Learning to Manipulate from Pixels on Rigid Body Robots with a Kinematic Critic

Johanna Hansen[*12†] Kyle Kastner[*23] Yuying Huang[14]
Aaron Courville[23‡] Dave Meger[12†] and Gregory Dudek[12†]

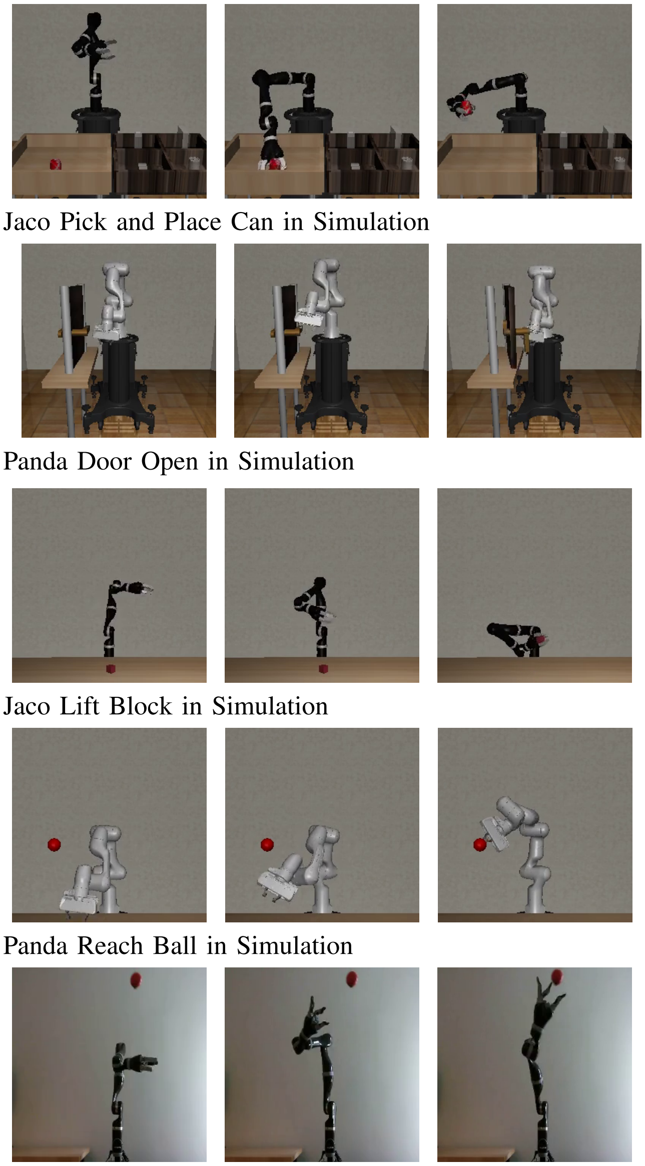
***Abstract*— This paper introduces a pixel-based actor-critic architecture featuring a differentiable Denavit-Hartenburg (DH) forward kinematics function in the critic sub-network, which achieves substantial improvement in average cumulative reward across several complex manipulation tasks and two robot arms in Robosuite [1], compared to strong baselines. Forward kinematics as described by DH parameterization for rigid-body robots is fully differentiable with respect to input joint angles, given fixed link-relative geometric information, and including this differentiable module improves training of reinforcement learning agents on an array of benchmark manipulation tasks. We show the importance of formulating a differentiable kinematic function for overall task performance in an ablation study, and demonstrate a simulation-learned policy running on a real Jaco 7DOF robot.**

I. Introduction

This paper examines the classic Denavit-Hartenburg (DH) [2] method of parameterizing robot kinematics as a differentiable function for improving pixel-based actor-critic reinforcement learning [3], [4], [5], [6] on several robotic manipulation tasks [1]. The DH method of joint parameterization was developed in 1955 by Jacques Denavit and Richard Hartenberg [2] and has been a staple tool for defining robotic kinematic functions. Modern automatic differentiation tools [7], [8] enable easy incorporation of this function (and its Jacobian) into deep learning models, including deep reinforcement learning frameworks.

Robots learning to solve manipulation tasks must inherently reason about controlling their own body. Models and controllers based on explicit analytic physical parameterization traditionally need detailed information such as manipulator redundancies, kinematic limits, friction, acceleration, and/or inertia, which can be difficult to define exhaustively and measure accurately. When solving tasks without explicit models, such as in model-free reinforcement learning (RL), dynamics and structure prediction is inherently coupled with task-solving in the agent. While recent methods learning model-free robot control from images such as DrQv2 [6] have strong performance on benchmarks, in this paper we show how providing a well-described differentiable robot kinematics function improves agents, allowing the agent to leveraging partial knowledge about the system kinematics to improve performance.

