Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence
Canadian AI 2026
25–29 May 2026, Vancouver, BC

Beyond Information Sufficiency: Observation-Action Space Alignment in Robotic Reinforcement Learning

Vishal Bhat [†], Zahra Suleymanova [†], Colin Bellinger[†, ‡, *]
[†] University of Ottawa, School of Electrical Engineering and Computer Science
[‡] Vector Institute for Artificial Intelligence

Abstract

Observation design is a fundamental yet under-specified component of robotic reinforcement learning (RL). While classical theory emphasizes that observations should be informationally sufficient, we show—through a focused reaching case study—that sufficiency alone does not guarantee learnability or sim-to-real transfer. Using PPO on a 6-DOF Kinova Gen3 Lite arm, we demonstrate that two observation spaces with equal dimensionality and theoretically equivalent information content (9D joint-based vs. 9D Cartesian-based) differ by over 60 percentage points in success when paired with Cartesian velocity control. Aligned Cartesian observations consistently learn faster, achieve higher success, and transfer zero-shot to the physical robot, whereas misaligned joint observations fail despite being sufficient in principle. Our findings highlight representational alignment between observations, actions, and rewards as a first-order design constraint in robotic RL, demonstrated through controlled simulation and zero-shot real-world deployment.

1. Introduction

Deep reinforcement learning has enabled notable progress in robotic manipulation, yet policy design choices that are routine in practice, such as the structure of the observation space, remain weakly constrained by theory. Most guidance reduces observation design to *information sufficiency*: observations should render the task Markovian and contain all task-relevant variables. However, this criterion offers little practical direction when multiple representations encode the same information. In this paper, we study a common but under-examined failure mode in robotic RL: *misalignment between observation representations and the chosen action and reward spaces*. Our central claim is not that more information is required, but that task-relevant information must be provided in a representation compatible with the control interface and reward structure.

We investigate this question through a deliberately focused case study: Cartesian target-reaching with a 6-DOF Kinova Gen3 Lite arm, trained using PPO in simulation and deployed on hardware. This setting allows us to isolate representational effects without conflating them with task complexity. We show that a 9D joint-space observation (joint angles + target) fails to learn under Cartesian velocity control, despite being theoretically sufficient via forward kinematics. In contrast, an equally sized 9D Cartesian observation (end-effector position, velocity, and target) learns reliably and transfers zero-shot to the real robot. We interpret these results as identifying a principled failure mode that arises when observation representations are misaligned with action and reward definitions. We argue that representational alignment should be treated as a first-order design consideration in robotic RL.

2. Related Work

State and observation representations have long been recognized as central to reinforcement learning performance [1]. Traditional theory emphasizes sufficiency and the Markov property, while more recent work explores learned or compact state representations [2, 3]. However, most frameworks treat the action interface as fixed and do not analyze compatibility between observation structure and control parameterization.

[*] colin.bellinger@uottawa.ca

In robotics, action space design has been shown to significantly affect learning efficiency and sim-to-real transfer [4, 5]. These findings suggest that representation choices impose structural constraints on the learning problem. Cartesian velocity control extends this perspective to the observation space, showing that alignment constraints apply symmetrically to what the agent observes and how it acts.

Closest to our study, Kim and Ha [6] conduct systematic ablations of observation spaces in simulated locomotion tasks and show that observation design materially affects RL performance. However, their analysis is restricted to simulation and does not consider how observation effectiveness depends on the control interface or whether successful observation choices transfer to physical robots. We directly address this gap through sim-to-real experiments in a robotic arm reaching setting.

3. Task and Observation Design

3.1. Reaching Task

We study a 3D target-reaching task using a 6-DOF Kinova Gen3 Lite arm. At the start of each episode, a Cartesian target position $\mathbf{p}_{target} \in \mathbb{R}^3$ is sampled uniformly from the reachable workspace. Success is achieved when the end-effector position $\mathbf{p}_{ee}$ satisfies $\|\mathbf{p}_{ee} - \mathbf{p}_{target}\| < 0.03\,\mathrm{m}$.

Policies are trained under two control interfaces: (1) absolute joint position control and (2) Cartesian velocity control. Unless otherwise stated, sim-to-real deployment uses Cartesian velocity control.

3.2. Observation Spaces

We focus on three observation spaces central to our analysis. These are summarized in Table 1. Additional variants are presented in the appendix for completeness.

