Learning Robotic Manipulation Tasks based on Incremental Demonstrations in a Virtual Environment

Giuseppe Rauso[1], Riccardo Caccavale[1] and Alberto Finzi[1]

[1]*Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università degli Studi di Napoli "Federico II"*

Abstract

The ability to grasp and manipulate objects is crucial for performing several complex tasks and it is highly desirable to transfer this skill effectively and naturally to robotic systems. Learning by demonstration provides a particularly interesting and promising technique to address this problem, as it allows us to leverage the guidance of the demonstrations provided by an expert to speed up task learning. In this work, we tackle the problem of learning manipulation tasks by demonstration in a virtual environment. Our aim is to develop an incremental, generalizable, and robust method for learning robotic manipulation tasks using a limited number of demonstrations, while assuming minimal information about the objects to be manipulated. The developed method combines imitation learning and reinforcement learning by proposing an incremental approach in which the operator first demonstrates specialized tasks to the robotic system, and subsequently more complex tasks, exploiting the skills learned during the previous phases. The experimental evaluation shows the feasibility and advantage of the proposed method in terms of modularity, low number of demonstrations, and reliability of the trained system.

Keywords

Learning from demonstration, robot manipulation, incremental learning.

1. Introduction

In this work, we address the problem of learning robotic manipulation tasks by training the system through demonstrations provided by a human operator in a virtual environment. In particular, an incremental approach to training is proposed, allowing the operator to first demonstrate specialized tasks in simple scenarios and then progressively train more complex and articulated tasks, leveraging the behaviors already trained in previous stages. The objective is to provide an incremental, generalizable, and robust method for learning robotic manipulation tasks using a limited number of demonstrations in a virtual environment, assuming restricted information about the objects to be manipulated. Another relevant aspect of the proposed approach relies on the use of virtual reality to provide demonstrations to the robotic system in a safe and simplified manner, in so avoiding the complexity of interacting with a real robotic system. The proposed method combines reinforcement learning techniques [1] with imitation learning methods [2] to achieve not only accelerated training through demonstration guidance, but also behavior influenced by the experts' preferences to leverage their contextual knowledge for object manipulation. Different approaches to learning from demonstration methods in a virtual environment can be found in the literature. Approaches based on learning from demonstration have been proposed to guide learning, reducing complexity and improving

10th Italian Workshop on Artificial Artificial Intelligence and Robotics (AIRO 2023)

giuseppe.rauso@unina.it (G. Rauso); riccardo.caccavale@unina.it (R. Caccavale); alberto.finzi@unina.it (A. Finzi)

efficiency [3, 4] in the presence of more or less complex end-effectors [5, 6]. In this context, experts' demonstrations are obtained in various manners: through robot teleoperation [7], video demonstrations [8], kinesthetic demonstrations [9, 10], or through motion capture [11, 12]. In this work, we focus on demonstrations provided interacting with a virtual manipulator endowed with a multi-fingered end-effector. A similar problem is addressed in [11], where the authors introduce a technique called *Demo Augmented Policy Gradient* (*DAPG*) that incorporates demonstrations recorded in virtual reality using a motion capture glove, while *Behavioral Cloning* (*BC*) [13, 14] is employed to support learning; however, here a different setup is employed and an incremental/modular approach is not deployed. An incremental method for training a robotic manipulator in a virtual environment is presented in [15], which proposes a combination of Reinforcement Learning and *Generative Adversarial Imitation Learning* (*GAIL*) [16]. On the other hand, this approach assumes a two-finger gripper that can only be opened and closed, therefore the grasp training is different from what is considered in the present study where a multi-fingered end-effector is deployed. Specifically, in this work we consider as a case study a simulated robotic setup composed of a *Barrett WAM Arm* manipulator equipped with a *BarrettHand* end-effector. As for the virtual environment, the virtual reality headset used for teleoperating the robots to obtain demonstrations is a Meta Quest 2. The simulated environments have been developed in *Unity 2021.3.16f1*, while the training was conducted using its *ML-Agents* [17] toolkit which seamlessly incorporates the algorithms used in this work and also allows for a combination of them.

