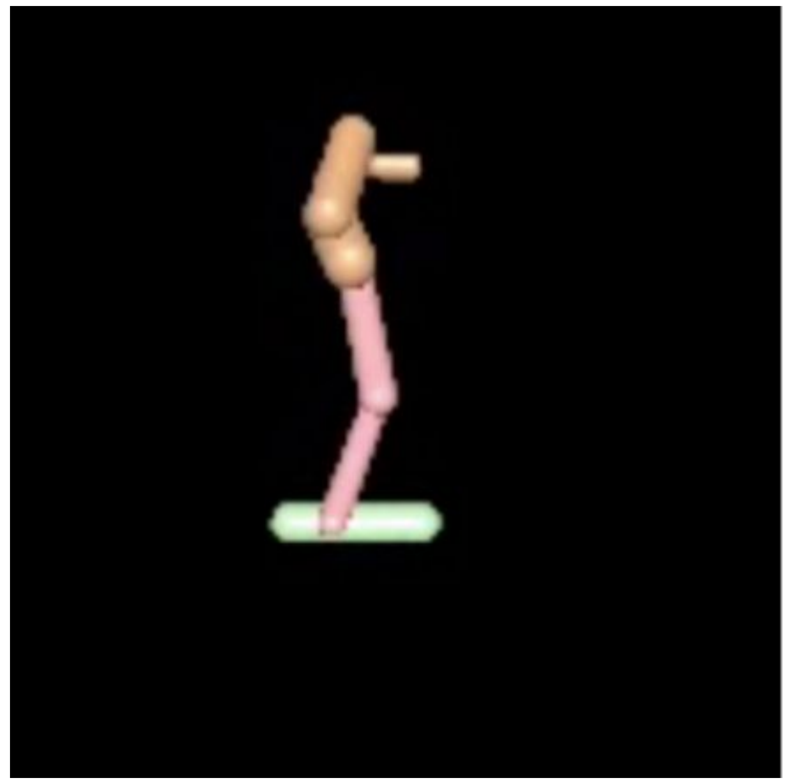
- So how clever is this paper?
  - Meh. Seems pretty obvious next step, but someone had to publish it.
  - The novel additions don’t seem that big.
  - I speculate that this was well underway before Deep Q was published.
  - Remember that all these ideas are from Deepmind.

