Reinforcement Learning for Diverse Visuomotor Skills

Sunanda Dixit
Associate Professor
Department of Computer Science and Engineering
BMS institute of Technology and Management
Bangalore, India

A Vijaya Sai Mythili
Department of Compuer Science and Engineering
Bm Institute of Technology and Management
Bangalore, India

***Abstract*—This paper proposes a model-free deep reinforcement learning method that leverages a small amount of demonstration data to assist a reinforcement learning agent. By applying this approach to robotic manipulation tasks and train end-to-end visuomotor policies that map directly from RGB camera inputs to joint velocities. Demonstrating the approach can solve a wide variety of visuomotor tasks, for which engineering a scripted controller would be laborious. The reinforcement and imitation agent achieves significantly better performances than agents trained with reinforcement learning or imitation learning alone. The working principles and training by using reinforcement and imitation learning is discussed in this paper.**

Keywords—Reinforcement Learning;Imitation Learning; Manipulation tasks; Visuomotor Skills; Learning Agent.

I. INTRODUCTION

Recent advance in deep reinforcement learning (RL) have performed very well in several challenging domains such as video games and Go. For robotics, RL in combination with powerful function approximators such as neural networks provide a general framework for designing sophisticated controllers that would hard to handcraft otherwise. Reinforcement learning methods have a long history in robotics control but have typically been used with low-dimensional movement representations. The last few years have seen a growing number of successful demonstrations of deep RL for robotic manipulation using model-based and model-free techniques, both in simulation and on real hardware. Nevertheless, end-to-end learning of visuomotor controllers for long-horizon and multi-stage manipulation tasks using model-free RL techniques remains a challenging problem.

Developing RL agents for robotics requires overcoming several significant challenges. Policies for robotics must transform multi-modal and partial observations from noisy sensors, such as cameras, into coordinated activity of many degrees of freedom. At the same time, realistic tasks often come with contact-rich dynamics and vary along multiple dimensions (visual appearance, position, shapes, etc.), posing significant generalization challenges. Model-based methods can have difficulties handling such complex dynamics and large variations. Directly training model-free methods on real robotics hardware can be daunting due to the high sample complexity. The difficulty of real-world RL training is compounded by safety considerations as well as the difficulty of accessing information about the state of the environment (e.g. the position of an object) to define a reward function.

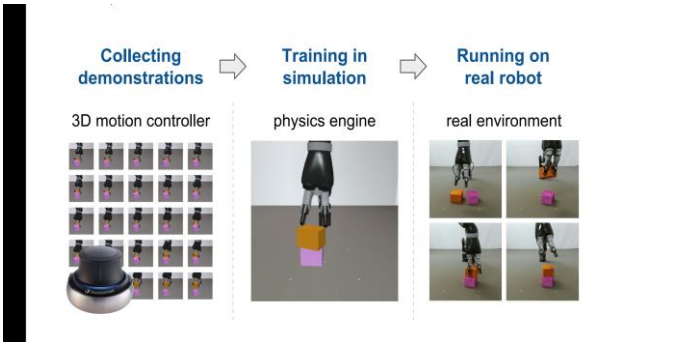

Fig 1: Principled robot learning pipeline. We used 3D motion controllers to collect human demonstrations of a task. Our reinforcement and imitation learning model leveraged these demonstrations to facilitate learning in a simulated physical engine. We then performed sim2real transfer to deploy the learned visuomotor policy to a real robot.

Finally, even in simulation when perfect state information and large amounts of training data are available, exploration can be a significant challenge, especially for on-policy methods [1]. This is partly due to the often high-dimensional and continuous action space, but also due to the difficulty of designing suitable reward functions.

A model-free deep RL method that can solve a variety of robotic manipulation tasks directly from pixel input. The main insights are 1) to reduce the difficulty of explanation in continuous domains by leveraging a handful of human demonstrations; 2) to leverage several new techniques that exploit privileged and task-specific information during training only which can accelerate and stabilize the learning of visuomotor policies in multi-stage tasks; and 3) to improve generalization by increasing the diversity of the training conditions. As a result, the policies work well under significant variations of system dynamics object appearances, task lengths, etc. Furthermore, we demonstrate promising preliminary results for two tasks, where the policies trained in simulation achieve zero-shot transfer to a real robot. Six manipulation tasks, including stacking, pouring, etc employed. The set of tasks includes multi-stage and long-horizon tasks, and they require full 9-DoF joint velocity control directly from pixels. The controllers need to be able to handle significant shape and appearance variations.

