IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids), Ginowan, Okinawa, Japan, November 2022.

Accelerating Interactive Human-like Manipulation Learning with GPU-based Simulation and High-quality Demonstrations

Malte Mosbach*, Kara Moraw, and Sven Behnke

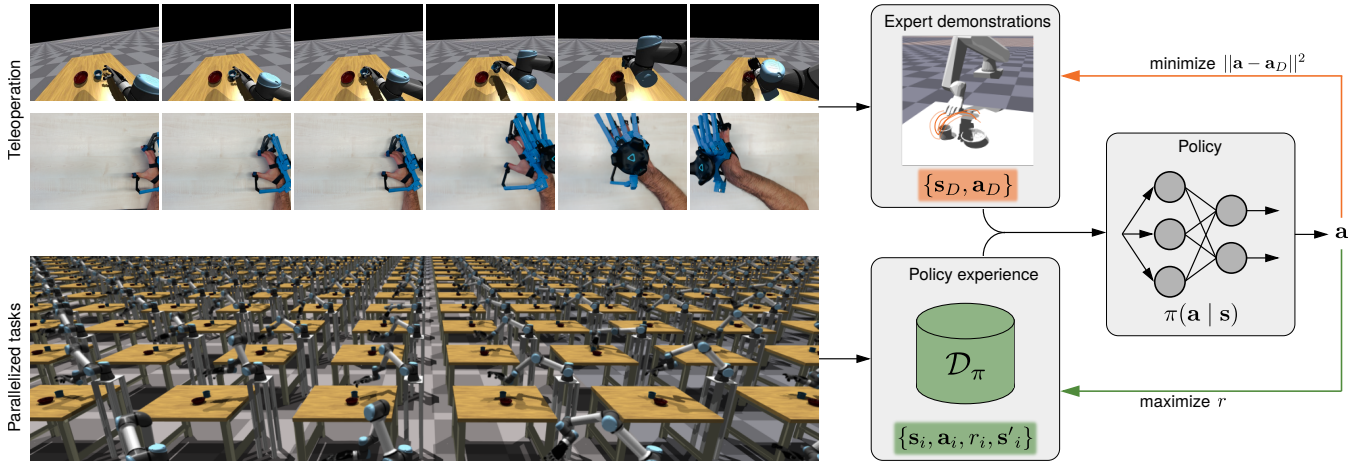


Fig. 1: We combine the strength of massively parallel reinforcement learning to produce proficient continuous control policies with the guidance of human demonstrations to solve challenging manipulation tasks. To this end, we introduce a suite of dexterous manipulation tasks together with a virtual reality teleoperation framework designed to enable human-like interactive manipulation in contact-rich environments.

***Abstract*— Dexterous manipulation with anthropomorphic robot hands remains a challenging problem in robotics because of the high-dimensional state and action spaces and complex contacts. Nevertheless, skillful closed-loop manipulation is required to enable humanoid robots to operate in unstructured real-world environments. Reinforcement learning (RL) has traditionally imposed enormous interaction data requirements for optimizing such complex control problems. We introduce a new framework that leverages recent advances in GPU-based simulation along with the strength of imitation learning in guiding policy search towards promising behaviors to make RL training feasible in these domains. To this end, we present an immersive virtual reality teleoperation interface designed for interactive human-like manipulation on contact rich tasks and a suite of manipulation environments inspired by tasks of daily living. Finally, we demonstrate the complementary strengths of massively parallel RL and imitation learning, yielding robust and natural behaviors. Videos of trained policies, our source code, and the collected demonstration datasets are available at `https://maltemosbach.github.io/interactive_human_like_manipulation/`.**

I. Introduction

Anthropomorphic robot hands provide a versatile interface to interact with a human-centric world. The ability to handle various objects and perform dexterous manipulation comes natural to humans but remains an important open problem in robotics. Prior work has tackled dexterous manipulation with multi-fingered robot hands through several approaches. Model-based trajectory optimization [15, 10] has demonstrated strong performance in simulation, but assumes access to accurate state information and dynamic models. These are difficult to obtain for contact-rich interaction tasks, which makes it challenging to apply these approaches to real-world scenarios and novel objects [20]. Further, imitation learning may be used, but it relies on high-quality demonstrations to succeed. While these are straightforward to obtain for drone flight or car driving [27], collecting proficient demonstrations for anthropomorphic robots requires more involved teleoperation setups. Moreover, imitation learning cannot improve upon the observed behaviors. Alternatively, RL has been used for robot grasping [6, 5, 25, 19] and object manipulation [1, 20], but struggles with the high-dimensional continuous state and action spaces posed by the tasks, leading to extremely high sample complexity. So far, manipulation robots remains far from reaching human-level dexterity.