Observation	Components
Joints–Target (9D)	Joint angles $\mathbf{q}$, target $\mathbf{p}_{target}$
EE–Pos–Target (6D)	End-effector position $\mathbf{p}_{ee}$, target $\mathbf{p}_{target}$
EE–Pos–Vel–Target (9D)	$\mathbf{p}_{ee}$, $\dot{\mathbf{p}}_{ee}$, target $\mathbf{p}_{target}$

Table 1. Primary observation spaces evaluated in this study.

Notably, the Joints–Target observation is *informationally sufficient*: $\mathbf{p}_{ee}$ and $\dot{\mathbf{p}}_{ee}$ can be recovered analytically from joint angles and velocities. Our experiments examine whether this sufficiency translates into practical learnability under Cartesian control.

3.3. Experimental Setting

Figure 1 illustrates the experimental setup for both simulation and real-world evaluation. The task consists of a fixed-base 6-DOF Kinova Gen3 Lite arm reaching toward a point target sampled in 3D Cartesian space within the robot's reachable workspace. An example, randomly selected, target is visualized as a small sphere. The targets do not involve contact or orientation constraints.

In simulation, the robot model includes joint limits, inertial parameters, actuator dynamics, and a ground plane, and supports both joint-space position control and Cartesian velocity control. For real-world deployment, the same control interface is used via the Kinova Kortex API, with safety limits on workspace bounds and command velocities. This setting allows us to isolate the effect of observation–action representation while keeping task structure and control loop timing fixed across simulation and hardware.

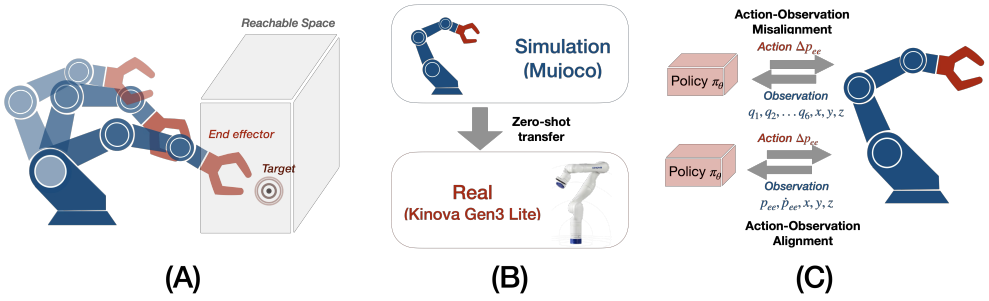

Figure 1. Experimental setup for Cartesian reaching. A target is randomly sampled within the reachable workspace of the Kinova Gen3 Lite arm. Policies operate either in joint space or Cartesian velocity space and are evaluated in both simulation and zero-shot hardware deployment.

4. Experimental Setup

Policies are trained in MuJoCo using PPO with a fixed hyperparameter configuration across all experiments. We randomize physical parameters and inject observation and action noise to encourage robustness [7]; full details are provided in the appendix.

Training is conducted with vectorized environments and normalized observations. Each configuration is trained with three random seeds. We report mean success and standard deviation over seeds, and select the best-performing checkpoint for hardware deployment. All observation normalization statistics and action scaling parameters used during training were preserved for evaluation and hardware deployment.

5. Results

5.1. Observation–Action Misalignment in Simulation

Table 2 summarizes success rates after 2.5M training steps. Results are reported as mean ± standard deviation over three seeds.

Observation	Joint Control	Cartesian Velocity
Joints–Target (9D)	83.2 ± 3.1	11.4 ± 7.6
EE–Pos–Target (6D)	66.5 ± 4.2	54.1 ± 5.3
EE–Pos–Vel–Target (9D)	74.3 ± 3.8	72.6 ± 4.1

Table 2. Success rates (%) under different observation–action pairings.

The key result is the collapse of the 9D Joints–Target observation under Cartesian velocity control. Despite containing all task-relevant information in principle, this representation fails to support learning when misaligned with the action space. In contrast, the aligned EE–Pos–Vel–Target observation succeeds with comparable dimensionality.

5.2. Interpretation

This disparity illustrates a representational bottleneck. Under Cartesian control, a policy operating on joint observations must implicitly learn forward kinematics and Jacobian mappings while simultaneously optimizing a velocity-dependent reward. While theoretically possible, this coupling substantially degrades learning. Aligned Cartesian observations eliminate this burden, enabling more direct credit assignment. While this interpretation

is consistent with the observed results, full causal verification of the mechanism is left for future work.

