Max Planck Institute for Intelligent Systems – Empirical Inference Group

Investigating the Impact of Action Representations in Policy Gradient Algorithms

Jan Schneider, Pierre Schumacher, Daniel Häufle, Bernhard Schölkopf, Dieter Büchler

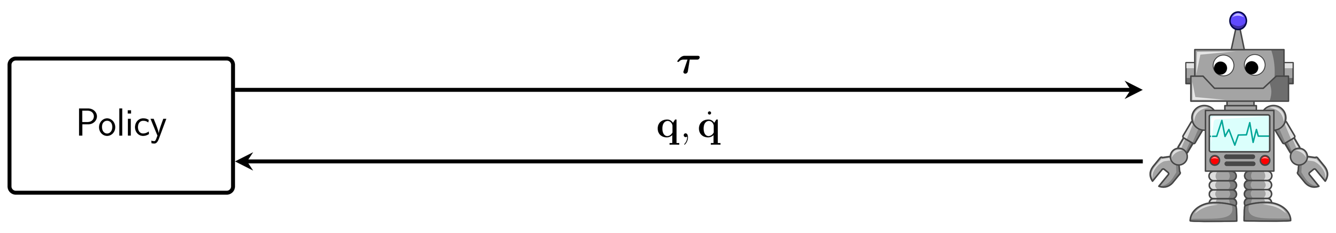
Overview

- In RL tasks, there are typically many choices for the action representation
  → Robotics: torques, joint positions/velocities, activations of artificial muscles, ...
- The choice of action representation has a significant impact on the performance of reinforcement learning (RL) algorithms
- The reasons for these performance differences are generally not clear
  → We apply two analysis techniques to investigate the influence of the action representation on the learning process
- Finally, we outline open challenges that need to be addressed to gain further insights into the causes of the performance differences

Action representations

Torque control

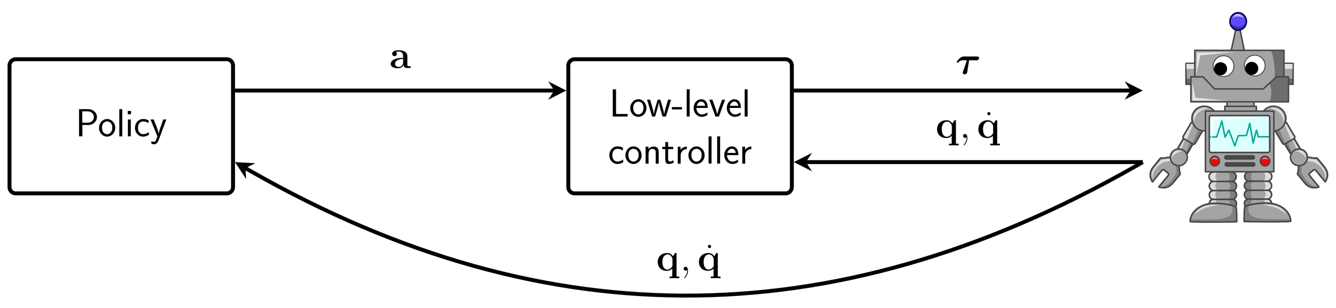
- The RL agent directly chooses the torques $\boldsymbol{\tau}$ applied on the robot



→ Direct control over the system but very low-level (the agent e.g., needs to learn to stabilize the system first)

High-level action representations

- Define an action representation $\mathbf{a}$ (e.g. desired joint positions)
- A low-level controller computes torques for the given the action