To address these challenges, combining imitation learning with reinforcement learning into a unified training framework. The approach utilizes demonstration data in two ways: first, it uses a hybrid reward that combines the task reward with an imitation reward based on Generative Adversarial Imitation Learning. This aids with exploration while still allowing the final controller to outperform the

human demonstrator on the task. Second, it uses demonstration trajectories to construct a curriculum of states along which to initialize the episodes during training. This enables the agent to learn about later stages of the task earlier in training, facilitating the solving of long tasks. It solves all six tasks, which neither the reinforcement learning nor imitation learning baseline can solve alone.

To sidestep the constrains of training on real hardware we embrace the sim2real paradigm which has recently shown promising results .Through the use of a physics engine and high-throughout RL algorithms. By simulating parallel copies of a robot arm to perform millions of complex physical interactions in a contact-rich environment, while eliminating the practical concerns of robot safety and system reset. Furthermore, we can, during training, exploit privileged and task –specific information about the true system state with several techniques, including learning policy and value in separate modalities, an object-centric GAIL discriminator, and auxiliary tasks for visual modules. These techniques stabilize and speed up policy learning, without imposing any constraints on the system at test time. Finally, we diversify training conditions such as visual appearance, object geometry, and system dynamics. This improves both generalizations with respect to different task conditions as well as transfer from simulation to reality.

As illustrated in Fig. 1 this instantiates a visuomotor learning pipeline going from collecting human demonstration to learning in simulation, and back to real-world deployment via sim2real policy transfer [1].

II. EASE OF USE

Reinforcement learning methods have been extensively used with low-dimensional policy representations such as movement primitives to solve a variety of control problems both in simulation and in reality. Three classes of RL algorithms are currently dominant for continuous control problems: guided policy search methods (GPS; Jonathan Ho and Stefano Ermon [2]), value-based methods such as the deterministic policy gradient (DPG) or the normalized advantage function (NAF) algorithm, and trust-region based policy gradient algorithms such as trust region policy optimization (TRPO) and proximal policy optimization (PPO) [3]. TRPO and PPO hold appeal due to their robustness to hyper parameter settings as well as their scalability but the lack of sample efficiency makes them unsuitable for training directly on robotics hardware.

GPS [2] has been used (e.g. Jonathan Ho and Stefano ermon et al. [2]) to learn visuomotor policies directly on areal robotics hardware after a network pre-training phase.

The idea of using large-scale data collection for training visuomotor controllers is to train a convolutional network to predict grasp success for diverse sets of objects using a large dataset with 10s or 100s of thousands of grasp attempts collected from multiple robots in a self-supervised setting.

Suitable cost functions and exploration strategies for control problems are challenging to design (e.g. Yuke Zhu, Ziyu Wang, Josh Merel et al. [1]) so demonstrations have long played an important role. Demonstrations can be used to initialize policies, design cost functions, guide exploration, augment the training data, or a combination of these. Cost functions can be derived from demonstrations either via tracking objectives or via inverse RL, or, as in our case, via adversarial learning. When expert policies are available, behavioral cloning can be used.

Most of these methods require observation and/or action spaces to be alignment between the robot and demonstrations. Sunanda Dixit et al [6-9] proposed different segmentation techniques. Machine learning algorithm explored in [10-11]

III. MODEL

The main goal is to learn a visuomotor policy with deep neural networks for robot manipulation tasks. The policy takes both an RGB camera observation and a proprioceptive feature vector that describes the joint positions and angular velocities. These two sensory modalities are also available on the real robot, allowing us to train in simulation and subsequently transfer the learned policy to the robot without modifications. Fig 2 provides an overview of the model. The deep visuomotor policy encodes the pixel observation with a convolutional network (CNN) and the proprioceptive feature with a multilayer perceptron (MLP). The features from these two modules are concatenated and passed to a recurrent long short term memory (LSTM) layer before producing the joint velocities (control commands). The whole network is trained end-to-end. We start with a brief review of the basis of generative adversarial imitation learning (GAIL) [2] and proximal policy optimization (PPO) [3].

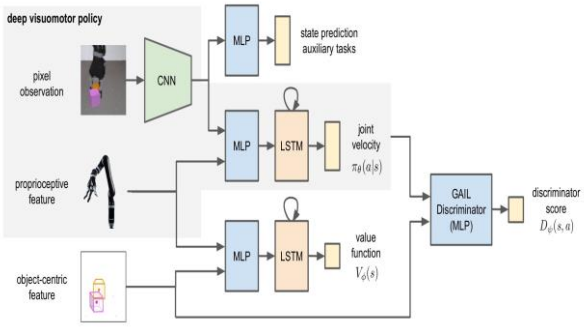

Fig 2: Model Overview. The core of the model is the deep visuomotor policy, which takes the camera observation and the propriceptive feature as input and produces the next joint velocities.