The main contributions of this paper are as follows:


Jaco Pick and Place Can in Simulation

Panda Door Open in Simulation

Jaco Lift Block in Simulation

Panda Reach Ball in Simulation

Jaco Reach Ball Real

Fig. 1: Training visual policies with a Denavit-Hartenburg view of robot joint positions enables agents to learn complex visual policies that can operate in the real world. This figure depicts successive frames of our Kinematic Critic performing robotic manipulation tasks. Full videos can be found at `https://johannah.github.io/kinematic-critic`

*Equal Contribution. [1]McGill University, Montreal, Quebec. [2]Mila, Montreal, Quebec. [3]Université de Montréal, Montreal, Quebec. [4]Stanford University, Stanford, California. [†]Supported by the National Canadian Robotics Network (NCRN) [‡]CIFAR Fellow.

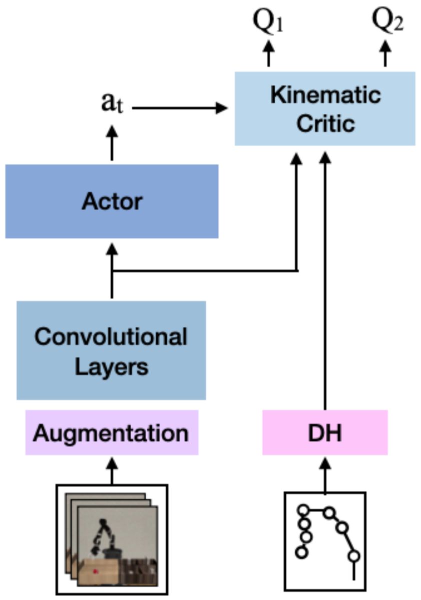


Fig. 2: Kinematic Critic Architecture: differentiate through the Denavit-Hartenburg Kinematics during training. Note that the sampling of action a_t admits differentiation with respect to the action distribution predicted by the actor, using the reparameterization trick [9], [10].

- We describe a method to incorporate forward kinematics with the Denavit-Hartenberg (DH) function in an automatic differentiation framework for use in a deep reinforcement learning algorithm, and show how to incorporate this structure for rigid body robots.
- We demonstrate notable improvement on a suite of robot manipulation benchmarks in simulation on Jaco and Panda robots using a pixel-based actor-critic algorithm compared to strong baselines.
- We show the applicability of our system, trained only in simulation, on a real Jaco robot with a hybrid simulation-real approach.

II. Background

A. Kinematics

Forward kinematics allow computing a robot's kinematic chain for a particular configuration to find the pose of the end-effector from *joint parameters* and joint angles, where *joint parameters* are known constants that define the geometric relationship between the serial links of a rigid body chain. There are several conventions for assigning joint parameters [11], but in this paper we use the popular Denavit-Hartenburg (DH) [2] method of rigid body parameterization, which is differentiable with respect to input joint angles given geometric information such as lengths and relative link rotations (see Fig 4).

For each joint, i, in a rigid body, the DH parameters are described by d_i (distance from joint i to the actuator axis $i-1$), θ_i (angle rotation about axis $i-1$), a_i (the distance of joint i along actuator axis $i-1$), and α_i (the angle between actuators of axis i and axis $i-1$).

$$[T] = \prod_{i=1}^{n} {}^{i-1}T_i(\theta_i) \tag{1}$$

In this convention, rigid body transformations are serially performed between n links with joint parameters θ_i as specified in Eq. 1 where ${}^{i-1}T_i(\theta_i)$, shown in Eq. 2 describes the transformation from link i to the previous link.

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

DH parameters as defined by the robot geometry are assumed constant throughout training, and not directly optimized via backpropagation or other means in this work. However, given a fixed set of DH parameters, it is possible to describe the Jacobian from end-effector position, through the DH transformation into joint angle space and thus any preceding functions (such as neural network layers of a policy network), thus enabling use of this function as part of a deep neural network trained by backpropagation. The importance of this differentiability is further detailed in Sec.IV.