2. Learning manipulation tasks by incremental demonstration

In the proposed method for learning robotic manipulation tasks, we exploit the expert's demonstrations provided in a virtual environment through an incremental approach. Specifically, we illustrate the approach at work considering two training phases (see Figure 1). In the first phase, a robotic hand is trained to grasp different types of objects from a proximal pose, without considering the movement of the robotic arm (*proximity grasping*). In the second phase, the hand trained during the first phase is exploited to train a more complex task, where the robotic arm is tasked to reach, grasp, and lift a specific object (*grasp-and-lift*). These two training phases are therefore sequential, as the result of the first phase is used for the second.

2.1. Proximity Grasping

The environment for this training phase is inspired by the one proposed by [18]. In this settings, the *BarrettHand* grasper is "suspended" and cannot move in space, i.e., it is not mounted on an arm or base (see Figure 2). At the beginning of the episode, objects are placed in front of the hand, and at the end of the episode, a gravity test is applied to verify the grip. In [18], objects and pre-grasp positions defined in [19] are used. For this work, however, the hand has a fixed 3D position and the objects are cubes, cylinders, and spheres, positioned in front of the agent (as illustrated in Figure 2) with random position, rotation, scale, and mass defined in reasonable ranges and suitable for the type of grip. A single episode has a maximum length of 1000 steps. Gravity for the objects is disabled for 800 steps from the start of the episode. After this interval, a stability test (or force test) is performed; specifically, similarly to [18], a force with an intensity

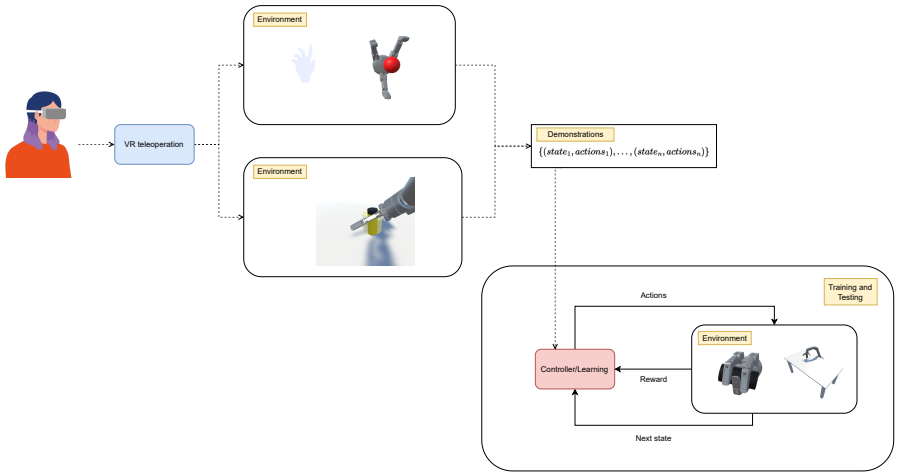

Figure 1: General pipeline of the training process

equal to gravity acting on the object is applied in four directions: upwards, downwards, to the right, and to the left. The episode ends with a positive reward if the distance between the hand and the object never exceeds a certain threshold. Therefore, if the grasp is stable, after the 200 test steps, the object will be at a distance less than the predetermined threshold, and the episode can be considered successfully completed.

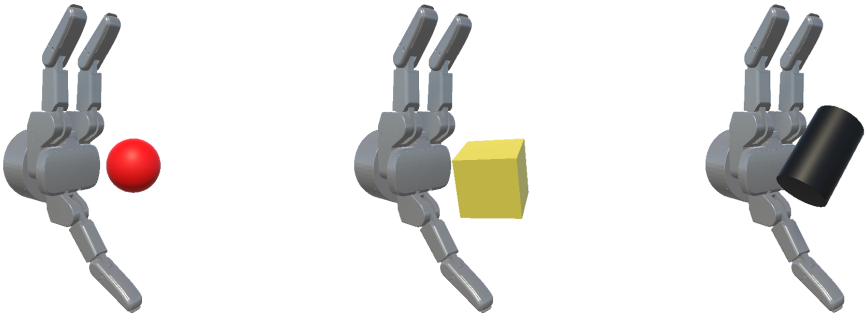

Figure 2: Environment for the grasping phase with BarrettHand using spheres, cubes, and cylinders