To help closing the gap between robot and human manipulation abilities, we developed a novel framework for learning-based dexterous manipulation, *gym-grasp*, comprising various environments that represent tasks of daily living. We integrate two approaches to address the high sample requirements of RL methods on complex manipulation tasks.

*All authors are with the Autonomous Intelligent Systems (AIS) Group, Computer Science Institute VI, University of Bonn, Germany; mosbach@ais.uni-bonn.de

This work has been funded by the German Ministry of Education and Research (BMBF), grant no. 01IS21080, project “Learn2Grasp: Learning Human-like Interactive Grasping based on Visual and Haptic Feedback”.

First, our simulation leverages Isaac Gym [12], which enables massive parallelization of RL training, resulting in high simulation throughput and empowering the performance of on-policy algorithms. Second, we present a virtual reality (VR) teleoperation framework, designed to enable skillful manipulation in contact-rich environments. In addition to immersive visualization, physical feedback from the environment is a crucial modality humans rely on during interaction with objects [4, 21]. We therefore hypothesize that tactile perception also impacts the quality of demonstrations and integrate haptic feedback based on simulator contact forces using a force-feedback glove. While massively parallel RL has been used to synthesize robust policies for dexterous control, prior works rely on dense rewards to guide policy search towards the desired behaviors [1, 23]. We demonstrate that imitation learning can be used to make this learning paradigm feasible even for sparse-reward tasks.

The main goal of this work is to foster research in learning-based dexterous manipulation through GPU-accelerated simulation and high-quality demonstrations and evince the complementary strengths of both paradigms. In summary, we present the following contributions:

- *Dexterous Manipulation Benchmark.* We introduce a suite of dexterous manipulation tasks that supports high-performance reinforcement learning via GPU-based physics simulation [12].
- *Virtual Reality Teleoperation.* We present a system for immersive VR teleoperation and evaluate the effect of haptic feedback on user preference and task success.
- *Human Demonstration Datasets.* We provide human demonstration datasets for the considered tasks to enable imitation learning and offline RL for dexterous manipulation.
- *Combining RL with Demonstrations.* We show the potential benefit of augmenting massively parallel RL with demonstration data.

II. Related Work

Robotic grasping and manipulation has been actively studied for decades [26]. Many prior approaches attempt to predict grasp poses either via analytical or data-driven methods [8]. Analytical methods are primarily based on mechanics, such as force-closure. They usually assume access to the exact object geometry and friction coefficients at contact points [16, 14]. Data-driven methods rely on machine learning and some form of training data, but can sometimes dispense with the very high demands of analytical methods for perfect observability of the work space and known object models. Nevertheless, both variants are fundamentally concerned with grasp synthesis, which is the problem of finding a suitable configuration of the robot's end effector, as to grasp a target object. This is in stark contrast to the dexterous manipulation observed in humans, which continuously interleaves perception and action. Recent works utilize RL to achieve such dexterous control. Kalashnikov et al. [6] demonstrate that RL is able to synthesize a vision-based grasping policy that can pick up unseen objects with a parallel gripper, at the cost of hundreds of thousands of real-world grasp attempts. Further, Rajeswaran et al. [20] explore the use of demonstrations for learning dexterous manipulation policies, but do not consider tactile perception. Chen et al. [1] study RL for in-hand reorientation and demonstrate strong performance of parallelized learning in GPU-based simulations.

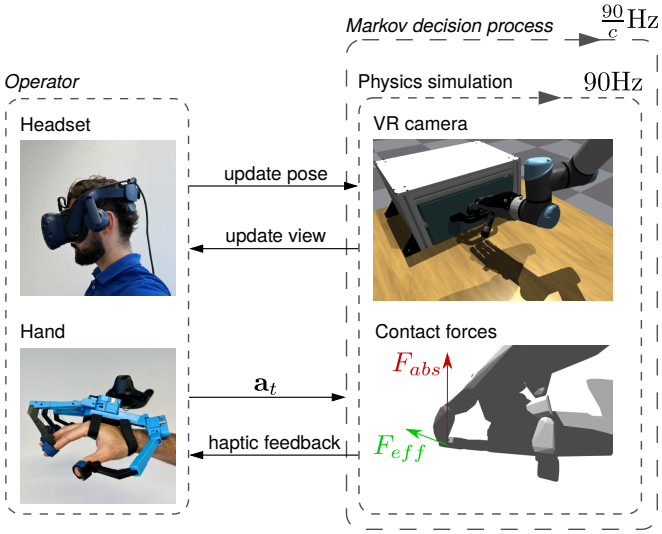


Fig. 2: Teleoperation framework overview.