A. *Background: GAIL and PPO*

Imitation learning (IL) is the problem of learning a behavior policy by mimicking a set of demonstrations. Here assume that human demonstrations are provided as a dataset of state-action pairs D = {(si, ai)} i=1…….N. Some IL methods cast the problem as one of supervised learning, i.e., behavior cloning. These methods use maximum likelihood to train a parameterized policy πθ: S→A, where S is the state space and A is the action space. The behavior cloning approach works effectively when demonstrations are abundant. However, as robot demonstrations can be costly and time-consuming to collect, we aim for a method that can learn from a handful of demonstrations. GAIL uses demonstration data efficiently by allowing the agent to interact with the environment and learn from its own experiences. Similar to Generative Adversarial Networks (GANs), GAIL [2] employs two networks, a policy network

and a discriminator network. It uses a min-max objective function similar to that of GANs:

It uses a min-max objective function similar to that of GANs. This objective encourages the policy πθ to have an occupancy measure close to that of the expert policy.

The work trained with, train πθ with policy gradient methods to maximize the discounted sum of the reward function clipped at a max value of 10. In continuous domains, trust region methods greatly stabilize policy training. GAIL was originally presented in combination with TRPO for updating the policy. Recently, PPO has been proposed as a simple and scalable approximation to TRPO. PPO (e.g. John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov et al. [3]) only relies on first-order gradients and can be easily implemented with recurrent networks in a distributed setting. PPO implements an approximate trust region that limits the change in the policy per iteration. This is achieved via a regulation term based on the Kullback-Leibler (KL) divergence, the strength of which is adjusted dynamically depending on actual change in the policy in past iterations.

B. Reinforcement and Imitation Learning Model

1) Hybrid IL RL Reward: Shaping rewards are a popular means of facilitating exploration. Although reward shaping can be very effective it can also lead to suboptimal solutions. Hence, we design the task rewards as sparse piecewise constant functions based on the different stages of the respective tasks. For example, we define three stages for the block stacking task, including reaching, lifting, and stacking Reward change only occurs when the task transits from one stage to another. In practice, [4] we find defining such a sparse multi-stage reward easier than handcrafting a dense shaping reward and less prone to producing suboptimal behaviors. Training agents in continuous domains with sparse or piecewise constant rewards is challenging. Maximizing this hybrid reward can be interpreted as simultaneous reinforcement and imitation learning, where the imitation reward encourages the policy to generate trajectories closer to demonstration trajectories, and the task reward encourages the policy to achieve high returns on the task. Setting λ to either 0 or 1 reduces this method to the standard RL or GAIL setups. In our experiments, with a balanced contribution of these two rewards the agents can solve tasks that neither GAIL nor RL can solve alone. Further, the final agents achieve higher returns than the human demonstrations owing to the exposure to task rewards.

2) Leveraging physical states in stimulation: The physics simulator we employ for training exposes the full state of the system. Even though such privileged information is un-available on a real system, we can take advantage of it when training the policy in simulation. We propose four techniques for leveraging the physical states in simulation to stabilize and accelerate learning (1)the use of a curriculum derived from demonstration states, (2)the use of privileged information for the value function (3)the use of object-centric features in the discriminator, and (4)auxiliary tasks. We elaborate theses four techniques as follows:

1. Demonstration as a curriculum. The problem of exploration in continuous domains is exacerbated by the long duration of realistic tasks. Previous work indicates that shaping the distribution of start states towards states where the optimal policy tends to visit can greatly improve policy learning. We alter the start state distribution with the demonstration states. We build a curriculum that contains clusters of states in different stages of a task.

For instance, we define three clusters for the pouring task, including reaching the mug grasping the mug and pouring. During training with probability, we then start an episode from a random initial state, and with probability 1- we uniformly select a cluster and initialize the episode with a demonstration state from that cluster. This is possible since our simulated system is fully characterized by the physical states.

2. Learning value functions from states. PPO uses a learnable value function to estimate the advantage required to compute the policy gradient. During training, each PPO worker executes the policy for K steps and uses the discounted sum of rewards and the value as an advantage function estimator. As the policy gradient relies on the value function to reduce variance, it is beneficial to accelerate learning of the value function. Rather than using pixels as inputs similar to the policy network, we take advantage of the low-level physical states (e.g., the position and velocity of the 3D objects and the robot arm) to train the value with a smaller multilayer perception. We find that training the policy and value in two different modalities stabilizes training and reduces oscillation of the agent's performance.