Observations. The agent observes the rotations on the individual rotation axes of the 8 joints of the BarrettHand in reduced space. Additionally, the agent observes a measure of force applied. The information regarding the object that the agent observes includes the bounding box (front-top-left and back-bottom-right points, and its rotation), velocity, angular velocity, and the object type. As hypothesized in [18, 20], information about contact with the object can improve performance and generalize to objects of different shapes and sizes. We therefore

define c_i as a variable with a value of 1 if the sensor detects a touch and 0 otherwise. These sensors are located only on the fingertips. The recorded value is passed as an observation to the agent.

Actions. In the proposed setting, the agent can rotate every joint of the robotic hand on a single axis. The action space is discrete and there are 3 possible actions for each of the 8 articulations: 0 to keep the rotation unchanged, 1 to increase the rotation, 2 to decrease the rotation. For each joint $i \in \{0, \dots, 7\}$, the target angle is increased or decreased starting from the current target rotation, which is the rotation that the articulation drive aims to reach applying a force or torque, with a fixed rotation speed. This means that if the agent decides to increase the target rotation of a joint i, the variation of the angle is fixed and the agent cannot choose the amount of rotation. The agent can only decide to change the target angle or leave it at the same value.

Demonstrations. To record demonstrations by directly teleoperating the robotic hand in simulation, a mapping has been proposed between Oculus hand tracking and the BarrettHand robotic hand (see Figure 3). In the Unity environment, the operator can directly control the robotic hand and show the grips for the objects used. Specifically, to record the increments or decrements of each joint of the robotic hand, the difference between the angle on the rotation axis of the human hand (tracked by the Oculus headset) and the target angle on the robotic hand is calculated. The sign of this difference will indicate whether to record an increase, a decrease, or no variation in the target angle.

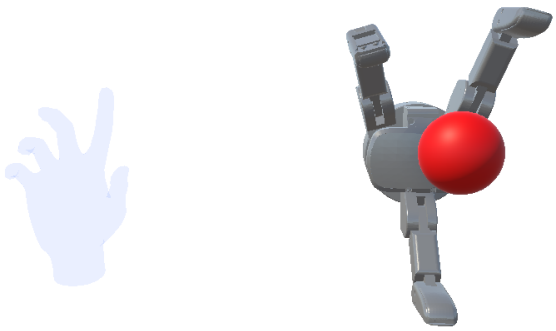

Figure 3: Mirroring between the expert's hand and the robotic hand to record the demonstrations.

Models and Training. A significant aspect of the proposed training method relies on a specific combination of imitation learning and reinforcement learning techniques. More precisely, we leverage *Proximal Policy Optimization* [21] to exploit the environment's reward, while simultaneously incorporating the GAIL reward as an intrinsic reward. Additionally, we perform weight updates of the policy through BC (Behavioral Cloning). This way, our approach capitalizes on the initial advantage offered by BC, allowing for the rapid acquisition of a policy that surpasses random initialization by mimicking the expert's actions. Subsequently, our goal

is to accumulate significant intrinsic rewards from GAIL in later stages to improve the imitation of the expert's behavior. As the training progresses, the influence of BC gradually diminishes until it converges to a weight of 0 at the conclusion of the training. The environmental reward in this context serves as a binary signal, indicating either the success or failure of each episode. For the training described in this section, we recorded only 15 episodes (approximately 3 minutes), specifically 5 episodes per object type. All episodes were successfully completed by the expert. The model used for the policy is an MLP composed of two fully-connected layers, each with 128 neurons, and a recurrent layer with LSTM with a hidden state dimensionality of 64 ($memory\_size = 128$, so $memory\_size/2$ because the initial memory will be divided between the hidden state and initial cell state and sequence length of 64. As observed in [22], the use of memory via LSTM can improve agent performance for object manipulation tasks. The overall training took about 8 hours on the machine used (i7-9700K and RTX 3070 Ti) for 7 million steps, but the agent reaches peak performance after approximately 1.3 million steps, which is just one hour of training.