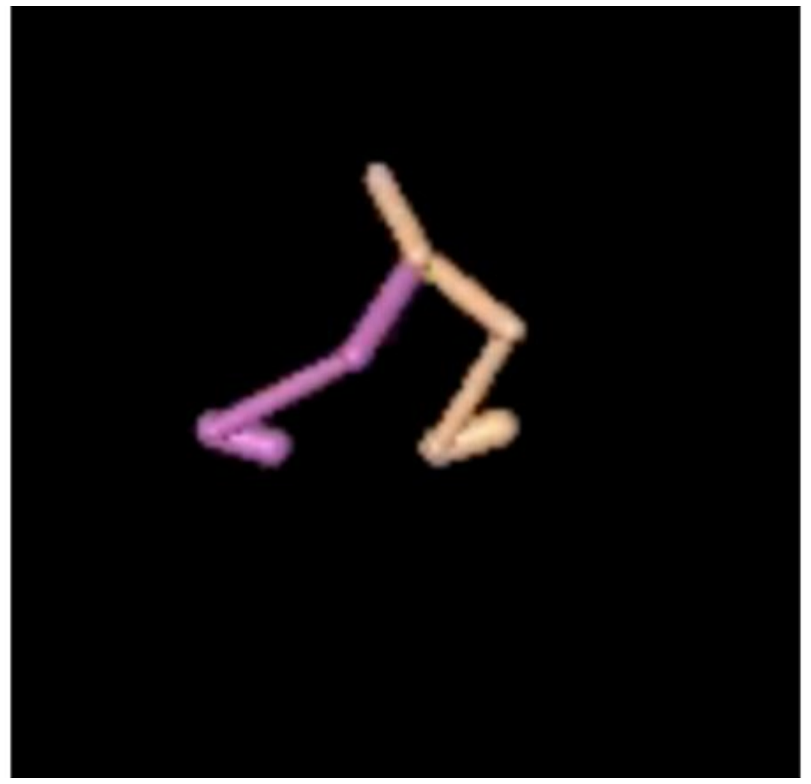
EXPERIMENT

TASKS

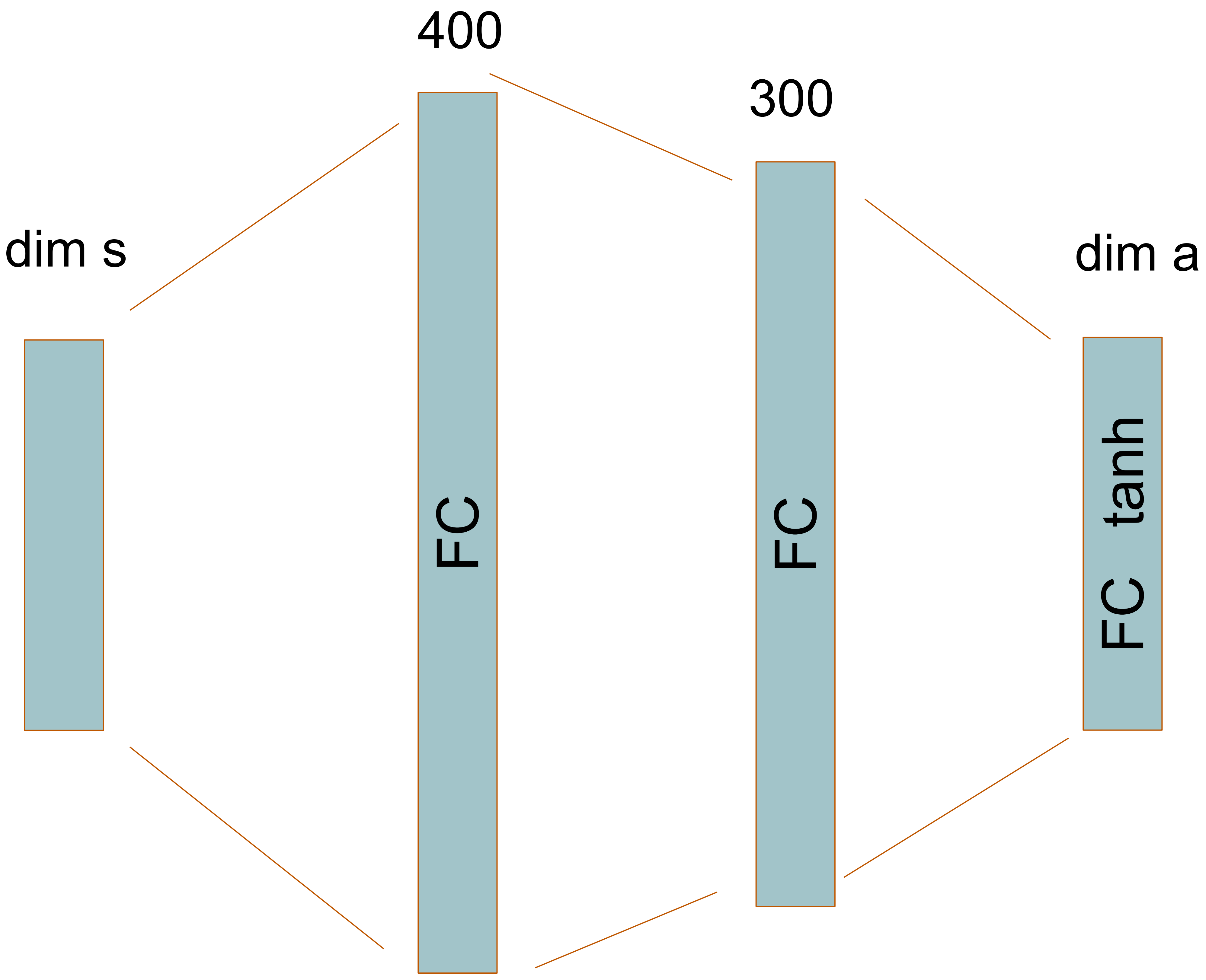
- Ran on 26 physics simulation problems in Mujoco (plus Torcs)
  - Chose these tasks because they are inherently continuous, and Deep Q could not have solved them.
  - Listed on right →
- Actions were the torques on the joints
  - Continuous
  - Varied from 1 through 12 action dimensions

environment
blockworld1
blockworld3da
canada
canada2d
cart
cartpole
cartpoleBalance
cartpoleParallelDouble
cartpoleSerialDouble
cartpoleSerialTriple
cheetah
fixedReacher
fixedReacherDouble
fixedReacherSingle
gripper
gripperRandom
hardCheetah
hopper
hyq
movingGripper
pendulum
reacher
reacher3daFixedTarget
reacher3daRandomTarget
reacherSingle
walker2d
torcs