5.3. **Sim-to-Real Transfer**

Policies trained with Cartesian velocity control were deployed zero-shot on a physical Kinova Gen3 Lite arm. Joints–Target policies failed to transfer reliably, exhibiting inconsistent and unsafe behavior. In contrast, both Full and EE–Pos–Vel–Target policies transferred successfully. A video demonstrating zero-shot sim-to-real deployment of the learned policies on the physical Kinova Gen3 Lite arm is available at the following link: https://www.youtube.com/@infiniteloop9602/videos.Figure 2 shows representative distance-to-target trajectories across ten test targets. The EE–Pos–Vel–Target policy achieved consistent convergence with centimeter-level accuracy. While successful, policies trained with minimal observations exhibited longer convergence times and mild oscillations, mirroring trends observed in simulation.

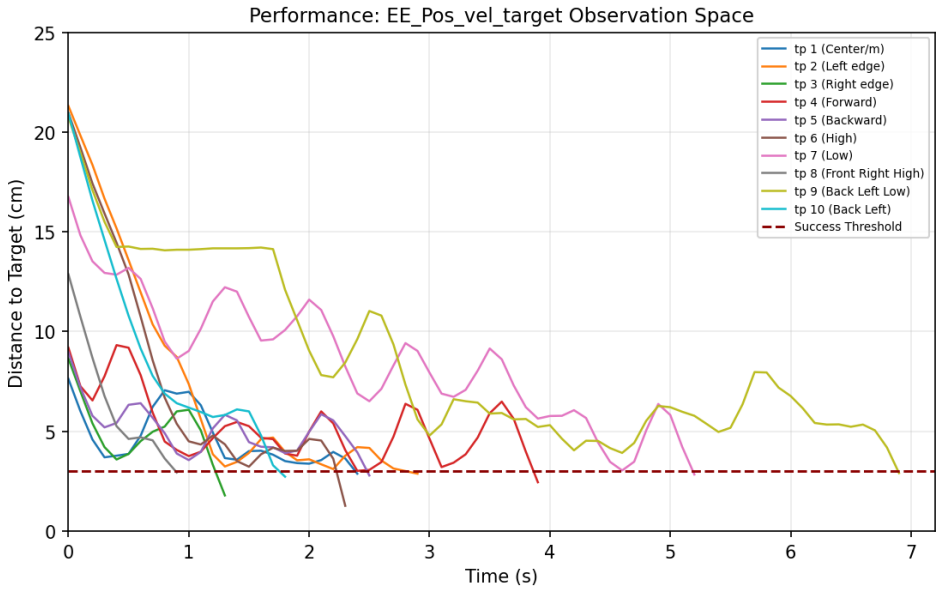

Figure 2. Zero-shot sim-to-real distance-to-target trajectories for the EE–Pos–Vel–Target policy across ten targets.

Although each policy was transferred from a single training run, consistent qualitative behaviour was observed across all evaluated target positions.

5.4. **Quantitative Real-World Performance**

To complement trajectory visualizations, Table 3 summarizes quantitative performance metrics measured on the physical robot across ten target positions. Each target was evaluated once per policy using the best-performing training checkpoint. We report success rate (distance $< 3\,\text{cm}$), time-to-success, and final distance error. All metrics are computed over the full episode duration. The results mirror simulation trends: minimal aligned observations enable successful zero-shot transfer, while redundant observations improve convergence speed and stability.

Observation	Success	Time (s)	Final Error (cm)
EE–Pos–Vel–Target (9D)	10/10	2.7 ± 0.8	2.2 ± 0.4
Full (21D)	10/10	1.9 ± 0.6	1.8 ± 0.3

Table 3. Real-world reaching performance on the Kinova Gen3 Lite across ten targets. Values show mean ± standard deviation.

6. Discussion

6.1. Mechanisms, Control Implications, and Limitations

Our core empirical finding is that observation–action misalignment can prevent learning even when observations are sufficient in principle. In the studied reaching task, joint-space observations paired with Cartesian velocity control consistently failed despite perfect observability, while aligned Cartesian observations learned reliably and transferred zero-shot.

A plausible explanation is that policies operating on joint observations under Cartesian control must implicitly learn forward kinematics and Jacobian mappings while simultaneously optimizing a velocity-dependent reward, a challenge previously noted in representation-sensitive robotic learning settings [2]. While our experiments do not isolate these mechanisms individually, indirect evidence supports this interpretation: removing Cartesian velocity from the observation degraded learning under velocity control, while adding joint information redundantly improved robustness without changing qualitative behavior.

An additional implication concerns the control paradigm itself. While joint-space control policies achieved high success in simulation, none transferred reliably to hardware in our setting. In contrast, Cartesian velocity control paired with aligned observations enabled stable zero-shot deployment. We hypothesize that velocity-based interfaces reduce sensitivity to timing mismatches, discretization error, and unmodeled dynamics; however, this paper does not attempt a systematic comparison.

