References

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

D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Y. Gal, R. McAllister, and C. E. Rasmussen. Improving pilco with bayesian neural network dynamics models. In *Data-Efficient Machine Learning workshop, ICML*, 2016.

F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pages 8583–8592. PMLR, 2020.

T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34: 28954–28967, 2021.

D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

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

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

M. Henaff, A. Canziani, and Y. LeCun. Model-predictive policy learning with uncertainty regularization for driving in dense traffic. *arXiv preprint arXiv:1901.02705*, 2019.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018a.

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*, 2018b.

J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*, 2019.

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

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://arxiv.org/abs/1801.01290.

H. Zhu, J. Yu, A. Gupta, D. Shah, K. Hartikainen, A. Singh, V. Kumar, and S. Levine. The ingredients of real-world robotic reinforcement learning, 2020.

J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47), oct 2020. doi:10.1126/scirobotics.abc5986. URL https://doi.org/10.1126%2Fscirobotics.abc5986.

J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning, 2021.

A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel. Adversarial motion priors make good substitutes for complex reward functions, 2022.

T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62), jan 2022. doi:10.1126/scirobotics.abk2822.

N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning, 2021.

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)*, pages 1–8, May 2018. doi:10.1109/ICRA.2018.8460528.

L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world, 2021.

D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation, 2018.

D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale, 2021.

F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets, 2021.

S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning, 2019.

S. James and A. J. Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation, 2021.

S. James, K. Wada, T. Laidlow, and A. J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation, 2021.

M. P. Deisenroth, G. Neumann, and J. Peters. 2013.

A. Nagabandi, K. Konoglie, S. Levine, and V. Kumar. Deep dynamics models for learning dexterous manipulation, 2019.

Y. Yang, K. Caluwaerts, A. Iscen, T. Zhang, J. Tan, and V. Sindhwani. Data efficient reinforcement learning for legged robots, 2019.

C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2786–2793. IEEE, 2017.

C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pages 64–72, 2016.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. ISSN 0893-6080.

T.-Y. Yang, T. Zhang, L. Luu, S. Ha, J. Tan, and W. Yu. Safe reinforcement learning for legged locomotion, 2022. URL https://arxiv.org/abs/2203.02638.

L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours, 2015.

H. Ha and S. Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. 2021.

OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. Learning dexterous in-hand manipulation, 2018.

E. Tzeng, C. Devin, J. Hoffman, C. Finn, P. Abbeel, S. Levine, K. Saenko, and T. Darrell. Adapting deep visuomotor representations with weak pairwise constraints, 2015.

A Appendix

A.1 Imagined Trajectories

To introspect the policy, we can roll out trajectories in Dreamer latent space, then decode the images to visualize the models intent. Figures A.1 and A.2 show examples of imagined rollouts for the UR and XArm. Each row is an imagined trajectory, showing every other frame for clarity.

Figure A.1: **UR5 Imagined Rollouts:** Latent rollouts on the UR5 environement. Multiple objects introduce more visual complexity that the network has to model. Note the second trajectory, which shows a static orange ball becoming a green ball.

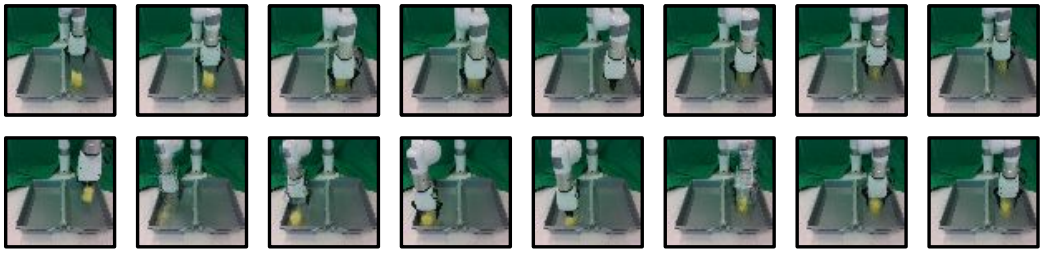

Figure A.2: **XArm Imagined Rollouts:** Latent rollouts on the XArm environement.