Using teleoperation to collect demonstrations is an especially suitable paradigm for human-like robots due to the similar morphology and straightforward control mapping. Prior works have used vision-based teleoperation via motion capture data [9] or hand pose estimation [18, 17, 3]. While vision-based pose estimation has low hardware requirements, it limits the workspace and cannot provide feedback from the environment. Zhang et al. [27] demonstrate how consumer-grade VR controllers can be used to teleoperate a PR2 robot, but focus neither on anthropomorphic end-effectors, nor on haptic feedback. Kim et al. [7] explore force feedback for bilateral teleoperation, but operate with two-finger grippers. Commercially available systems, such as the Shadow Teleoperation System [22] provide haptic feedback and could also be used to collect demonstrations in contact-rich domains. However, this requires expensive, specialized hardware. Our system, on the other hand, enables haptic feedback during teleoperation of an anthropomorphic hand based on inexpensive VR-components and can be used with different robotic hands.

Concurrently to our work, Chen et al. [2] introduced a benchmark for bimanual hand manipulation based on Isaac Gym. Unlike our work, their focus is on learning to coordinate two hands and they do not consider learning from human demonstrations.

III. Virtual Reality Teleoperation

In the following, we introduce our VR teleoperation system and explain how it facilitates the collection of proficient demonstrations on contact-rich manipulation tasks.

A. System Overview

Our operator interfaces combines the Vive VR system and the SenseGlove DK1 force-feedback glove. Fig. 2 depicts the interaction of the main components. The operator observes the simulated scene through a dedicated camera in Isaac Gym. We update the camera pose instantaneously to follow the operator's head movements tracked by the headset. For each finger, the SenseGlove detects 4 DoF finger joint positions and features separately controllable force and vibration feedback. We mount a Vive tracker on top of the glove, providing sub-millimeter 6 DoF pose information at 90 Hz. Fusing the information from both devices enables intuitive control of the anthropomorphic robot hand. We employ a clutch-like mechanism, where the user can move their hand freely and only starts teleoperating the robot hand when indicated via the keyboard. The pose of the operator and robot hand then remain locked until the termination of the episode.

While an update frequency of 90 Hz is recommended for immersive VR operation, we may want to select actions at a lower rate since RL and imitation learning become increasingly challenging for long horizon tasks. We therefore decouple the update frequency of the head-mounted display and force feedback from the control frequency of the Markov decision process (cf. Fig. 2). Accordingly, we divide the simulation side into the actual physics simulation, which runs at 90 steps per second to provide fast updates for the headset and haptic feedback, and the MDP, which queries actions at $\frac{90}{c}$ Hz, where c is the control frequency interval. We found a control frequency of 30 Hz to be sufficient for accurate control by teleoperation and effective learning via RL.

B. Haptic Feedback

As touch is of crucial importance to humans when performing dexterous manipulations, we incorporate this modality via force and vibration feedback. To allow the user to feel the presence of objects, we determine the rigid-body contact forces of the fingertips and decompose them into the absolute force and a directional component, which acts through the fingertip and prevents the hand from closing. The component of the force acting against closing a finger, F_{eff}, is mapped to the SenseGlove's force feedback command, which activates a braking system and increases the resistance to closing the finger further. This mimics the resistance felt when holding an object and is beneficial for grasping and transporting objects. The absolute force magnitude values are smoothed using a simple moving average low-pass filter. We then map the high-frequency components of the absolute force $||F_{abs}|| - \mathrm{MA}(||F_{abs}||)$, where MA refers to the moving average, to the vibrational feedback. This results in feedback responses for sudden increases in the contact force, such as collisions, but no feedback for sustained contact.

IV. Learning Dexterous Manipulation

A. Task Design

All implemented tasks feature a robotic arm and hand that we have access to for real-world experiments (in future work). Specifically, a UR5 arm and Schunk SIH hand are used. The actuation of the hand features 5 degrees-of-freedom (DoF). Tendons are used to control the rotating of the thumb and bending of the fingers, resulting in the coupled control scheme depicted in the bottom right of Fig. 3, which we replicate through the coupling of position targets in Isaac Gym. The actuation scheme is shown in Fig. 3. Actions are 11-dimensional and are interpreted as the relative change to the desired wrist pose (6 dimensions) and finger positions (5 DoFs of the Schunk SIH hand). We compute the UR5 joint position targets from the desired wrist pose using the Jacobian transpose method.