B. Learning Controllers

The choice of controller to use for robot learning can have a large impact on task difficulty - for instance a ping-pong robot will need precise control of both end-effector pose and velocity, while the task of box stacking may be simpler to learn with a Cartesian controller [12]. Many manipulation agents utilize Cartesian controllers [13] where low-level control of joints is handled by a controller specified from expert robot knowledge [14], [15], [16], [17]. In Cartesian (also known as Task) Space, actions are the target position and/or orientation of the end-effector in Euclidean space (x, y, z). This high level control method can simplify many tasks, but has some drawbacks. General Cartesian controllers make assumptions about the operating environment and agent structure in order to perform the complex task of Cartesian space to joint space mapping. These assumptions may make accounting for nuance such as correcting for a grasped object's weight or avoiding obstacles more difficult.

At the other end of the control spectrum are agents which learn to control robots directly from torque applied to motors [18], [6], [19], [20], [21], [22], without the use of higher level controllers. This can make solving tasks requiring long-term planning challenging, as agents typically need to operate at higher and less forgiving control rates while also learning physical dynamics. In this paper we utilize joint space control to enable the direct applicability of the DH parameterization for kinematics. Other works [23], [24] have demonstrated joint position control to be a feasible method for controlling RL agents on real robots. JAiLeR [24] details a paradigm for training a joint position controller which maps from Cartesian space to joint space using model-free RL on

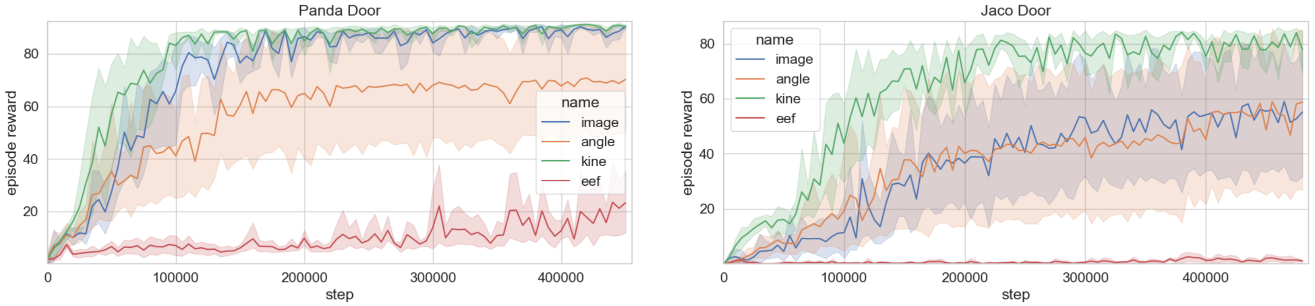


Fig. 3: Evaluation curves depicting the mean (solid line) and the shaded 95 percent confidence interval around the mean, with performance measured over 5 randomly chosen seeds. The Kinematic Critic architecture (green) outperforms other ablations on average, including agents with critics which directly calculate the expected end-effector pose without back-propagating gradient information (red) and agents which are given robot joint angles directly (orange). The stark discrepancy between the green and red reward curves emphasize the power of allowing gradient propagation.

proprioceptive state observations. JAiLeR also employs curriculum learning to produce an inverse kinematics controller with similar performance to OSC on goal-conditioned reach tasks, demonstrating the capability to incorporate obstacle avoidance directly into the observation state space of the agent.

C. *Model-Based Control*

A common example of a controller using physical attributes is in the widely used [25] Cartesian-space Operational Space Controller [13]. OSC relies heavily on the accuracy of the physical parameterization of the robot, particularly, the mass matrix which can vary based on load and configuration. Controller performance can deteriorate quickly if this physics estimate is inaccurate [25].

There is a large field of research on developing models for use by robot controllers. One can impose varying degrees of prior knowledge about the system into the model, including kinematic equations and factors such as mass, friction, or inertia [26], [27], [28]. Despite its appeal, prescribed knowledge can be difficult or idiosyncratic to define for particular robots but can potentially provide both generalization and interpretability. On the other hand, purely data-driven models [21], [29] use function approximators to fit the complex dynamics that govern robot control. Data-driven approaches tend to be limited by the coverage of their dataset, but may provide significant performance benefit over heavily prescribed and factorized models, if true system characteristics differ from those prescribed by model factorization but are well captured in the dataset itself.

Most practical systems are a blend of prescribed physics with learned models that attempt to overcome the inevitable dynamics errors of our physics assumptions in complex robots. Williams *et al.* [30] demonstrate the power of learning a controller with factorized dynamics models. They incorporate kinematic equations on several robotic systems, including an aggressive driving task. Model-based Action-Gradient-Estimator Policy Optimization (MAGE) [31] is a continuous-control DDPG actor-critic algorithm which explicitly trains the critic to provide action-gradients by backpropagatating through a learned dynamics model. OSCAR [15], introduces a data-driven variant of OSC which learns to adapt the physics model online for task-specific and task-agnostic manipulation. For a more exhaustive review of design choices in incorporating learned dynamics models with physics, refer to Lutter *et al.*[26].