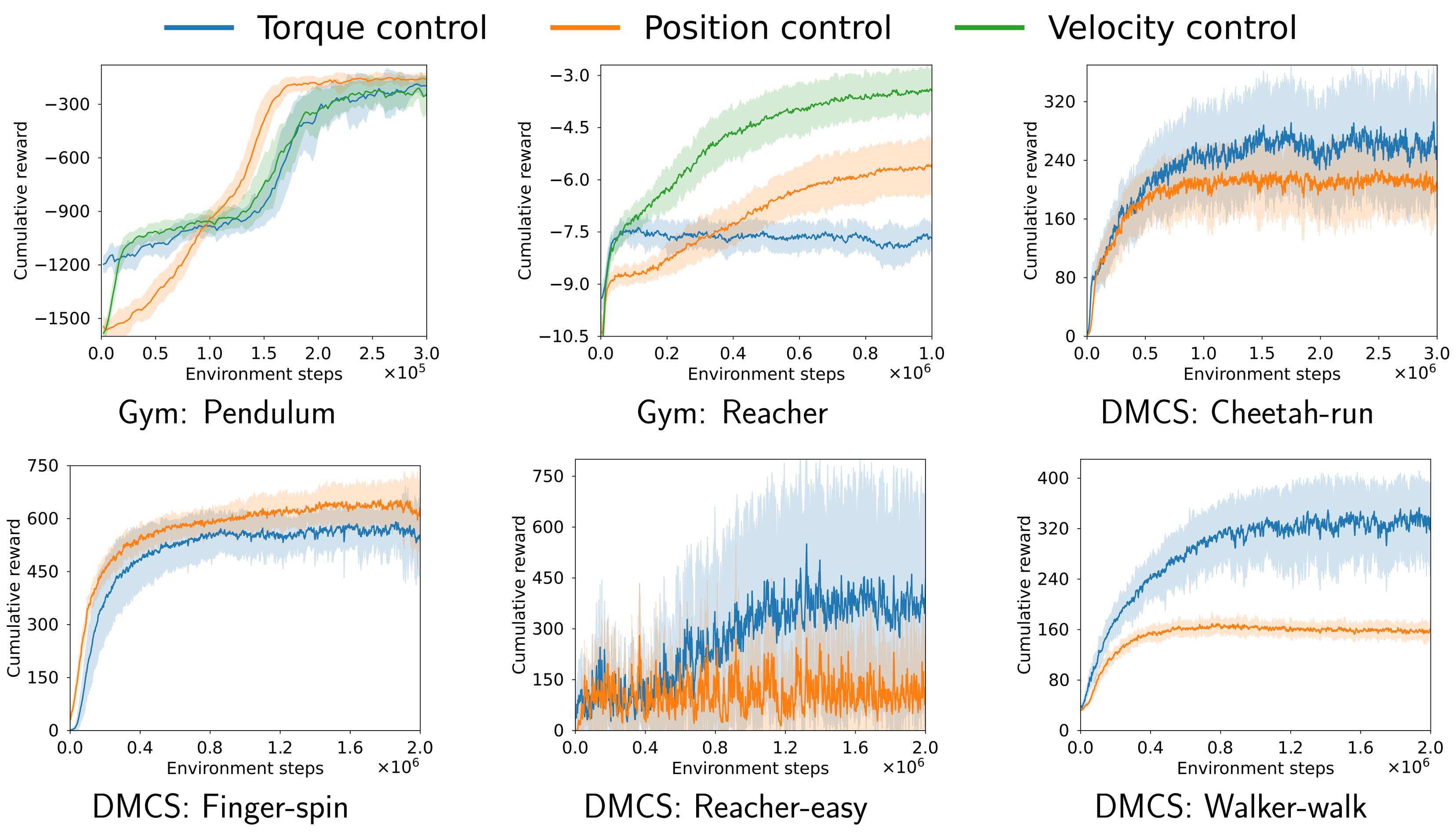


→ These representations can have beneficial properties (e.g., open-loop stability or robustness to perturbations)

- We compare *torques*, *joint positions*, and *joint velocities* as action representations for RL
- Position controller: $\boldsymbol{\tau} = K_p^{PC}(\mathbf{a} - \mathbf{q}) - K_d^{PC}\dot{\mathbf{q}}$
- Velocity controller: $\boldsymbol{\tau} = K_d^{VC}(\mathbf{a} - \dot{\mathbf{q}})$
- Controller gains $K_p^{PC}, K_d^{PC}, K_d^{VC}$ are tuned to minimize the tracking error

Learning performance

- Benchmark tasks from OpenAI Gym [1] and the DeepMind Control Suite (DMCS) [2]
- Learning performance of PPO [3] with different action representations