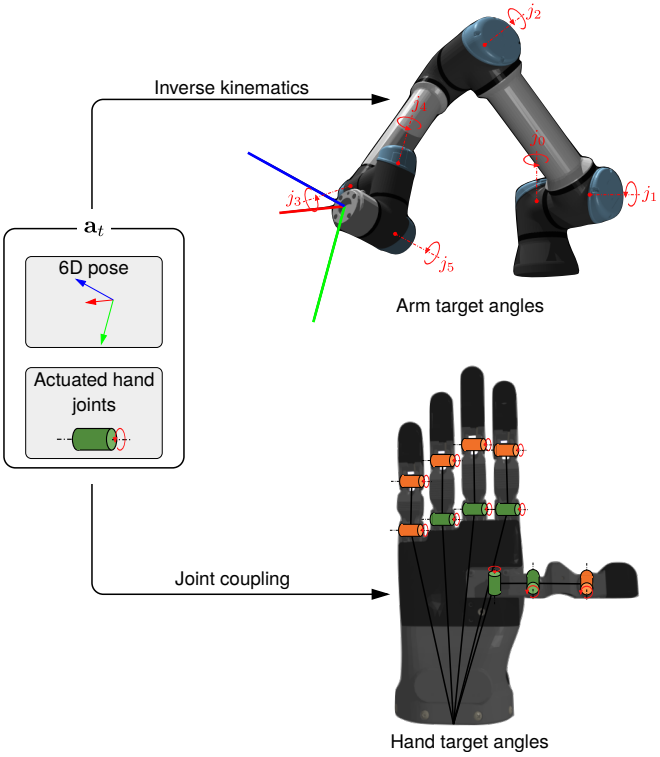
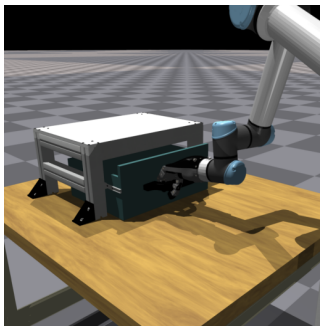
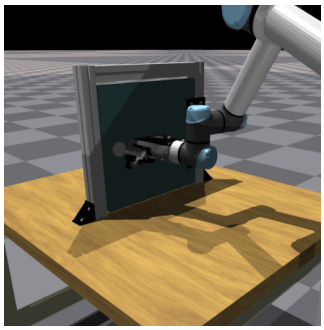


Fig. 3: Actuation of robot arm and hand. We actuate the robot hand via the 6D target pose of the wrist and the target joint angles of the fingers. Only the joints of the hand marked in green can be actuated directly. The orange joints are underactuated and positioned via equality constraints.

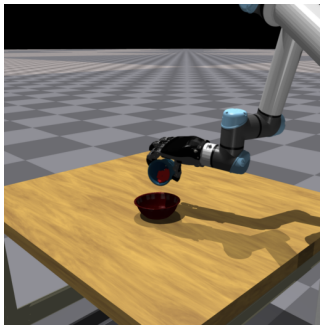
The selected manipulation tasks are shown in 4. All tasks provide proprioceptive observations of wrist pose and finger positions of the robot. On the *OpenDrawer* task, the agent is informed about how far the drawer is opened, from which the handle position can be inferred. Dense rewards penalize the distance of the robot hand from the handle, while continuously rewarding the opening of the drawer. The task is solved once the drawer is pulled open by 0.2m, which causes the environment to terminate. The *OpenDoor* task provides the pose of the door handle and is completed successfully once the door is opened by 45°. When dense rewards are used, the agent is rewarded continuously for how far the door is opened and the distance of the robot hand to the door handle is penalized. On the *PourCup* task,



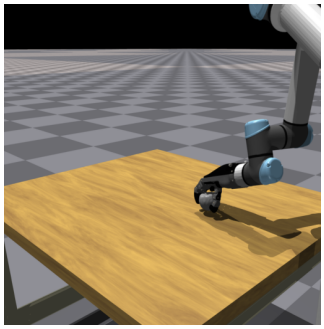
(a) OpenDrawer



(b) OpenDoor



(c) PourCup



(d) LiftObject

Fig. 4: Dexterous manipulation tasks.

we provide the pose of the full cup. The agent is rewarded proportional to the amount of particles poured into the bowl, while the distance of the hand to the cup is penalized. The *LiftObject* task exposes the pose of the object. When using dense rewards, the agent is rewarded for positive changes to the object height, while the distance of the robot hand from the object is penalized. The task is completed successfully and terminates once the object is lifted 0.2m above the table. In addition to the dense reward functions described above, we also investigate sparse rewards because we want to show how imitation learning can be used to bridge the performance gap between the two. Sparse rewards are 0 if a task is completed successfully and -1 otherwise for all tasks.