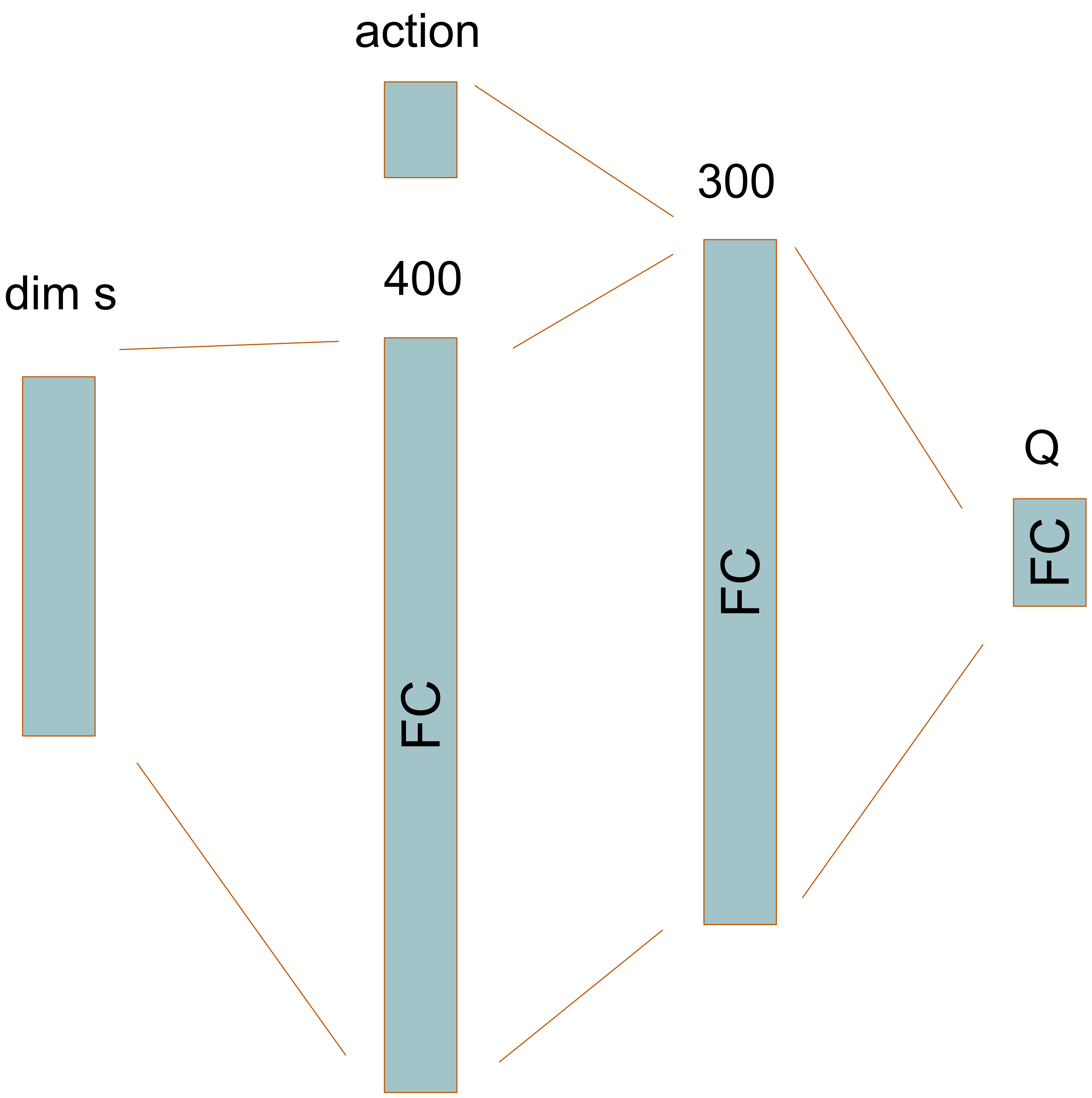




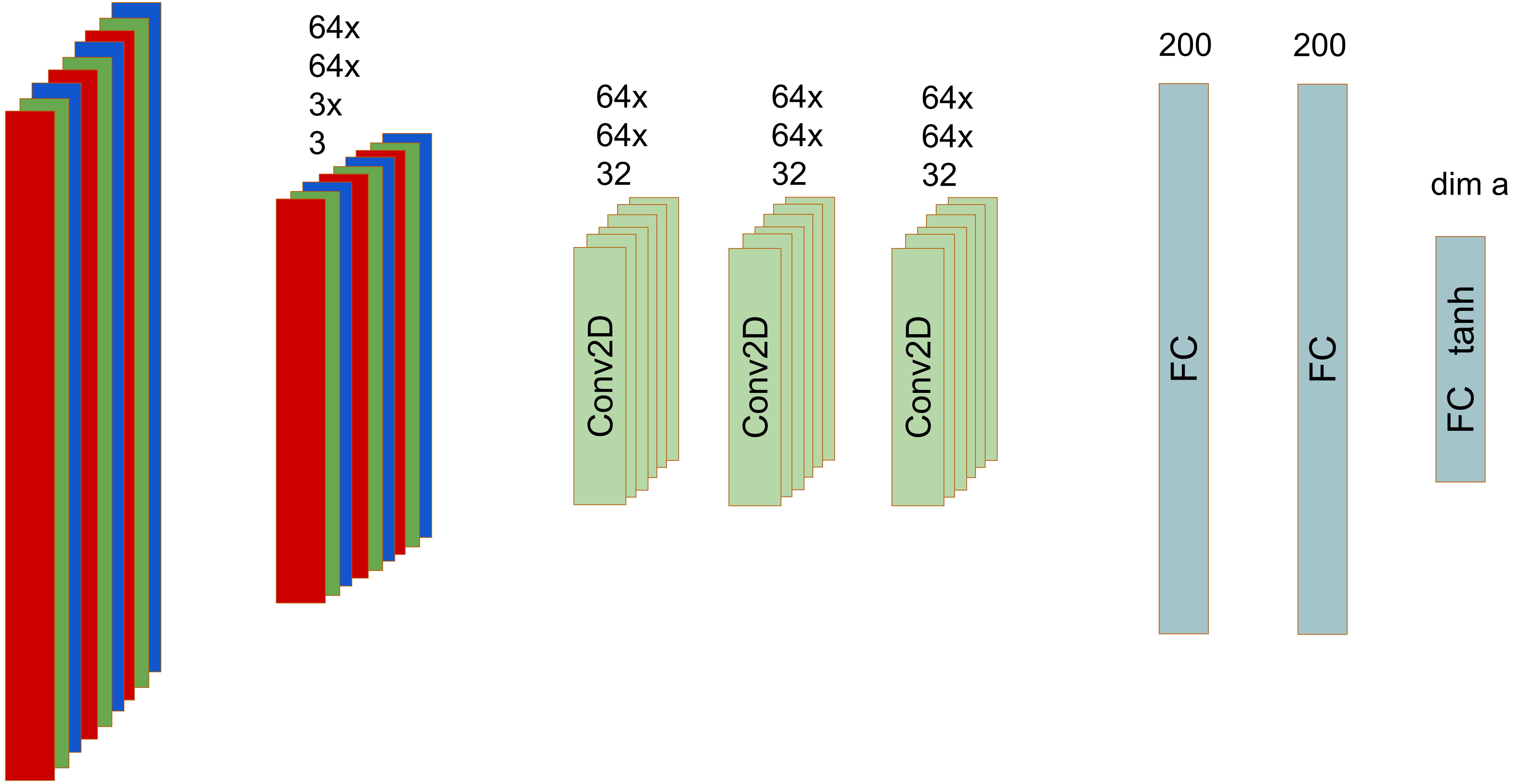
NETWORK: LOW-D STATE ACTOR



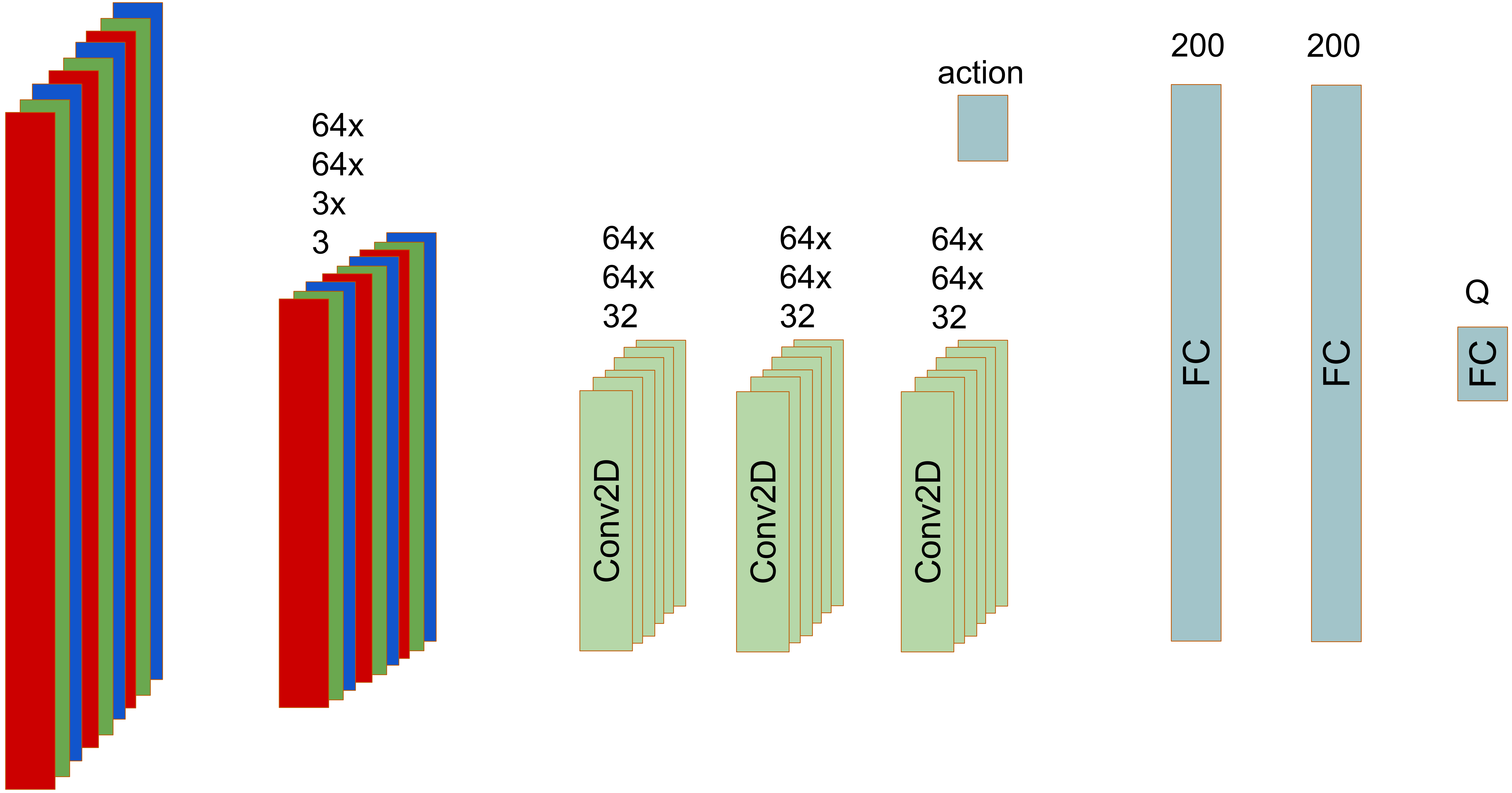
NETWORK: LOW-D STATE CRITIC / Q



NETWORK: PIXEL ACTOR



NETWORK: PIXEL CRITIC / Q



TRAINING

- Pretty much everything was done according to DPG or Deep-Q
  - Actor updates

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s_t \sim \rho^\beta} \left[\nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta_\mu} \mu(s|\theta^\mu)|_{s=s_t} \right]$$

  - Critic Updates
    - Deterministic Bellman (no E over actions) $Q^\mu(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E} \left[r(s_t, a_t) + \gamma Q^\mu(s_{t+1}, \mu(s_{t+1})) \right]$