D. *Differentiable Physics Engines*

Incorporating differentiable physics into robot learning agents has become increasingly possible thanks to the continually improving fidelity and speed of simulators [32]. For instance, Deluca [33], a Jax-based library introduces several fully differentiable classic control tasks and ChainQueen [34] presents a real-time differentiable simulator for deformable objects. Millard *et al.* [35] demonstrate a differentiable simulator for rigid body dynamics with realistic integrators which are constrained to the laws of physics. The strength of their simulator accuracy is demonstrated over long-horizon adaptive model-predictive control (MPC).

E. *Learning Pose with Kinematic Constraints*

We study the problem of learning to solve manipulation tasks from images, where the robot must complete a task with state information being provided from camera observations and has access to some well-defined proprioceptive information, such as joint angles. When learning robot policies from images, the agent must implicitly learn to map the image and its actions back to its own pose. Learning this action-observation mapping of multi-link articulated objects is inherently difficult because the pose is not only high dimensional, but also has structural constraints inherent in the rigid body chain that makes up the robot. Past work on mixture density estimation using neural networks [36] utilizes a neural network which, given input joint angles and assuming fixed link lengths, predicts the parameters of a mixture distribution over end-effector positions for robot kinematics of a simplified two link arm. Though this work was not utilized directly for control, this illustrative example serves as an introduction to the more general kinematics concepts at play in this work. Deep Kinematic Pose Regression [38] embeds a differentiable kinematic object model into a neural network for predicting pose from images. Their network predicts the joint motion parameters of the object, while learning directly on the joint location loss described by a kinematics chain or kinematics tree. This model is demonstrated on a toy 2D robot and 3D human pose, achieving state-of-the-art results on the Human3.6M dataset. The formulation of the kinematics loss based on pixel input bears similarity to the overall approach featured in our work, however Deep Kinematic Pose Regression was used in a supervised learning setting with a direct loss on pose, rather than for robotic control.

F. Learning Kinematic and Robot Tasks from Images

There have been an enormous spectrum of papers published on learning kinematic [39] and robotic tasks from images [22] and/or expert traces [40]. The key concepts which unify many of these works are:

- Prior knowledge of the *skeleton* of the agent, including important information such as the number of links and link length
- Focus on task specific training schemes (sometimes with the inclusion of expert traces, behavior cloning, and/or imitation learning)
- Frequent use of curriculum learning, replay buffers, and other training techniques common to the fields of reinforcement and continual learning.

Many of these methods also utilize a combination of auxiliary objectives, in addition to the main task loss or task reward. In Time Contrastive Networks (TCN), Sermanet *et al.* [41] use self-supervision to learn representations of human and robotic behaviors from unlabeled videos. The time contrastive training scheme allows the the model learn representations of humans and robots, enabling a robot to predict its own joint angles given an image of itself, and demonstrating a robot imitating human motion without explicit joint-level correspondences between the target robot platform and human demonstrations. Asymmetric Actor-Critic [42] employs an actor-critic training algorithm in which the critic is trained on full state observations, while the actor only receives image observations. They emphasise the benefits of fully utilizing the simulator for speeding up training with auxiliary tasks, and adding robustness in sim2real.

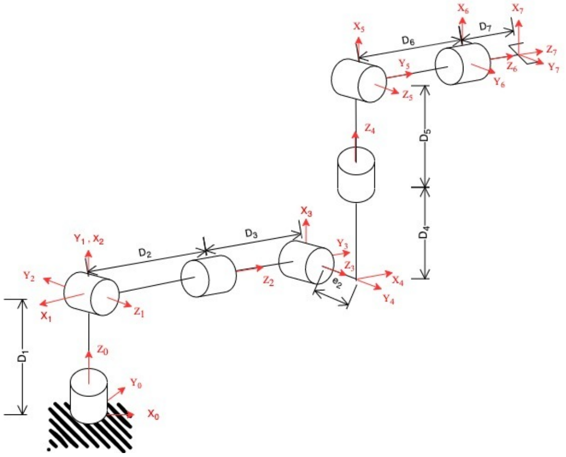


Fig. 4: DH parameters for the Jaco 7DOF Robot as given by the robot manufacturer [37]. DH parameters are usually publicly available for commercial robots, but can also be found by measuring DH parameter lengths directly on the physical robot.