B. Massively Parallel Reinforcement Learning

While GPU-based parallelization has been widely adopted for neural network training, data collection in reinforcement learning is still mostly performed via single CPU simulations. Parallelizing data collection by running many instances of a robot in a single simulation has the potential to accelerate RL training by several orders of magnitude [12, 23]. We leverage the high-performance capabilities of Isaac Gym to scale up our training on dexterous manipulation tasks. To show the efficiency of our framework, we evaluate the simulation throughput during RL training on a single machine with an AMD Ryzen 9 5950X CPU and NVIDIA RTX A6000 GPU. All tasks run 16,384 parallel environments. Note that the number of simulated steps is three times higher than the MDP steps given in Table I because we maintain the control frequency interval of three to obtain consistent results for imitation and reinforcement learning.

TABLE I: Simulation throughput (MDP steps / s).

OpenDrawer	OpenDoor	PourCup	LiftObject
$58,887 \pm 1,726$	$56,513 \pm 1,442$	$21,977 \pm 507$	$58,980 \pm 1,224$

While the massive parallelization of agent instances in a single simulation leads to rapid training, it also presents unique challenges. Take the LiftObject task (see Fig. 4d), for example. Since objects of arbitrary geometry may be used, we drop them onto the table from a random seed pose to find feasible initializations before starting the actual task. This procedure cannot simply be repeated when an environment terminates, since a step in the physics simulation to drop the object would also advance the simulation in all other environments. Instead, we only drop the objects at random positions initially and then reset each environment to the exact pose found at the beginning. The randomization of object positions is therefore handled by the large number of environment instances instead of the separate reset phase.

C. Imitation and Reinforcement Learning

We aim to demonstrate the complementary strengths of imitation learning and massively parallel RL for dexterous manipulation. RL models the agent-environment interaction as a Markov decision process (MDP). An MDP is defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, T, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the sets of states and actions, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}_+$ is a state-transition probability function, which represents the probability of transitioning to the next state $\mathbf{s}_{t+1} \in \mathcal{S}$ given the current state $\mathbf{s}_t \in \mathcal{S}$ and action $\mathbf{a}_t \in \mathcal{A}$, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function, and $0 \leq \gamma \leq 1$ is a discount factor. The state-action marginal of the trajectory distribution induced by a policy $\pi(\mathbf{a}_t|\mathbf{s}_t)$ is denoted as $\rho_\pi(\mathbf{s}_t, \mathbf{a}_t)$. The objective of the RL problem is to maximize the expected sum of rewards discounted over time $J = \sum_t \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi}[\gamma^t R(\mathbf{s}_t, \mathbf{a}_t)]$.

We employ proximal policy optimization (PPO) [24], which is a state-of-the-art on-policy RL algorithm and base our implementation on [11]. For imitation learning, we use simple behavior cloning (BC) and leverage the implementation provided by Robomimic et al. [13]. Hence, we collect demonstration data as state-action tuples $(\mathbf{s}_i, \mathbf{a}_i)$ and train the policy to reproduce the expert actions for the same states. Finally, we combine reinforcement and imitation learning by incorporating the losses of both paradigms to form a demo augmented policy gradient (DAPG) algorithm [20]. In contrast to the original variant, which employs a natural policy gradient, we stick with the now widely used PPO algorithm. The trade-off between both losses is weighed as $\mathcal{L}_{DAPG} = \mathcal{L}_{PPO} + \lambda_0 \lambda_1^k \mathcal{L}_{BC}$, where k is the current training epoch. We set $\lambda_0 = 50$ and $\lambda_1 = 0.99$ in all experiments.

V. Experiments

We investigate the effectiveness of learning from demonstrations collected by our teleoperation framework, as well as the potential of utilizing demonstration data in massively parallel on-policy RL. Therefore, we aim to address the following questions:

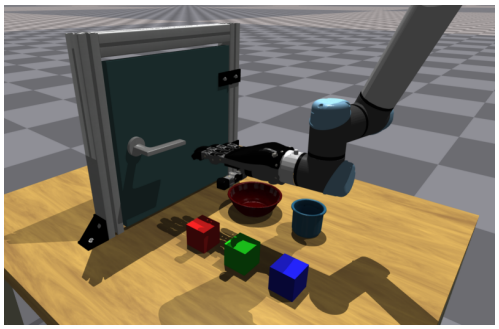


Fig. 5: User study environment.