2.2. Grasp-and-lift with manipulator

In this second stage of learning, our primary objective is to comprehensively train the robotic system's full range of motion. Specifically, our aim is to facilitate the acquisition of skillful hand movements and positioning strategies in close proximity to the target object, enabling successful grasping. In this context, the robotic hand is attached to the robotic arm, i.e., the *Barrett WAM Arm*, positioned deliberately in front of a table designated for object placement. For this particular experimental scenario, we have opted to exclusively use cylindrical objects. In this setting, two distinct grasping techniques have been defined, depending on the task: "top grasp" for grasping the cylinder from the top and "side grasp" for grasping it from the side The environment designed for this second phase, as illustrated in Figure 4, includes a table on which the cylinder is be placed and the robotic arm positioned in front of it. At the beginning of each episode, the object is placed on the table in a random position within a defined area, and a scale for the cylinder is randomly selected within preset intervals. The agent is granted 1000 steps to position the hand within the designated zone according to the task and activate the grasp Once the grasp is initiated, the agent becomes "locked," meaning that no further progress is made in the steps for the robotic arm, no decisions are made, and the outcome of the grasp is awaited. The hand is allotted 800 steps to adjust the finger positions, after which the same 200-step force test, utilized in the first phase, is activated, this time in the following global directions: right, left, forward, backward. Furthermore, the cylinder is subjected to gravitational force from the beginning of the episode and also during the test. In fact, to enhance the test's robustness, the hand (while holding the cylinder) is moved upward while the forces are applied. The episode concludes with a positive reward if the test succeeds, i.e., if the distance between the hand and the object is less than or equal to *max_dist* throughout or with a zero reward if one of the failure conditions is met.

Observations. To move the robotic hand, the agent controls the target position that the end-effector must reach by manipulating the arm joints through inverse kinematics. The agent thus observes spatial information related to the end-effector, including position, rotation and target

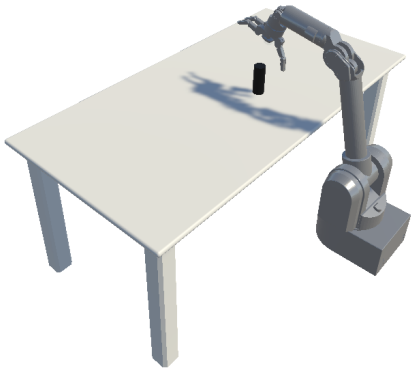

Figure 4: Environment for the grasp-and-lift phase

position, wrist rotation (yaw and pitch), hand velocity and hand angular velocity. Furthermore, the agent observes the type of task (top grasp or side grasp) and the same information about the object described in the previous section.

Actions. The displacement of the end-effector, and thus the hand, is achieved through inverse kinematics. The agent, therefore, controls the target position that the hand must reach. Additionally, the agent must rotate the wrist about two axes (yaw and pitch) and activate the grasp to complete tasks. The agent has the following branches of actions at its disposal: the displacement of the IK target along the x, y, and z axes, the pitch and yaw rotations of the wrist, and the activation of the grasp. Also in this case, the action space is discrete: actions for target displacement and yaw and pitch rotations can assume values of 0, 1, and 2 (no change, increase, decrease). The final action, to activate the grasp, can only take on values of 0 and 1, representing "grasp not activated" and "activate grasp," respectively.