3. Object-centric discriminator. As for the value function, we exploit the availability of the physical states for the GAIL [2] discriminator and provide task specific features as input. We find that object-centric representations (e.g., absolute and relative positions of the objects) provide the salient and relevant signals to the discriminator. The states of the robot arm in contrast lead the discriminator to focus on irrelevant aspects of the behavior of the controller and are detrimental for training of the policy. The construction of the object-centric representation requires a certain amount of domain knowledge of the tasks. We find that the relative positions of objects and displacements from the gripper to the objects usually provide the most informative characterization of a task. Empirically, we find that our model is not very sensitive to the particular choices of object-centric features, as long as they carry sufficient task-specific information.

4. State prediction auxiliary tasks. Auxiliary tasks have been shown to be effective in improving the learning efficiency and the final performance of deep RL methods. To facilitate learning visuomotor policies we add a state prediction layer on the top of the CNN module to predict the locations of objects from the camera observation[12] . We use a fully-connected layer to regress the 3D coordinates of objects in the task, minimizing the loss between the predicted and ground-truth object locations. The auxiliary tasks are not required for our model to learn good visuomotor policies; however, adding the additional supervision can often accelerate the training of the CNN module.

3) Sim2Real Policy Transfer: We perform policy transfer experiments on a real-world robot arm. The simulation was manually adjusted to roughly match the appearance and dynamics of the laboratory setup: a Kinect camera was

visually calibrated to match the position and orientation of the simulated camera, and the simulation's dynamics parameters were manually adjusted to match the dynamics of the real arm. Instead of using professional calibration equipment, our approach to sim2real transfer relies on domain randomization of camera position and orientation [5]. In addition, to improve robustness of our controllers to latency effects on the real robot, we also fine tune our policies while subjecting them to action dropping.

The following steps explains about the working

A. Environment Setup

Kinova Jaco arm that has 9 degrees of freedom: six arm joints and three actuated fingers. The robot arm interacts with a diverse set of objects on a tabletop. The visuomotor policy controls the robot by setting the joint velocity commands, producing 9-dimensional continuous velocities in the range of [-1, 1] at 20Hz. These propriceptive feature consist of the positions and angular velocities of the arm joints and the fingers. Visual observations of the table-top scene are provided via a suitably positioned real-time RGB camera [1]. The proprioceptive features and the camera observations are available in both simulation and real environments thus enabling policy transfer. We use a large variety of objects, ranging from basic geometric shapes to procedurally generated 3D objects built from ensembles of primitive shape. Increase the diversity of objects by randomizing various physical properties, including dimension, color, mass, friction, etc. We collect demonstrations using a 3D motion controller, which allows us to operate the robot arm with a position controller, and gather 30 episodes of demonstration for each task including observations, actions, and physical states [4]. As each episode takes less than a minute to complete, demonstrating each task can be done within half an hour.

B. Robot Arm Manipulation Tasks

Fig 3 shows the six manipulation tasks. The first column shows the six manipulation tasks in simulated environments, and the second column shows the real-world setup of the block-lifting and stacking tasks.

C. Robot Arm Manipulation Tasks

The first column shows the six tasks in simulated environments, and the second column shows the real-world setup of the block lifting and stacking tasks. We see obvious visual discrepancies of the same task in simulation and reality. These six tasks exhibit learning challenges to varying degrees. The first three tasks use simple colored blocks, which makes east to replicate a similar setup on the real robot. We study sim2real policy transfer with the block lifting and stacking tasks.

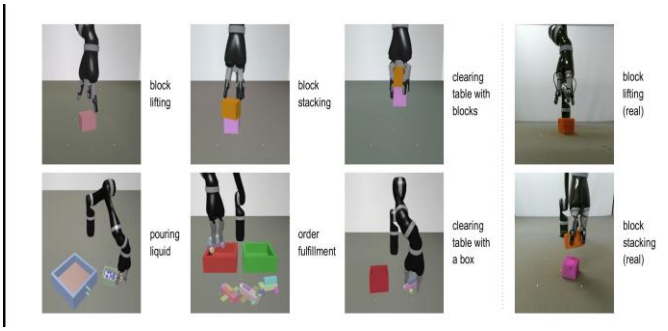

Fig 3: Visualizations of the six manipulation tasks in our experiments. The left column shows RGB images of all six tasks in the simulated environments. These images correspond to the actual pixel observations as input to the visuomotor policies. The right column shows the two tasks with color blocks on the real robot.