This study is intentionally narrow—limited to a single reaching task, robot platform, and RL algorithm—and should therefore be interpreted as identifying a recurring failure mode rather than a universal guarantee.

6.2. Design Guidelines and Common Failure Modes

Beyond the specific reaching task studied here, our results suggest a set of practical guidelines for observation-design decisions in robotic reinforcement learning. These guidelines are not universal guarantees, but heuristics distilled from a failure mode that emerged consistently in both simulation and real-world deployment.

Guideline 1: Align observation frames with action frames. When actions are defined in task space, joint-space observations may be informationally sufficient yet practically ineffective. In our experiments, Cartesian velocity control paired with joint-based observations failed despite perfect observability in principle. Providing observations in the same coordinate frame as actions avoids implicit representation learning burdens and supports more stable credit assignment.

Guideline 2: Match observation content to reward dependencies. If the reward function depends on velocities or other temporal derivatives, those quantities should be included explicitly in the observation and in the same reference frame. Our ablations show that omitting Cartesian velocity while retaining a velocity-based reward significantly degrades learning, consistent with increased effective partial observability.

Guideline 3: Prefer minimal aligned observations, then add redundancy for robustness. A minimal, well-aligned observation space can be sufficient for learning and transfer, as demonstrated by the 9D end-effector position–velocity–target representation. However, redundant

observations (e.g., joint states) improve robustness, convergence speed, and stability near task boundaries, at the cost of additional sensing and computation.
Common failure mode. Across experiments, policies trained with misaligned observations did not fail gradually; instead, learning often collapsed entirely despite nominal simulation success or theoretical sufficiency. Such failures are difficult to diagnose using standard RL diagnostics, as they stem from representational structure rather than reward scaling or exploration deficiencies. We therefore recommend validating observation–action compatibility early, particularly when changing control interfaces between simulation and deployment.

Together, these guidelines frame observation design not as a passive enumeration of measurable variables, but as an active co-design problem alongside the control and reward structures that define the learning task.

7. Conclusion

We showed that representational alignment between observations, actions, and rewards is critical for robotic reinforcement learning, even when observations are theoretically sufficient. In a controlled reaching task, a 9D joint-based observation failed under Cartesian velocity control, while a 9D Cartesian observation succeeded and transferred zero-shot to real hardware. These results underscore the importance of treating observation design as a first-order system choice—on par with action and reward design, rather than a passive listing of available variables.

Declaration of AI Use: Generative AI was used for editing and clarity in this report; no AI system contributed to experimental design, results, or scientific claims.

References

[1] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press, 2018.

[2] R. Jonschkowski and O. Brock. "Learning State Representations with Robotic Priors". In: *Autonomous Robots* 39.3 (2015), pp. 407–428.

[3] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. "Learning Latent Dynamics for Planning from Pixels". In: *International Conference on Machine Learning (ICML)*. 2019, pp. 2555–2565.

[4] E. Aljalbout, F. Frank, M. Karl, and P. van der Smagt. "On the Role of the Action Space in Robot Manipulation Learning and Sim-to-Real Transfer". In: *IEEE Robotics and Automation Letters* 9.6 (2024), pp. 5895–5902.

[5] R. Martín-Martín, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg. "Variable Impedance Control in End-Effector Space: An Action Space for Reinforcement Learning in Contact-Rich Tasks". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 1010–1017.

[6] J. T. Kim and S. Ha. "Observation Space Matters: Benchmark and Optimization Algorithm". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 2507–2513.

[7] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. "Sim-to-real transfer of robotic control with dynamics randomization". In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 3803–3810.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. *Proximal Policy Optimization Algorithms*. arXiv preprint arXiv:1707.06347. 2017.

Appendix A. Appendix

A.1. Additional Observation Spaces

For completeness, we also evaluated:

- **Full (21D):** joint positions and velocities, end-effector pose and velocity, and target
- **EE–Pose–Target (10D):** end-effector position, orientation (quaternion), and target

The Full observation consistently improved robustness and reduced variance but did not change qualitative conclusions.

A.2. Reward Function

The reward combined distance shaping, progress, velocity alignment, action regularization, and a success bonus. Velocity-dependent terms explicitly reference Cartesian end-effector velocity, motivating the need for aligned velocity observations. Full equations and coefficients are provided here for reproducibility:

$$R_{dist} = \exp(-8\|\mathbf{p}_{ee} - \mathbf{p}_{target}\|) \tag{A.1}$$

$$R_{vel} = 0.05\langle \dot{\mathbf{p}}_{ee}, \hat{\mathbf{d}} \rangle \tag{A.2}$$

$$R_{act} = 0.001\|a_t\|^2 \tag{A.3}$$

A.3. Domain Randomization and Noise

At episode reset, we randomized masses, friction coefficients, joint damping, and actuator gains within ±5–20% of nominal values. Gaussian noise was injected into observations and actions to mitigate overfitting. These settings were fixed across all experiments.