Demonstrations. During the demonstration phase, we employ the controller of the Meta Quest 2 headset to manipulate the target position of the end-effector, activate the grasp using the controller, and adjust wrist rotation using the stick. Given that we operate within a discrete action space, during the demonstration, we cannot directly record the controller's position. Instead, we determine, for each coordinate, whether to increment, decrement, or maintain the value unchanged based on the controller's motion. Thus, we calculate the direction vector $d = \text{pos}_{\text{controller}} - \text{pos}_{\text{target}}$ between the controller's position and the target position, and the sign of each coordinate dictates whether to increase, decrease, or leave unchanged the corresponding coordinate of the target position. Similarly, for pitch and yaw rotations, the motion is discretized, and actions are recorded by indicating an increment, a decrement, or no change based on the analog stick's movement. In this case as well, the movement and rotation speeds are fixed.

Models and Training. Analogously to the first phase, the algorithm use for training is PPO in combination with BC and GAIL. In this case, we recorded 30 episodes for demonstrations (approximately 9 minutes in total), consisting of 15 episodes for top grasp and 15 episodes for side grasp. Once again, it is worth noting that all of these episodes were successfully completed by the expert. The neural network model does not utilize LSTM, instead, it has a structure with 3 fully-connected layers, each consisting of 512 neurons. A different machine was used for the training in the second phase (i5-9300H and GTX 1050 Ti). In this setting, the best training process took approximately 71 hours to complete 50 million steps.

3. Results and discussion

In this section, we discuss the performance of proposed method for each training phase.

3.1. Proximity Grasping

To evaluate proximity grasping, two tests were conducted. The first test focuses on the same objects used during the training phase. These objects, characterized by inherent variability and primitive shapes, can serve as an approximation of more complex objects or object parts. The second test is carried out using 3 objects from the dataset proposed in [19], i.e., a bottle, a donut, and a hammer. These objects were chosen for their shapes and the irregularities they exhibit, which can pose an interesting challenge for the trained model.

As for the first test, we compared the proposed combination of RL, BC, and GAIL with respect to other ways to combine these algorithms (i.e., RL, RL + BC, RL + GAIL, and RL + BC + GAIL with different parametrizations) showing that the proposed method (and setting) achieves a significantly higher success rate with respect to the alternatives. In the second test, the agent achieves a satisfactory success rate (about 75.3% ± 1.5) with the novel dataset. In particular, it attains a success rate of 66.5% ± 4 for bottles and 71.5% ± 6.9, including the ”awkward” positions, meaning with the neck of the bottle or the handle of the hammer facing the palm. However, performance improves (90% ± 2.2) when excluding these unnatural grasp positions for this type of object.

3.2. Grasp-and-lift with manipulator

For the second phase, in addition to the success rate, the percentage of successful activations of the grasp when it is *correctly* activated has also been calculated. The ratio is then calculated between the number of episodes completed correctly and the number of grasps activated correctly. This constitutes an additional quality test for the grasp policy obtained in the first part and for the hand placement near the object executed by the trained agent in the second part. Also in this case, the combination of RL+GAIL+BC proves to be the most effective among those tested, achieving an overall success rate of 84.5% ± 1.3 in the test phase. Specifically, it achieves an 81% ± 5.5 success rate for the vertical grasp and 87.3% ± 3.4 for the horizontal grasp. The success rate for correctly activated grasps is at 94.1% ± 0.8. During the experimentation, it was observed that the top grasp is the most complex. Even when executed successfully, the applied grasp may not be entirely ”correct” or natural.

4. Conclusion

In this work, an incremental and modular method for learning object manipulation through demonstrations in a virtual environment has been presented. Specifically, the proposed approach addresses the problem of grasping and manipulating objects. The focus was initially on the problem of proximity grasping before addressing the task of grasp-and-lift. The system has been designed to allow the operator to interact easily with a simulated robotic system, thereby avoiding the complexity of interacting with a real robotic system. The defined setup enables the recording of demonstrations in a simple manner, using cost-effective hardware. The experimental results illustrate the effectiveness of the proposed approach and how the two algorithms, BC and GAIL, manage to balance the disadvantages of the two imitation learning techniques and make the most of their potential when used together.