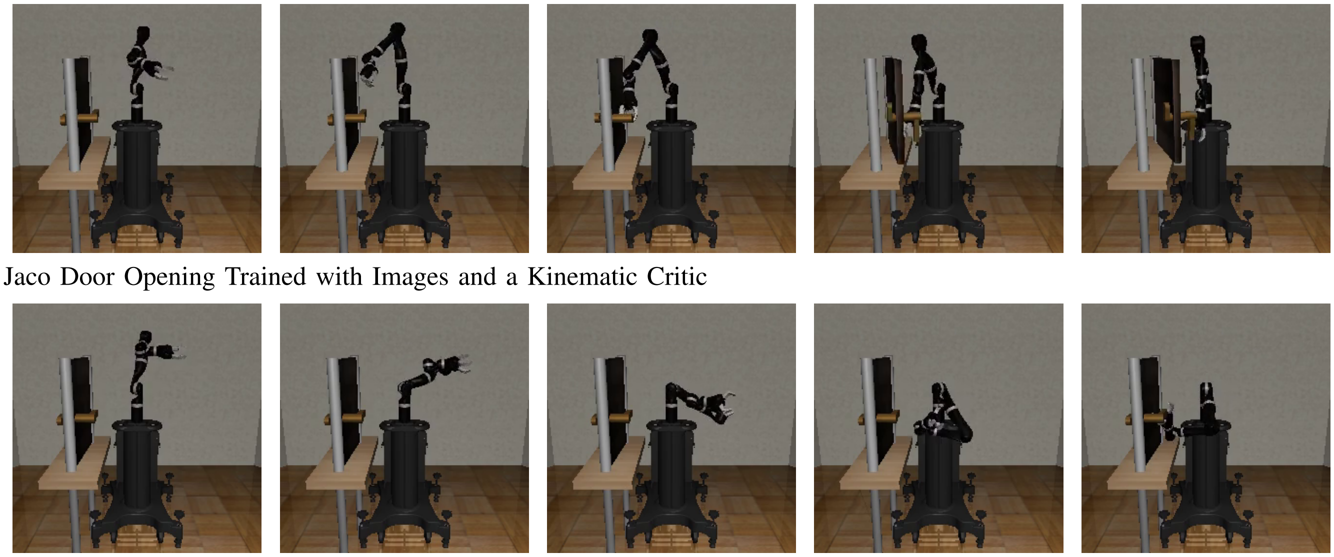
III. Method

We employ a Deep Deterministic Policy Gradients (DDPG) [3] reinforcement learning agent that utilizes an actor-critic [4] network structure. As in [42], our approach uses the fact that the actor and critic are two separate networks to give extra information to the critic during training. Our reinforcement learning agent is built on the DrQv2 [6] network architecture. DrQv2, which iterated on its predecessor DrQ [18], utilizes image augmentation to achieve sample-efficient high performance on continuous robot controls tasks. DrQv2 employs a DDPG learner with uses n-step returns to estimate TD error. All of our environment hyperparameters match the official implementation of DrQv2, aside from the replay buffer size, which we adjust from 1e6 to 500k stored image-states. To adapt DrQv2 from Deepmind Control Suite [43] Torque Control to Robosuite [1] Joint Position Control, we do not repeat actions, instead the agent runs its controller requesting relative joint angles at a rate of 10Hz. We adapt the *image* critic architecture from DrQv2, with *angle*, *eef*, and *kine* ablations. In the *angle* ablation, current robot joint angles are concatenated with the image observation (after encoding by a convolutional network) and actions in the critic. For both *eef* and *kine*, we estimate the expected pose of the end-effector for a given relative action by adding the current joint angles to the action and running the forward kinematics. In *kine*, this operation allows differentiation, and can be used by the agent as a gradient path for learning and backpropagation, but in the *eef* experiments, we *prevent* gradient propagation through the DH function to study the importance of *backward flow* in training the overall architecture, while providing access to roughly the same information as *kine* overall.

A. Relating This Work to Background Materials

The primary differences of our method to the aforementioned are as follows

- Our work builds directly on a strong actor-critic visual RL baseline with no use of expert traces, imitation learning, or behavior cloning.
- We utilize dense rewards from the environment and do not formulate pose-specific losses or employ multi-task training. We provide end-effector pose information (via a differentiable DH function) through the internal workings of the critic sub-network, thus allowing the overall actor-critic model to decide how best to use this



Jaco Door Opening Trained with Images and a Kinematic Critic

Jaco Door Opening Trained with Images Only

Fig. 5: Representative traces on the door opening task from an agent with access to a differentiable DH forward kinematics function during training, and the baseline with access only to images. In general we observe a much higher coordination among the joints on agents trained with a Kinematic Critic, even when rewards are similar.

information to maximize overall reward. This means that useful pose related dynamics (such as smoothness or stability) are learned implicitly, without explicit pose specific losses or rewards.