    - Loss for critic: $L(\theta^Q) = \mathbb{E}_{s_t \sim \rho^\beta, a_t \sim \beta, r_t \sim E} \left[\left(Q(s_t, a_t|\theta^Q) - \quad r(s_t, a_t) + \gamma Q(s_{t+1}, \mu(s_{t+1})|\theta^Q) \right)^2 \right]$
  - Replay
    - Saved 1,000,000 steps, 16 used in minibatch (64 for low-D)

BATCH NORM

- Had just been published by Ioffe & Szegedy (2015)
- For a given minibatch, whiten the activations (mean zero, variance one).
- Determines an average normalization, to use during testing (when there is no minibatch)
- Why?
  - “Minimize covariance shift”, where inputs change over time.
  - But mostly it seems to adjust the scale of inputs from different tasks, so that the same hyperparameters will work on them all.

TARGET NETWORK UPDATES

- Deep-Q copied the learned Q function to the target Q function every "C" steps
- DDPG finds it must have target networks for actor **and** critic to achieve same stability.
- Slight variation to evolve slowly

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta' \text{ with } \tau \ll 1$$

- Use $\tau = 0.001$, which is a 680-step halflife

  i.e. it takes a while for the target networks to reflect new learning

EXPLORATION NOISE

- Adds noise to the policy:

$$\mu'(s_t) = \mu(s_t|\theta_t^{\mu}) + \mathcal{N}$$

(this terminology seems to overlap with the target policy network)

- Encourages exploration

- Used momentum in their noise (Ornstien-Uhlenbeck process) to make meaningful deviations.