Acknowledgments

The research leading to these results has been partially supported by MELODY, grant P2022XALNS, PRIN 2022 PNRR, European Union - NextGenerationEU; Harmony, grant 101017008, European Union's Horizon 2020; Inverse, grant 101136067, and euROBIN, grant 101070596, European Union's Horizon Europe.

References

[1] R. Sutton, A. Barto, Reinforcement Learning, second edition: An Introduction, Adaptive Computation and Machine Learning series, MIT Press, 2018. URL: https://books.google.it/books?id=uWV0DwAAQBAJ.

[2] B. Zheng, S. Verma, J. Zhou, I. Tsang, F. Chen, Imitation learning: Progress, taxonomies and challenges, 2022. arXiv:2106.12177.

[3] P. Sharma, D. Pathak, A. Gupta, Third-person visual imitation learning via decoupled hierarchical controller, 2019. arXiv:1911.09676.

[4] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, Time-contrastive networks: Self-supervised learning from video, 2018. arXiv:1704.06888.

[5] A. Gupta, C. Eppner, S. Levine, P. Abbeel, Learning dexterous manipulation for a soft robotic hand from human demonstration, 2017. arXiv:1603.06348.

[6] D. Jain, A. Li, S. Singhal, A. Rajeswaran, V. Kumar, E. Todorov, Learning deep visuomotor policies for dexterous hand manipulation, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 3636–3643. doi:10.1109/ICRA.2019.8794033.

[7] A. Handa, K. V. Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, D. Fox, Dexpilot: Vision based teleoperation of dexterous robotic hand-arm system, 2019. arXiv:1910.03135.

[8] P. Sharma, L. Mohan, L. Pinto, A. Gupta, Multiple interactions made easy (mime): Large scale demonstrations data for imitation, 2018. arXiv:1810.07121.

[9] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe,

M. Riedmiller, Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards, 2018. arXiv:1707.08817.

[10] R. Caccavale, M. Saveriano, A. Finzi, D. Lee, Kinesthetic teaching and attentional supervision of structured tasks in human-robot interaction, Auton. Robots 43 (2019) 1291–1307.

[11] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, S. Levine, Learning complex dexterous manipulation with deep reinforcement learning and demonstrations, 2018. arXiv:1709.10087.

[12] R. Caccavale, M. Saveriano, G. A. Fontanelli, F. Ficuciello, D. Lee, A. Finzi, Imitation learning and attentional supervision of dual-arm structured tasks, in: 2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics, ICDL-EpiRob 2017, Lisbon, Portugal, September 18-21, 2017, IEEE, 2017, pp. 66–71.

[13] D. A. Pomerleau, Alvinn: An autonomous land vehicle in a neural network, in: NIPS, 1988. URL: https://api.semanticscholar.org/CorpusID:18420840.

[14] C. Sammut, S. Hurst, D. Kedzier, D. Michie, Learning to fly, in: Proceedings of the Ninth International Workshop on Machine Learning, ML '92, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992, p. 385–393.

[15] D. Kawakami, R. Ishikawa, M. Roxas, Y. Sato, T. Oishi, Learning 6dof grasping using reward-consistent demonstration, 2021. arXiv:2103.12321.

[16] J. Ho, S. Ermon, Generative adversarial imitation learning, 2016. arXiv:1606.03476.

[17] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, D. Lange, Unity: A general platform for intelligent agents, 2020. arXiv:1809.02627.

[18] H. Merzic, M. Bogdanovic, D. Kappler, L. Righetti, J. Bohg, Leveraging contact forces for learning to grasp, 2018. arXiv:1809.07004.

[19] D. Kappler, J. Bohg, S. Schaal, Leveraging big data for grasp planning, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 4304–4311. doi:10.1109/ICRA.2015.7139793.

[20] V. Kumar, T. Hermans, D. Fox, S. Birchfield, J. Tremblay, Contextual reinforcement learning of visuo-tactile multi-fingered grasping policies, 2019. arXiv:1911.09233.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, 2017. arXiv:1707.06347.

[22] 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, W. Zaremba, Learning dexterous in-hand manipulation, 2019. arXiv:1808.00177.