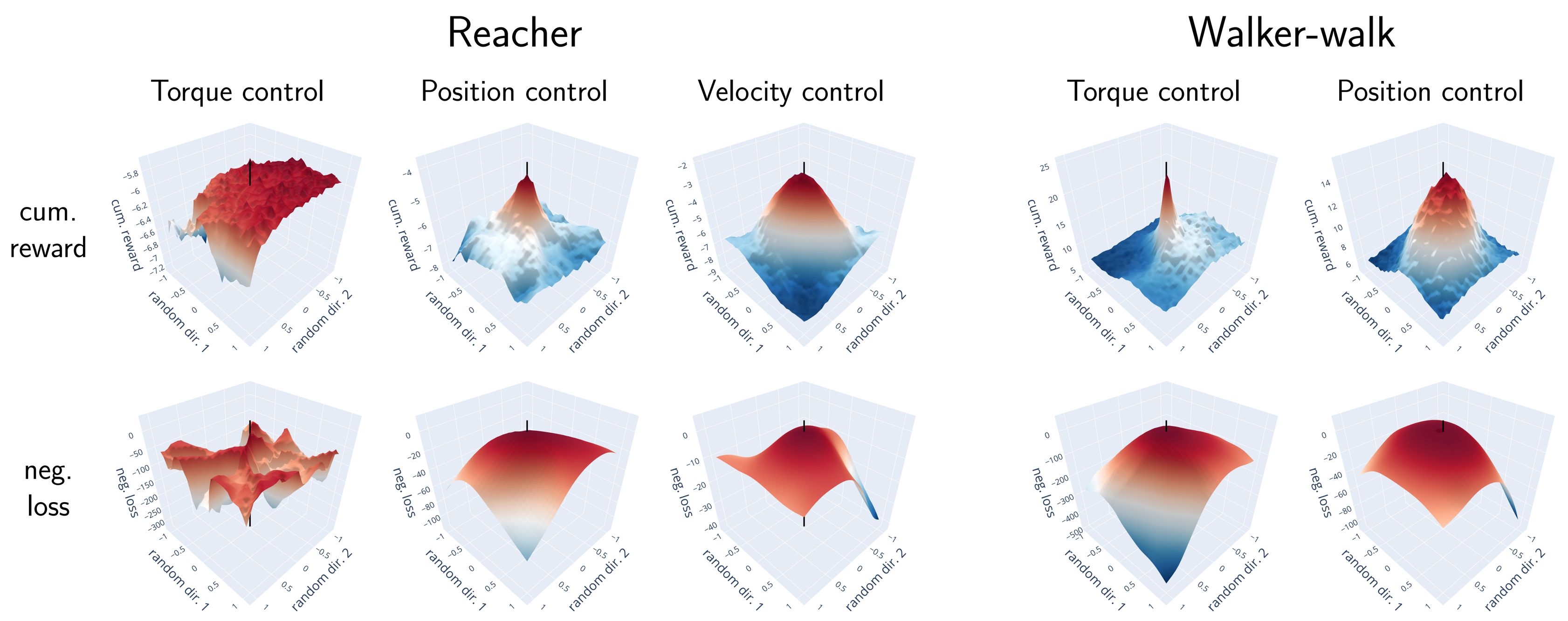


→ Action representations have a significant impact on learning performance
→ No representation is superior for all tasks

→ **These performance differences warrant further investigation into the influences on different components of the RL algorithm**