EVALUATION

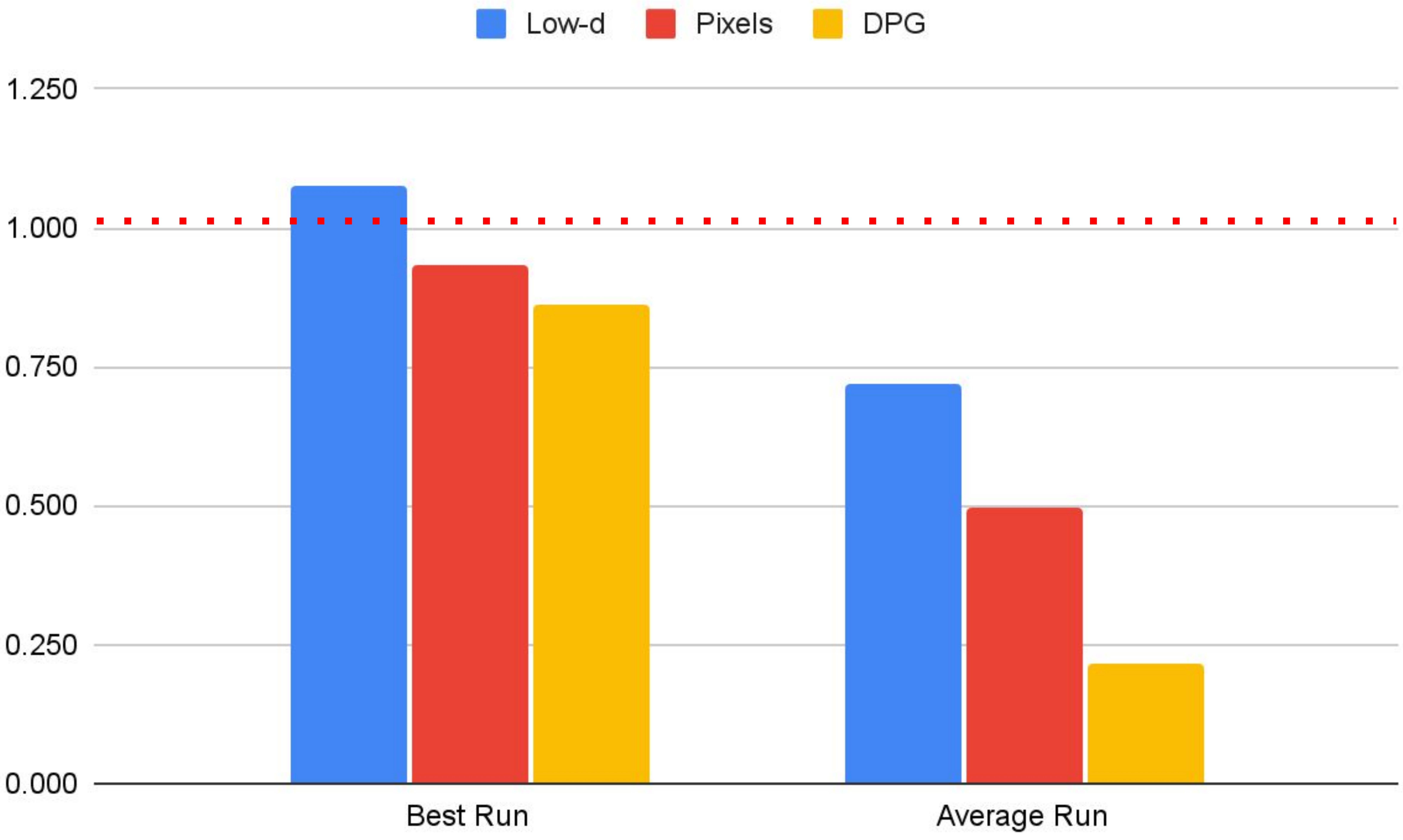
COMPARED TO

- Random agent
  - What it sounds like.
  - Sets the "0" mark for their performance scale.