A.4. Training Details and Example Learning Curves

All policies were trained using PPO with identical network architectures, learning rates, and rollout horizons. Normalization statistics were frozen at evaluation time and reused for hardware deployment. Figure 3 shows representative training curves for minimal and full observations under Cartesian control, illustrating higher variance for minimal representations.

Hyperparameter	Value
Algorithm	PPO [8]
Policy network	MLP, 2 hidden layers × 256 units, tanh
Value network	MLP, 2 hidden layers × 256 units, tanh
Learning rate	3×10^{-4} (linear decay to 0)
Clip ratio ϵ	0.2
Entropy coefficient	0.01
GAE λ	0.95
Discount factor γ	0.99
Rollout length per env	2048 steps
Mini-batch size	256
PPO epochs per update	10
Parallel environments	16
Total training timesteps	2.5 M
Observation normalization	Running mean/variance (frozen at eval)
Action scaling	Clipped to $[-1, 1]$, scaled to velocity limits
Domain randomization (per episode reset)	
Body mass	$\mathcal{U}(0.95\, m_i^{\text{nom}},\ 1.05\, m_i^{\text{nom}})$
Contact friction	$\mathcal{U}(0.8\, \mu_i^{\text{nom}},\ 1.2\, \mu_i^{\text{nom}})$
Joint damping	$\mathcal{U}(0.8\, d_i^{\text{nom}},\ 1.2\, d_i^{\text{nom}})$
Actuator gain	$\mathcal{U}(0.9\, k_i^{\text{nom}},\ 1.1\, k_i^{\text{nom}})$
Obs. noise σ_o	0.002
Action noise σ_a	0.01
Position noise σ_p	0.003 m
Deployment	
Control frequency	10 Hz (`TwistCommand` via Kortex API)
Episode length	200 steps
Success threshold ϵ	0.03 m

Table 4. Full hyperparameters and implementation details.

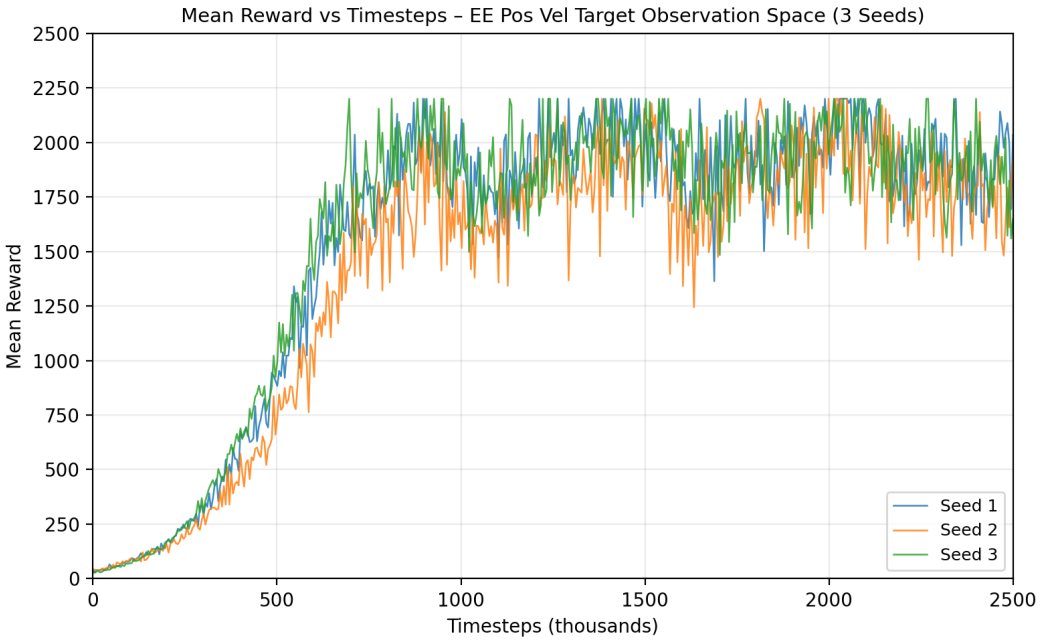

Figure 3. Training reward curves across three random seeds (Cartesian velocity control).