Analysis: Optimization landscape visualization

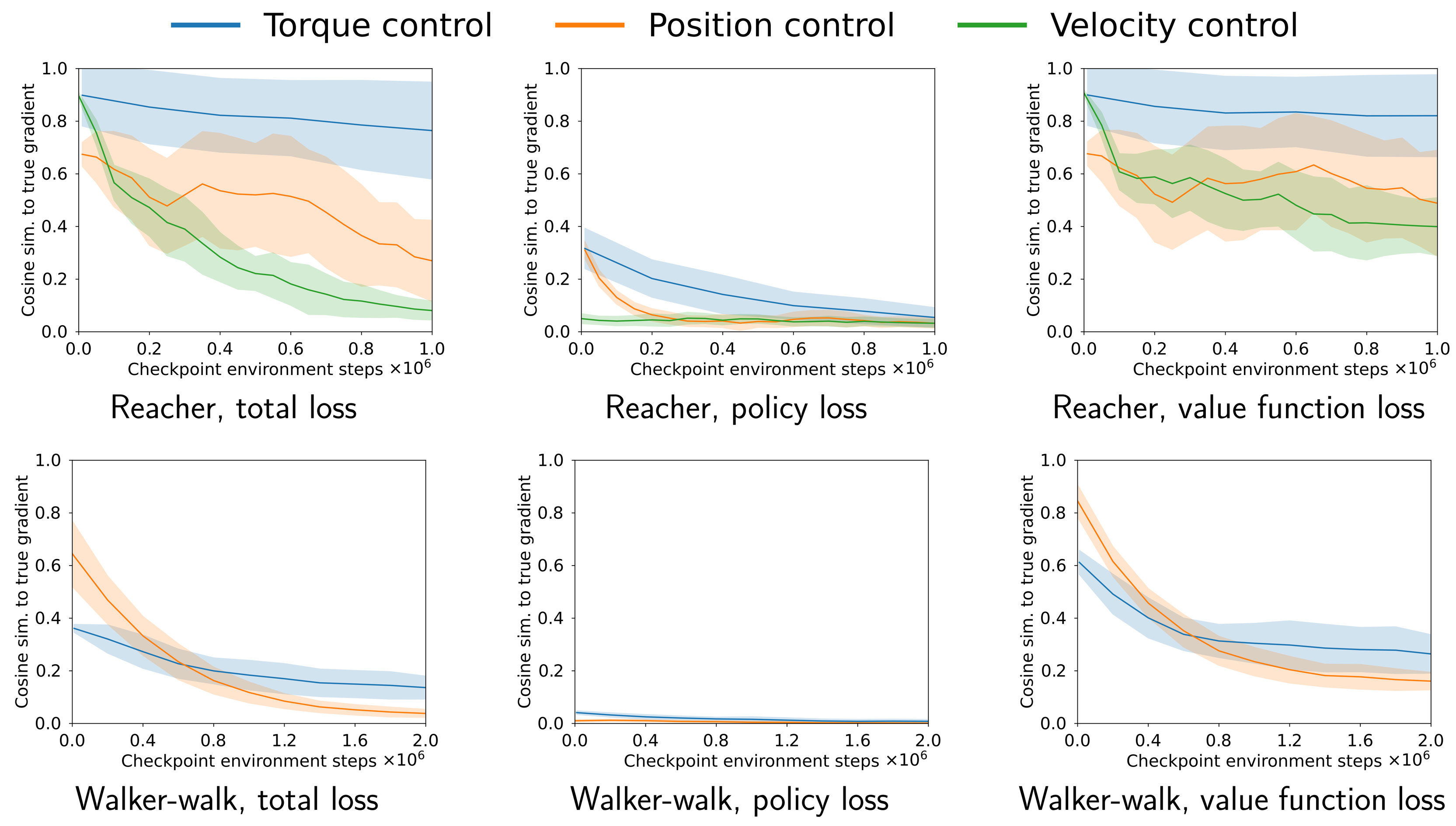
- Objective: Getting an intuition of the impact on the optimization difficulty
- Based on work of Li *et al.* [4]
- Due to the large number of parameters in neural networks, we cannot plot the optimization landscape directly
  → Dimensionality reduction: Plot along two random directions in parameter space
- Plot the values of two criteria
  – Cumulative reward (the true measure of policy performance)
  – Surrogate loss (the criterion that the algorithm optimizes)



→ Reacher, torque control: Rugged loss landscape explains poor learning performance
→ Other configurations: No clear intuition about the reasons for performance differences

Analysis: Gradient estimation accuracy

- Objective: Understanding the influence on the gradient estimation
- Based on work of Ilyas *et al.* [5]
- Approximate the true gradient with 10^7 samples (in comparison: 64 samples are used for gradient estimation during training)
- Compare cosine similarity between gradients used during training and this "true" gradient
- The PPO loss is the sum of a policy and a value function term
  → Plot the gradient quality also for each term individually



→ No clear correlation between gradient quality and learning performance
  → Higher policy performance makes gradient estimation harder
→ The gradient quality is significantly worse for the policy than for the value function

Open challenges of the analysis methods

- Normalizing the analysis results with respect to the learning progress
- Disentangling different effects on the RL algorithm
- Taking into account the effect of hyperparameters and controller gains

References

[1] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," *arXiv preprint arXiv:1606.01540*, 2016.
[2] S. Tunyasuvunakool, A. Muldal, Y. Doron, *et al.*, "dm_control: Software and tasks for continuous control," *Software Impacts*, vol. 6, p. 100022, 2020, ISSN: 2665-9638. DOI: https://doi.org/10.1016/j.simpa.2020.100022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665963820300099.
[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
[4] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
[5] A. Ilyas, L. Engstrom, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry, "A closer look at deep policy gradients," in *International Conference on Learning Representations*, 2020.