A.2 Extended Baselines for Pick and Place

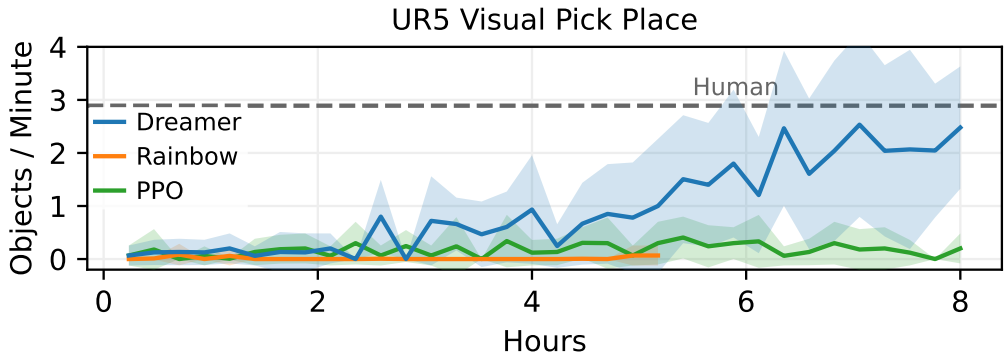

Figure A.3: **UR5 Multi Object Pick and Place**: The learning curves of Dreamer and two baselines (Rainbow DQN, and PPO) are shown here. Both methods are outperformed by Dreamer, with both baselines making little progress on the task.

In focused RL research settings, often a specific observation and action modality are assumed. In practical robotics settings however, its advantageous to take advantage of all sensory modalities available, making it difficult to find an effective baseline for the robot manipulator pick and place experiments. Again, the UR5 multi-object visual pick and place uses image observations and proprioceptive observations. The action space is discretized to include delta controls for each

Cartesian direction and an gripper toggle. In addition, we constrain the z-axis to the table when the gripper is empty and open the gripper when the robot is above the alternate bin. Most state of the art RL algorithms make assumptions about the observation space (state only or image only), or the action space (continuous only or discrete only), making it difficult to directly use such an algorithm off the shelf. Our modified Raindow DQN, described and shown in Section 3 is one attempt at this.

Another widely used and well performing generic RL algorithm is proximal policy optimization (PPO) introduced by Schulman et al. (2017). Here we attempt to use a modified categorical PPO for the robot pick and place experiments and report the results here as an additional baseline. To account for proprioceptive readings, we concatenate them in the channel dimension. The results for the UR5 are seen in Figure A.3. We see that PPO is comparable to Rainbow DQN and is unable to learn. We suspect this is due to the difficulty of the task due to sparse rewards.

A.3 Model Adaptation

One challenge faced in the real world robotics experiments is changing environmental conditions, such as lighting changes, as well as changes to the robot dynamics, such as parts getting worn down over time. The experiments performed in this work faced similar challenges. Unexpectedly, we found that that Dreamer is able to adapt the agent effectively to its current environmental conditions, with no change to the learning algorithm, showing promise for using world models in continual learning settings (Parisi et al., 2019). We report our observations here of these adaptations here.

A1 Quadruped: Adaptation to external perturbations. Due to the reset free nature of the A1, after converging to a walking policy the agent never has to flip over from its back again. When put on its back, the A1 struggles again before resetting its body to a upright position. We manually perturb the robot with external forces causing the A1 to flip over. As the A1 is initially trained without these manual perturbations, the agent struggles to recover. However after approximately 10 minuets of training with external perturbations, the A1 is able to learn to adapt, and quickly recover from perturbations. We refer to the provided videos which illustrate this behavior. This qualitative behaviour is promising for the rapid adaptation of RL agents.

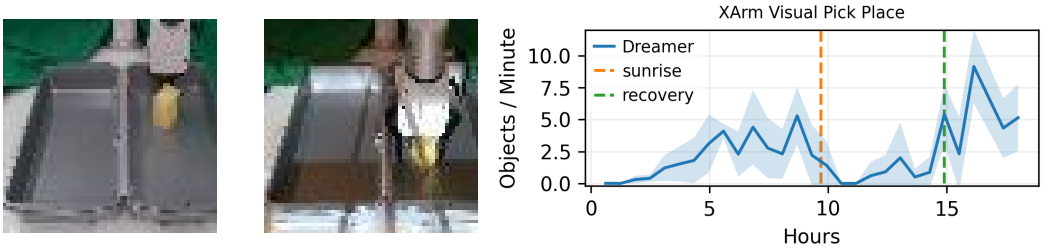

Figure A.4: The left two images are raw observations consumed by Dreamer. The left most image is an observation from as seen by the robot at night, when it was trained. The next image shows an observation during sunrise. Despite the vast difference in pixel space, the agent is able to recover performance in 5 hours and even reach better performance than before (up to 7 object per minute).