- Rather than using the entire feature-based state information in the critic like Asymmetric Actor-Critic [42], we use only joint angles of the robot along with the DH parameterization to improve training. This is advantageous as joint angles are likely to be well-modeled in sim2real transfer for a variety of robot platforms.
- Dynamics learning is relegated to the underlying RL agent, and we do not explicitly factorize dynamics learning using physics knowledge - only kinematic structure, on a per-timestep basis, is used. While kinematic structure in this work is closely related to the adjoint calculations used in the Articulated Body Algorithm (ABA) [44], [27], as well as the kinematic calculations in Deep Kinematic Pose Regression [38], QuaterNet [39], or many other works utilizing kinematic chain or tree calculations, it is not coupled with estimations of velocity, acceleration, mass, inertia, and other physics related quantities. Instead we utilize a combination of RL and standard joint-position controllers for overall problem dynamics.
- We operate directly from pixels, with no goal conditioning or state information [24] beyond knowledge of the robot geometry (assumed constant throughout training on a per-task basis, and given to the critic sub-network) and robot joint angles (which are also used by the controller). The actor only consumes images as inputs, as is common in continuous control from pixels work in reinforcement learning [6], [19].

IV. Experiments and Discussion

We investigate the impact of adding differentiable kinematics structure into the critic of an image-based reinforcement learning architecture for a set of tasks trained in Robosuite [1], a robotics simulation framework powered by the MuJoCo physics engine [45]. For all experiments, we utilize a Joint Position controller with fixed impedance parameters. The joint position controller has a max action step of 0.15 radians for all joints. We tune the controller for Jaco to set KP and damping ratios for the 7 joints to $(30, 60, 50, 70, 60, 70, 90)$ and $(0.1, 0.17, 0.2, 0.3, 0.1, 0.1, 0.1)$, respectively.

We demonstrate performance on 4 tasks: Door, Reach, Lift, and Can (in approximate order of increasing difficulty) for the Jaco 7DOF manipulator and the Panda 7DOF arm. Agents view the scene with a single RGB camera and receive a dense reward for all tasks. In the PickPlaceCan task, we greatly improved sample efficiency on all agents by adapting the dense reward in Robosuite to include a *touching* reward which encourages each fingerpad of the manipulator to make contact with the object of interest (in this case, the can). This touch-based reward was especially important in the compliant 3-finger Jaco gripper where collecting the benchmark *grasp* reward was difficult as it required caging with all fingers. Object position and robot initial joints are randomized on reset, with range of variability set per task.

Our experiments show that the differentiability of the DH function is critical to driving effective learning of the network. This is particularly evident in the reward traces for the Jaco Door opening task. Without propagating information from the DH function inside the critic, through the action space, into the actor and convolutional encoder networks,

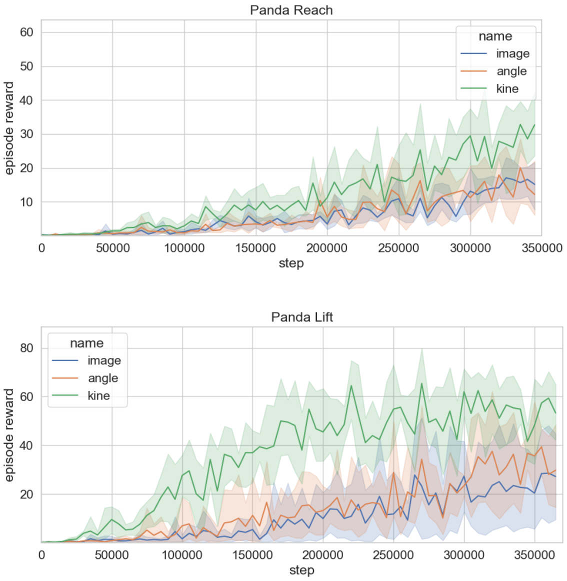


Fig. 6: Panda evaluation curves depicting the mean (solid line) and the 95 percent shaded confidence interval around the mean over 5 randomly chosen seeds. Our agents, using the Kinematic Critic architecture (green), outperform other agents which have access to only images (blue) and image agents with critics which are also given robot joint angles directly (orange).

we see performance far below baseline levels (traces labeled *eef*, in red vs *kine* in green). Coupled with improved performance when using differentiable DH (trace labeled *kine*, in green), this shows the importance of gradient information for improved agent learning compared to other methods which supply proprioceptive information but do not admit backpropagation or automatic differentiation.