1) Can the demonstrations collected with our VR teleoperation system be used to train successful policies for dexterous manipulation tasks?
2) How does learning from scratch in the massively parallel regime perform on the selected tasks?
3) How does incorporating demonstration data into the learning process impact the performance of on-policy RL?

A. Experimental Setup

To answer the proposed questions, we selected a set of challenging manipulation tasks that mimic real-world problems that a humanoid robot might face (see Fig. 4). The difficulty of these tasks depends on various factors. OpenDrawer is the simplest task, as it requires only a single skill and can be solved without very precise control of the fingers. OpenDoor is more complex, requiring the concatenation of the two skills of turning the handle and then pulling the door open. PourCup requires delicate manipulation and has a high risk of failure, since there is no way to recover if the cup is spilled somewhere other than the target location. The liquid in a real cup is represented by spherical particles, since we are working with a rigid-body physics simulation. This increased number of simulated bodies and contacts accounts for the lower simulation throughput on this task compared to the other environments (see Table I). The LiftObject task requires precise positioning of the fingers to produce a stable grip, as well as generalization to random initial positions and orientations of the object to be lifted.

The neural network architecture is kept constant across all evaluated methods. We represent the policy by an MLP consisting of 3 hidden layers of 512, 256, 256 neurons. Each RL agent is trained for 1000 epochs, where each epoch consists of 32 steps in the 16,384 parallel environment instances, resulting in a total 524 million steps.

B. Haptic Teleoperation User Study

First, we conduct a small user study to evaluate our teleoperation system. We analyze whether the system can be used by operators without prior experience to solve manipulation tasks. Further, we measure the influence of haptic feedback on user preference and task completion. Six participants with no prior experience operating the framework took part in our study.

TABLE II: Success rates and completion times for the tasks performed in the user study.

Haptic feedback	Task	Success rate	Completion time [s]	
			Mean	StdDev
✓	Stack cubes	0.83	**35.17**	**9.98**
	Open door	**1.0**	**6.47**	**1.82**
	Pour cup	**1.0**	**13.86**	**3.46**
✗	Stack cubes	**1.0**	43.26	22.13
	Open door	**1.0**	9.37	5.26
	Pour cup	**1.0**	15.63	3.56

The participants were asked to perform three tasks in the environment shown in Figure 5. Specifically, the tasks were to stack three cubes, open a door, and pour particles from a cup into a bowl. A task was marked as a failure if a maximum time of 3 min was reached or a non-recoverable failure occurred, e.g. dropping a cube off the table or spilling the cup. Each participant was asked to perform the tasks with and without haptic feedback. The order of these two trials was randomized to mitigate the influence of learning during teleoperation.

Quantitative results of the user study in terms of success rate on the tasks and the time required are shown in Table II. During all trials, only a single failure occurred, where a cube fell off the table. The high success rates for diverse tasks show that the system is intuitive to use even without prior experience. Lower completion times were observed in all tasks when haptic feedback was enabled, highlighting the added value of integrating this modality into teleoperation. The task of stacking three cubes shows the highest standard deviation in completion time, since the tower can be knocked over during construction, requiring the user to start over.

After each trial, participants were asked to rate their experience using seven-level Likert items. The results, shown in Fig. 6, show that intuitive control of the robot arm and fingers was generally rated highly. Haptic feedback had a positive impact on whether users felt they were interacting directly with the objects. However, the most pronounced difference between the two modes of operation is evident in the ability to recognize moments of contact. Users reported that this feature of haptic feedback provided a more confident sense of the exact position of the hand relative to other objects.

C. Results and Analysis

In the following, we structure our analysis of the obtained results by the questions posed at the beginning of this section.

1) *Can the demonstrations collected with our VR teleoperation system be used to train successful policies for dexterous manipulation tasks?*

To assess this question, we collected datasets of 200 demonstrations for each of the investigated tasks, which are split into 90% training and 10% validation data. We then train simple behavior cloning policies to replicate the observed demonstrations and use the validation loss to check for overfitting. The success rates are shown in the first row