XArm: Adaptation to changing lighting conditions. Our robotics lab, which the experimental setup for the XArm was located, has direct access to large windows. This meant that the lighting in the room would change throughout the day. Our XArm experiments were intended to run after sundown to keeping the lighting conditions constant. Figure A.4 shows the learning curve of the XArm. As expected the performance of the XArm drops during sunrise. However, we see that the XArm is able to adapt to the change in lighting conditions and recover the original performance faster than training from scratch. A careful inspection of the image observations of the robot at these times, shown in Figure A.4, reveals that the robot received observations with strong light rays covering the scene in contrast to the original training observations.

A.4 Hyperparameters

Name	Symbol	Value
General		
Replay capacity (FIFO)	—	10^6
Start learning	—	10^4
Batch size	B	32
Batch length	T	32
MLP size	—	4×512
Activation	—	$\mathrm{LayerNorm} + \mathrm{ELU}$
World Model		
RSSM size	—	512
Number of latents	—	32
Classes per latent	—	32
KL balancing	—	0.8
Actor Critic		
Imagination horizon	H	15
Discount	γ	0.95
Return lambda	λ	0.95
Target update interval	—	100
All Optimizers		
Gradient clipping	—	100
Learning rate	—	10^{-4}
Adam epsilon	ϵ	10^{-6}

Table A.1: Hyperparameters.

A.5 Extended Related Works

Reinforcement learning on real world robot hardware is extremely challenging due to the large range of possible dynamic behavior and visual complexity. In addition to the complex algorithmic challenges already present in reinforcement learning algorithms, deploying these algorithms on real world hardware provides additional systems challenges (Zhu et al., 2020). We provide a brief overview of some common approaches to robot learning in in the real world.

RL for locomotion Reinforcement learning for locomotion is a well studied problem that requires training robots to reason about contact sequences and navigate their environment. A common approach to tackling the locomotion problem is to train RL agents in simulation with large amounts of simulated data under domain and dynamics randomization (Lee et al., 2020; Siekmann et al., 2021; Escontrela et al., 2022; Miki et al., 2022; Rudin et al., 2021; Peng et al., 2018), then deploying the learned policy in the real world. Learning locomotion policies directly in the real world poses additional challenges due to the sample inefficiency of common RL algorithms and the added risk of damaging expensive hardware. To overcome this issue Smith et al. (2021) explored pre-training policies in simulation and fine-tuning them with real world data. An alternative approach proposed by (Yang et al., 2022) trains locomotion policies in the real world but leverages a recovery controller trained in simulation to prevent the robot from entering unsafe states. In contrast, we do not utilize any simulator, and directly train our policies on hardware. (Yang et al., 2019) investigated learning a dynamics model using a multi-step loss and using model predictive control to accomplish a specified task.

RL for manipulation Learning promises a scalable avenue to enabling robot manipulators to learn and solve tasks in open real world environments. One class of methods attempts to scale model free

RL by collecting experience with a fleet of robots (Kalashnikov et al., 2018; 2021). This can vastly increase the amount of experience a trained agent can collect in the real world. In our experiments, we only leverage one robot, but parallelize an agents experience by using the learned world model. Another common approach is to leverage expert demonstrations (**??**) or other task priors (Pinto and Gupta, 2015; Ha and Song, 2021). James and Davison (2021); James et al. (2021) increases the sample efficiency of Q learning by utilizing a novel attention based mechanism to focus the learner on important aspects of the scene. This approach however, still relies on having a few human demonstrations to initialize the learner, which may not always be available. Other approaches, as in locomotion first utilize a simulator, then transfer to the real world (OpenAI et al., 2018; Tzeng et al., 2015).

Model-based reinforcement learning Model based RL, due to its higher sample efficiency over model free methods, is a common class of algorithms applied to learning on real world robots (Deisenroth et al., 2013). A model based method first learns a dynamics model, which can then be used to plan actions (Nagabandi et al., 2019; Hafner et al., 2018), or be used as a simulator to learn a model free policy as in dreamer (Hafner et al., 2019; 2020). One approach to tackle the high visual complexity of the world learns an action conditioned video prediction model (Finn and Levine, 2017; Ebert et al., 2018; Finn et al., 2016). One down side of this approach is the need to directly predict high dimensional observations, which can be computationally inefficient and easily drift. Dreamer learns a dynamics model in a latent space, allowing more efficient rollouts and avoids relying on high quality visual reconstructions for the policy.