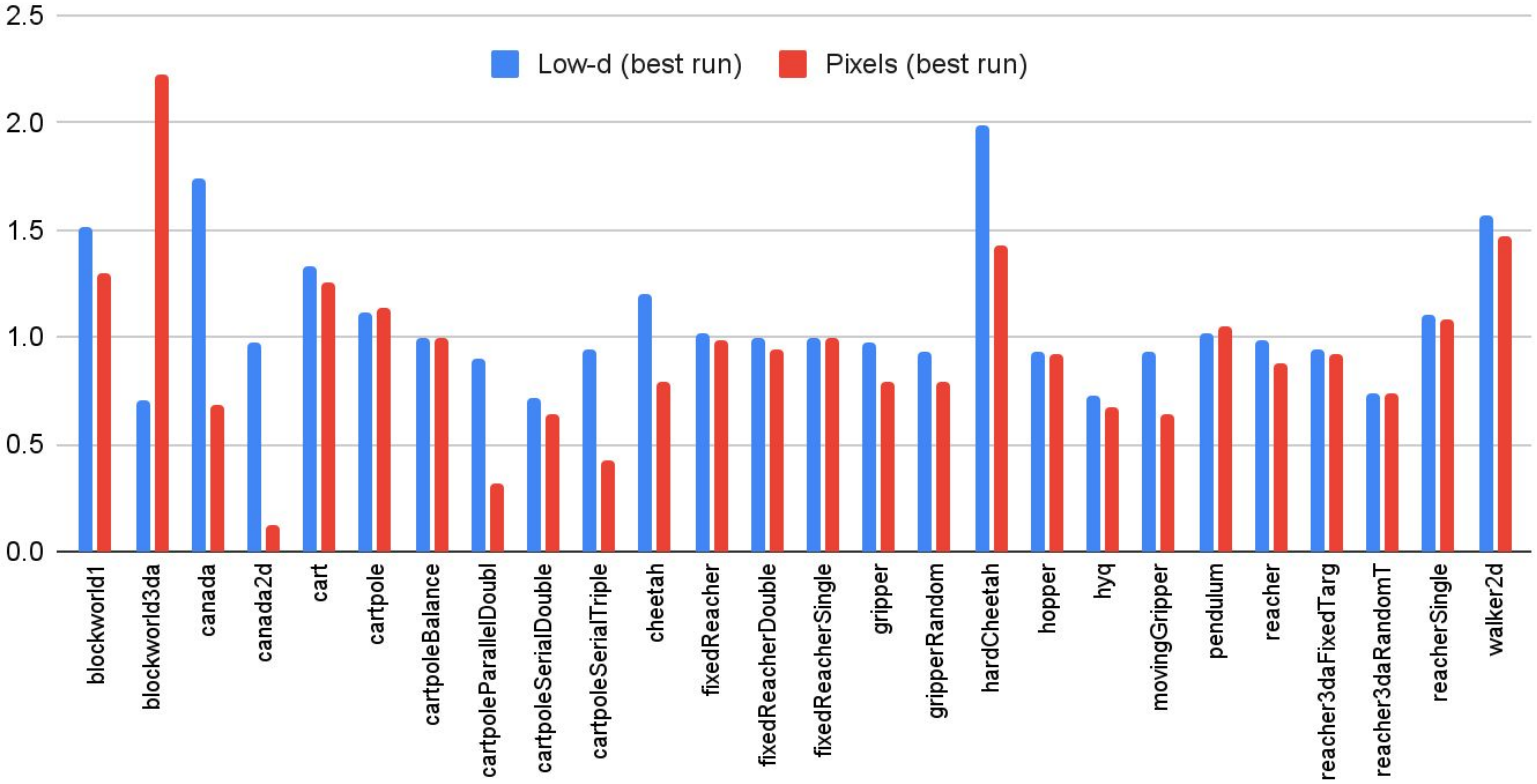
- iLGQ model-predictive controller
  - Simulates the future of the physics out 0.25 to 0.60 seconds, and optimizes the action on the simulated future.
  - Sets the "1" mark for their performance scale.

- Basic DPG
- Ablated "ours"