Block lifting. The goal is to grasp and lift a randomized block, allowing us to evaluate the model's robustness. We vary several random factors, including the robot arm dynamics, lighting conditions, camera poses, background colors, as well as the properties of the block. Each episode starts with a new configuration with these random factors uniformly drawn from a preset range.

Block stacking. The goal is to stack on top of the other block. Together with the block lifting task, this is evaluated in sim2real transfer experiments.

Clearing table with blocks. This task requires lifting two blocks off the tabletop. One solution is to stack the blocks and lift them both together. This task requires longer time and a more dexterous controller, introducing a significant challenge for exploration.

The next three tasks involve a large variety of procedurally generated 3D shapes, making them difficult to recreate in real environments [1]. To generalize across object variations in long and complex tasks.

Clearing table with a box. The goal is to clear the tabletop that has a box and a toy car. One strategy is to grasp the toy, put it into the box, and lift the box. Both the box and the toy car are randomly generated for each episode.

Pouring liquid. Modeling and reasoning about deformable objects and fluids is a long-standing challenge in the robotics community. We design a pouring task where we use many small spheres to simulate liquid. The goal is to pour the "liquid" from one mug to the other container. This task is particularly challenging due to the dexterity required. Even humans struggled to demonstrate the task with our 3D motion controller after extensive practice.

Order fulfillment. In this task we randomly place a variable number of procedurally generated toy planes and cars on the table. The goal is to place all the planes into the green box and

all the cars into the red box. This task requires the policy to generalize at an abstract level. It needs to recognize the object categories, perform successful grasps on diverse shapes, and handle tasks with variable lengths.

IV. CONCLUSION

Combining reinforcement and imitation learning considerably improves the ability to train systems capable of solving challenging dexterous manipulation tasks from pixels. The method describes all three stages of a pipeline for robot skill learning. Working principle of robot with steps in visuomotor skills is explained.

REFERENCES

[1] Yuke Zhu, Ziyu Wang, Josh Merel, "Reinforcement and Imitation learning for diverse visuomotor skills", Computer Science Department, Standard University, USA, DeepMind, London, UK, arXiv:1802.09564v2[cs.RO], 2018.

[2] Jonathan Ho and Stefano Ermon, "Generative adversarial imitation learning", In NIPS, pages 4565-4573, 2016.

[3] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms", arXivpreprint arXiv"1707.06347, 2017.

[4] Josh Merel, Yuval Tassa, Dhruva TB, Sriram Srinivasan, Jay Lemmon, Ziyu Wang, Greg Wayne, and Nicolas Heess, "Learning human behaviors from motion capture by adversarial imitation", arXiv:1707.02201, 2017.

[5] Stephen James, Andrew J.Davison and Edward Johns, "Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task", arXiv:1707.02267, 2017.

[6] Sunanda Dixit and Suresh Hosahalli Narayan, "Segmentation of Kannada Handwritten Text Line through Computation of Variance", Computer Science & Information Security, Volume 12, No. 2, ISSN: 1947-5500, pp. 56-60, 2014.

[7] Sunanda Dixit, Mahesh BV and Suma V, T1 and T2 MRI Brain Images Registration and Fusion TechniqueInternational Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-6, March 2020.

[8] Sunanda Dixit, S. Ranjitha and H.N.Suresh, "Segmentation of Handwritten Kannada Text Document through Computation of Standard Error and Weighted Bucket Algorithm", International Journal of Advanced Computer Technology, Volume 3, Number 2, ISSN:2319-7900, pp. 55-62, 2014,

[9] Sunanda Dixit and Dr. H.N. Suresh, "South Indian Tamil Language Handwritten Document Text Line Segmentation Technique With Aid of Sliding Window and Skewing Operations", Journal of Theoretical and Applied Information Technology, ISSN: 1992-8645, E-ISSN: 1817-3195, Volume 58, No.2, pp. 430-439, 2013.

[10] Sunanda Dixit, Mahesh B V and Suma V, T1 and T2 MRI Brain Images Registration and Fusion Technique, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-6, March 2020

[11] InduYekkala, Sunanda Dixit and M.A.Jabbar, "Prediction of Heart Disease Using Random Forest and Rough Set Based Feature Selection," International Journal of Big Data and Analytics in Healthcare (IJBDAH, IGI Global), Volume 3, issue 1, 2018.

[12] Parameshachari B D et. al "Big Data Analytics on Weather Data: Predictive Analysis Using Multi Node Cluster Architecture", International Journal of Computer Applications (0975 – 8887) proceedings of National Conference on Electronics, Signals and Communication – 2017, pp 12-17,2017