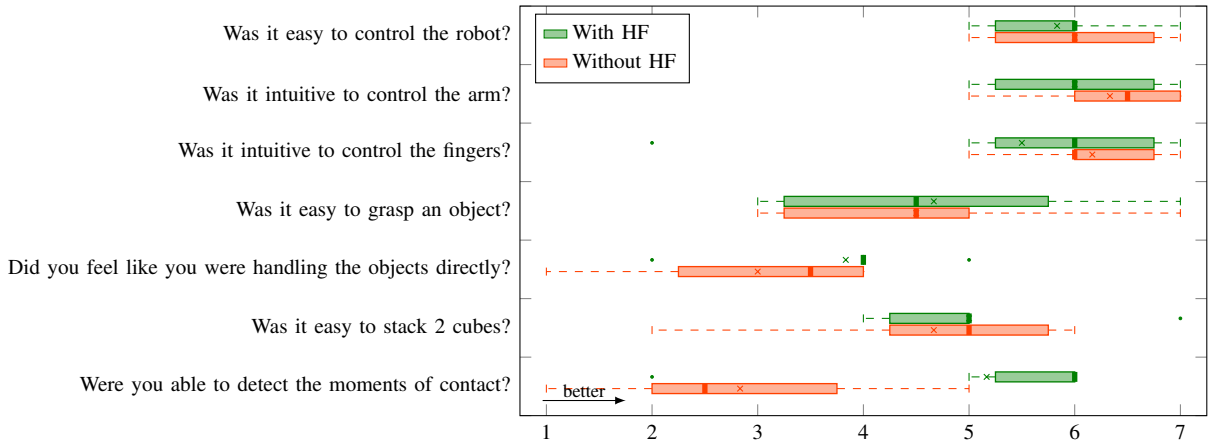


Fig. 6: Results of the user questionnaire. We depict the median, lower and upper quartile, lower and upper fence, outliers (•), and the average value (×).

of Table III. Near perfect results can be achieved using only imitation learning on the OpenDrawer and OpenDoor task. For PourCup, the results are still strong considering the complexity of the task and control space of the robot, but performance is not as consistent. Lastly, despite using only a single object geometry in this study, LiftObject is the most difficult to learn from demonstrations alone. This may be due to the fact that even for a human operator it is not straightforward to grasp an object securely, resulting in frequent regrasps or adjustments. Further, the generalization demanded by the random initialization of object poses appears to make imitation learning substantially harder. Overall, we were able to verify that the proposed pipeline can be used to collect demonstration for an anthropomorphic robot hand, but note that behavior cloning alone struggles to produce robust policies for more involved tasks.

2) *How does learning from scratch in the massively parallel regime perform on the selected tasks?*

We divide our analysis into the dense and sparse rewards setting. While it is generally more difficult to learn from sparse rewards, they are easy to specify. Providing dense rewards, on the other hand, requires tedious reward shaping and can bias learned behaviors towards unintended solutions. The strong results of PPO for dense rewards across all tasks studied suggest that massively parallel on-policy RL will find proficient solutions to tasks of the studied difficulty as long as a meaningful learning signal is present. In contrast, sparse rewards were only sufficient to get satisfactory solutions to the OpenDrawer task. While LiftObject and PourCup make some progress towards solving the respective problem, the results vary strongly between seeds. OpenDoor made no learning progress across all seeds. In summary, sparse rewards were sufficient for learning only in simpler tasks where the solution can be discovered repeatedly through random exploration. Tasks that require the combination of multiple behaviors in succession along with precise actions are very difficult to solve in this way, even with the vast amounts of policy experience used in this study.

3) *How does incorporating demonstration data into the learning process impact the performance of on-policy RL?*

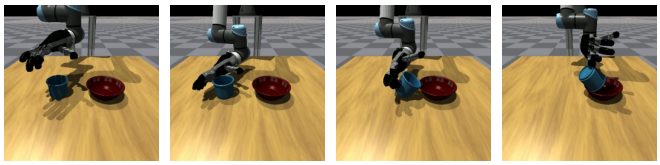
Finally, we analyze whether the integration of demonstrations into the learning process is sufficient to provide RL with the learning signal needed to generate robust policies. The perfect results for all tasks reported in the bottom of Table III confirm this hypothesis and highlight how the complementary strengths of massively parallel RL and imitation learning provide a powerful tool for solving challenging manipulation tasks. In this way, on-policy training in GPU-accelerated environments can be used to refine behaviors for which imitation learning alone does not lead to the desired robustness or generalization. For all tasks of the investigated complexity, this paradigm was able to produce robust policies, leading to no observed cases of failure in the final test-rollouts.

In Figure 7 we show examples of the learned behaviors for pure RL and DAPG on the PourCup task. An interesting difference is that PPO learns an unexpected strategy, where

TABLE III: Success Rates of evaluated methods. We performed at least three runs with 100 test episodes per run of each configuration.