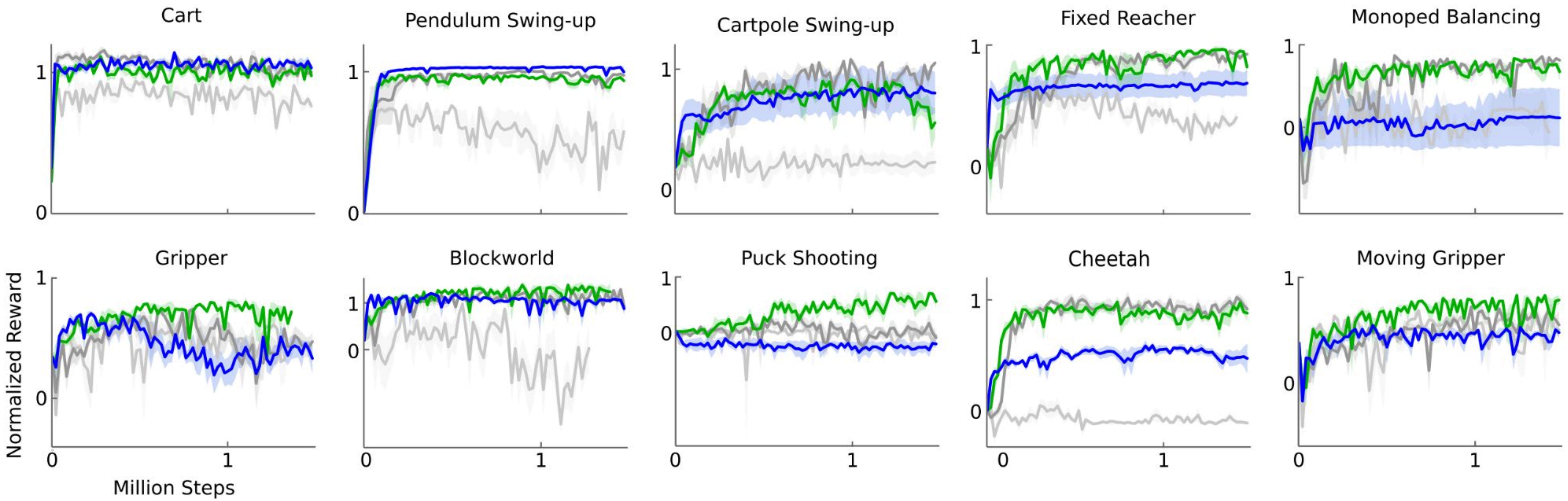
AVERAGE TESTING RESULTS



DETAILED TESTING RESULTS



ABLATION RESULTS



- Green is low-d, blue is from pixels, dark grey is no batch-norm, light gray is no target network
- They conclude that the target network is the important part

LIMITATIONS

- Sample efficiency
  - The authors say this is still a problem for all model-free methods.
  - But DDPG is an improvement over on-policy methods.

- Deterministic policies
  - Multiplayer games often require stochastic policies to play optimally (poker bluffing, tennis serve placement, etc)
  - Claim that the “reparamaterization trick” can be used to apply to stochastic policies.

CRITIQUES

- Duan et al (2016):
  - “converge significantly faster”
  - “less stable than batch algorithms” (eg TRPO)
- Haarnoja et al (2017):
  - “as dynamics become more unstable (e.g. in Hopper-v1) performance gains rapidly diminish” due to exploration noise
  - Found more stability in other algorithms, though DDPG was fastest in many tasks

- Generally, stability is still cited as the real problem.

WHAT COMES NEXT

- Various attempts to make it more stable
  - e.g. Haarnoja et al (2017):
    - Scaling rewards helps stability considerably

- TD3: Twin Delay Deep Deterministic Policy Gradient, Fujimoto et al (2018)
  - Min of two value functions to reduce overestimation (“twin”)
  - No policy updates until values are partially learned (“delay”)

- Soft Actor Critic
  - Adds entropy and ends up with more stability.

VIDEO OF AGENTS: LOW-D CHEETAH

VIDEO OF AGENTS: PIXEL 7-DOF REACHING

SUMMARY OF DDPG

- Deep-Q but with high-dimensional, continuous actions
- Enabled new tasks, trained fast, but lacked stability

SOURCES

- Main paper
  - *Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.*
- Building Blocks
  - Watkins, C. J. C. H. (1989). Learning from delayed rewards.
  - Degris, T., White, M., & Sutton, R. S. (2012). Off-policy actor-critic. arXiv preprint arXiv:1205.4839.
  - Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014, January). Deterministic policy gradient algorithms. In *International conference on machine learning* (pp. 387-395). PMLR.
  - Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540), 529-533.
- Critiques
  - Duan, Y., Chen, X., Houthooft, R., Schulman, J. & Abbeel, P.. (2016). Benchmarking Deep Reinforcement Learning for Continuous Control. *Proceedings of The 33rd International Conference on Machine Learning*, in *Proceedings of Machine Learning Research* 48:1329-1338 Available from https://proceedings.mlr.press/v48/duan16.html.
  - Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S.. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. <i>Proceedings of the 35th International Conference on Machine Learning</i>, in <i>Proceedings of Machine Learning Research</i> 80:1861-1870 Available from https://proceedings.mlr.press/v80/haarnoja18b.html.
- What Comes Next
  - Fujimoto, S., Hoof, H. & Meger, D.. (2018). Addressing Function Approximation Error in Actor-Critic Methods. *Proceedings of the 35th International Conference on Machine Learning*, in *Proceedings of Machine Learning Research* 80:1587-1596 Available from https://proceedings.mlr.press/v80/fujimoto18a.html.
  - Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.
- Robot Arm Image
  - H. Song, Y. -S. Kim, J. Yoon, S. -H. Yun, J. Seo and Y. -J. Kim, "Development of Low-Inertia High-Stiffness Manipulator LIMS2 for High-Speed Manipulation of Foldable Objects," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4145-4151, doi: 10.1109/IROS.2018.8594005.

VIDEO OF AGENTS (WHOLE THING)