We also demonstrate the Reach policy on a real Jaco arm (see Fig.1(m)). To test the learned agent controller, we first sync the simulator to the real setting, approximately matching ball position between the two modalities, then feeding a set of simulator images to the policy model. We then predict actions based on the simulator frames, apply the predicted action to the real robot, update the simulated robot pose as the real arm moves during the experiment, and again feed the resulting simulated visual observations into the agent to repeat the prediction loop.

Despite the limitations of this hybrid simulation-real approach, we see success in utilizing simulation learned policies in a real world setting. More sophisticated approaches utilizing domain adaptation or sim2real methods [46] to better blend simulation and real world operation could drastically improve this setting, but fall outside the scope of this publication.

V. Conclusion

In this work, we introduce a method of incorporating differentiable Denavit-Hartenburg (DH) transformations into an actor-critic reinforcement learning algorithm based on DrQv2 [6]. By training a critic with access to DH while training its actor only on images, we learn vision-based policies for complex manipulation tasks. Evaluation shows that the differentiability of the DH transformation in the critic is crucial for effective training, and overall our method improves upon strong actor-critic baselines across numerous benchmark tasks. The learned policies are directly transferrable to real-world settings, and we demonstrate this by tasking a real-world robot using simulation learned policies.

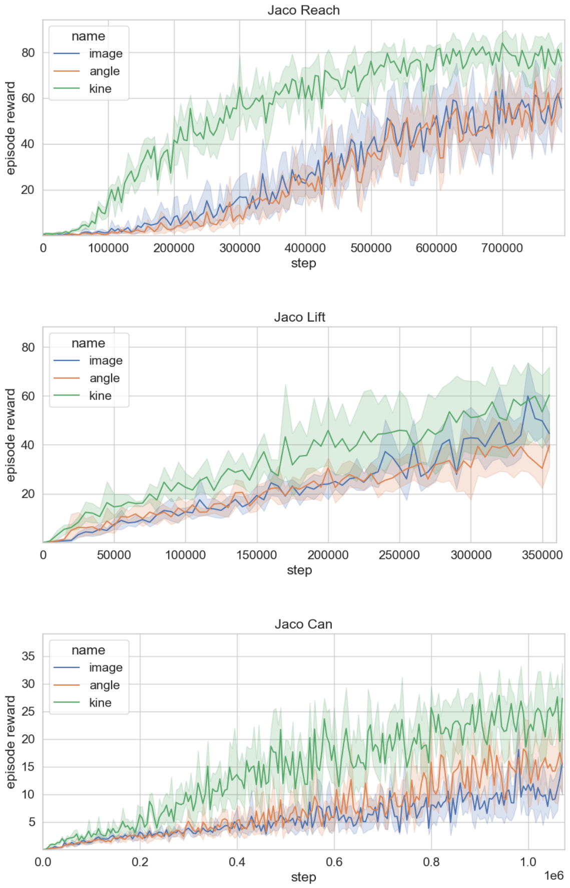


Fig. 7: Jaco Evaluation curves depicting the mean (solid line) and the 95 percent shaded confidence interval around the mean over 5 randomly chosen seeds. Our agents using the Kinematic Critic architecture (green) on average outperform other tested agents which have access to only images (blue) and image agents with critics which are given robot joint angles directly (orange).

ACKNOWLEDGMENTS

Thanks to Francois Hogan for discussion and advice on all things manipulation. We are also thankful to the members of the Mobile Robotics Lab who helped facilitate real-world experiments, especially Sahand Rezaei Shostari, for discussions on robot controllers, and Melissa Mozifian, for advice on the sim2real interface.

References

[1] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín, "robosuite: A modular simulation framework and benchmark for robot learning," *CoRR*, vol. abs/2009.12293, 2020. [Online]. Available: https://arxiv.org/abs/2009.12293

[2] R. S. Denavit, Jacques; Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Trans ASME J. Appl. Mech*, vol. 23, pp. 215–221, 1955.

[3] 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," 2019.

[4] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds. MIT Press.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[6] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, "Mastering visual continuous control: Improved data-augmented reinforcement learning," *arXiv preprint arXiv:2107.09645*, 2021.

[7] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[8] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/google/jax

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[10] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, pp. 229–256, 1992. [Online]. Available: https://doi.org/10.1007/BF00992696

[11] "A comparison between the denavit hartenberg and the screw-based methods used in kinematic modeling of robot manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 4, pp. 723–728, 2011, conference papers of Flexible Automation and Intelligent Manufacturing.

[12] 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 *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1010–1017.

[13] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *The International Journal of Robotics Research*, vol. 14, no. 1, pp. 19–36, 1995.

[14] J. Hansen, F. Hogan, D. Rivkin, D. Meger, M. Jenkin, and G. Dudek, "Visuotactile-rl: Learning multimodal manipulation policies with deep reinforcement learning," in *International Conference on Robotics and Automation*, 2022.

[15] J. Wong, V. Makoviychuk, A. Anandkumar, and Y. Zhu, "Oscar: Data-driven operational space control for adaptive and robust robot manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[16] S. Sodhani, A. Zhang, and J. Pineau, "Multi-task reinforcement learning with context-based representations," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9767–9779.

[17] J. Peters and S. Schaal, "Reinforcement learning by reward-weighted regression for operational space control," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 745–750.

[18] D. Yarats, I. Kostrikov, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=GY6-6sTvGaf

[19] A. Srinivas, M. Laskin, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," *arXiv preprint arXiv:2004.04136*, 2020.

[20] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[21] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*. Citeseer, 2011, pp. 465–472.

[22] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[23] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[24] V. Kumar, D. Hoeller, B. Sundaralingam, J. Tremblay, and S. Birchfield, "Joint space control via deep reinforcement learning," *International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[25] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[26] M. Lutter, L. Hasenclever, A. Byravan, G. Dulac-Arnold, P. Trochim, N. Heess, J. Merel, and Y. Tassa, "Learning dynamics models for model predictive agents," *arXiv preprint arXiv:2109.14311*, 2021.

[27] M. Lutter, J. Silberbauer, J. Watson, and J. Peters, "A differentiable newton-euler algorithm for real-world robotics," 2021.

[28] S. East, M. Gallieri, J. Masci, J. Koutník, and M. Cannon, "Infinite-horizon differentiable model predictive control," *arXiv preprint arXiv:2001.02244*, 2020.

[29] J. C. G. Higuera, D. Meger, and G. Dudek, "Synthesizing neural network controllers with probabilistic model-based reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2538–2544.

[30] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1714–1721.

[31] P. D'Oro and W. Jaśkowski, "How to learn a useful critic? model-based action-gradient-estimator policy optimization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 313–324, 2020.

[32] K. M. Jatavallabhula, M. Macklin, F. Golemo, V. Voleti, L. Petrini, M. Weiss, B. Considine, J. Parent-Levesque, K. Xie, K. Erleben, L. Paull, F. Shkurti, D. Nowrouzezahrai, and S. Fidler, "gradsim: Differentiable simulation for system identification and visuomotor control," 2021.

[33] P. Gradu, J. Hallman, D. Suo, A. Yu, N. Agarwal, U. Ghai, K. Singh, C. Zhang, A. Majumdar, and E. Hazan, "Deluca - A differentiable control library: Environments, methods, and benchmarking," *CoRR*, vol. abs/2102.09968, 2021.

[34] Y. Hu, J. Liu, A. Spielberg, J. B. Tenenbaum, W. T. Freeman, J. Wu, D. Rus, and W. Matusik, "Chainqueen: A real-time differentiable physical simulator for soft robotics," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6265–6271.

[35] D. Millard, E. Heiden, S. Agrawal, and G. S. Sukhatme, "Automatic differentiation and continuous sensitivity analysis of rigid body dynamics," *arXiv preprint arXiv:2001.08539*, 2020.

[36] C. M. Bishop, "Mixture density networks," Tech. Rep., 1994.

[37] K. Robotics, "Kinova ultra lightweight robotic arm 7 dof spherical," pp. 1–2, 2018.

[38] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei, "Deep kinematic pose regression," *CoRR*, vol. abs/1609.05317, 2016. [Online]. Available: http://arxiv.org/abs/1609.05317

[39] D. Pavllo, D. Grangier, and M. Auli, "Quaternet: A quaternion-based recurrent model for human motion," in *British Machine Vision Conference (BMVC)*, 2018.

[40] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.

[41] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine, "Time-contrastive networks: Self-supervised learning from video," *Proceedings of International Conference in Robotics and Automation (ICRA)*, 2018.

[42] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," 2017.

[43] Y. Tassa, S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, and N. Heess, “dmcontrol: Software and tasks for continuous control,” 2020.

[44] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[45] E. Todorov, T. Erez, and Y. Tassa.

[46] M. Mozifian, A. Zhang, J. Pineau, and D. Meger, “Intervention design for effective sim2real transfer,” 2020.