Method	OpenDrawer	OpenDoor	PourCup	LiftObject
BC	**1.0 ± 0.0**	0.96 ± 0.02	0.76 ± 0.23	0.27 ± 0.03
PPO-dense	**1.0 ± 0.0**	**1.0 ± 0.0**	0.98 ± 0.01	0.97 ± 0.08
PPO-sparse	**1.0 ± 0.0**	0.0 ± 0.0	0.48 ± 0.48	0.14 ± 0.33
DAPG-sparse	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**

PPO



DAPG

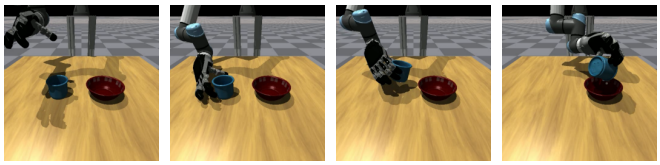


Fig. 7: Different solution strategies employed by PPO and DAPG on the PourCup task.

the particles in the cup are poured by knocking it over with the palm of the hand in just the right way. Even though this technically fulfills the objective of the problem, the intended behavior has not been learned. DAPG, on the other hand, is more likely to stick with the behaviors observed in the initial demonstrations. How much the algorithm deviates from these behaviors in favor of strategies that yield higher rewards is determined by the weighting terms λ_0 and λ_1.

VI. Discussion and Conclusions

Our results show that adding demonstration data to the learning process of massively parallel model-free RL can leverage the complementary strengths of both approaches. Guidance from expert demonstrations provides RL with the training signal needed to make meaningful progress, while highly-parallelized RL training can be used to refine the behavior seen in the demonstrations. The purpose of this work is to highlight this useful connection and provide tools to foster research on the intersection of reinforcement and imitation learning. In future work, we aim to transfer policies learned in Nvidia Isaac Gym to a real robot setup. Therefore, accurate pose estimation or operation from visual perception directly will be required. Further interesting avenues for future work are to learn from demonstrations in multi-object environments, such as picking from cluttered bins, where generalization of the observed behaviors is crucial or to investigate whether a behavior cloning loss can be used to bias the solution found by RL, for example for functional grasps of objects. Lastly, our results showed that DAPG on GPU-accelerated RL environments is capable of solving all the manipulation tasks proposed here. Therefore, it would be interesting to investigate more complex multi-step tasks, such as removing an object from an initially closed drawer, to better explore the limitations of this approach.

References

[1] Tao Chen, Jie Xu, and Pulkit Agrawal. "A system for general in-hand object re-orientation". In: *Conference on Robot Learning (CoRL)*. PMLR. 2022, pp. 297–307.

[2] Yuanpei Chen et al. "Towards human-Level bimanual dexterous manipulation with reinforcement learning". In: *arXiv preprint arXiv:2206.08686* (2022).

[3] Ankur Handa et al. "Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system". In: *International Conference on Robotics and Automation (ICRA)*. 2020, pp. 9164–9170.

[4] Felix C Huang, R Brent Gillespie, and Arthur D Kuo. "Visual and haptic feedback contribute to tuning and online control during object manipulation". In: *Journal of Motor Behavior* 39.3 (2007), pp. 179–193.

[5] Shariq Iqbal et al. "Toward sim-to-real directional semantic grasping". In: *International Conference on Robotics and Automation (ICRA)*. 2020, pp. 7247–7253.

[6] Dmitry Kalashnikov et al. "Scalable deep reinforcement learning for vision-based robotic manipulation". In: *Conference on Robot Learning (CoRL)*. PMLR. 2018, pp. 651–673.

[7] Heecheol Kim et al. "Training robots without robots: deep imitation learning for master-to-robot policy transfer". In: *arXiv preprint arXiv:2202.09574* (2022).

[8] Kilian Kleeberger et al. "A survey on learning-based robotic grasping". In: *Current Robotics Reports* 1.4 (2020), pp. 239–249.

[9] Vikash Kumar and Emanuel Todorov. "Mujoco haptix: A virtual reality system for hand manipulation". In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE. 2015, pp. 657–663.

[10] Vikash Kumar et al. "Real-time behaviour synthesis for dynamic hand-manipulation". In: *International Conference on Robotics and Automation (ICRA)*. 2014, pp. 6808–6815.

[11] Denys Makoviichuk and Viktor Makoviychuk. *rl-games: A High-performance Framework for Reinforcement Learning*. `https://github.com/Denys88/rl_games`. 2022.

[12] Viktor Makoviychuk et al. "Isaac Gym: High performance GPU-based physics simulation for robot learning". In: *arXiv preprint arXiv:2108.10470* (2021).

[13] Ajay Mandlekar et al. "What matters in learning from offline human demonstrations for robot manipulation". In: *Conference on Robot Learning (CoRL)*. 2021.

[14] Xanthippi Markenscoff and Christos H Papadimitriou. "Optimum grip of a polygon". In: *The International Journal of Robotics Research* 8.2 (1989), pp. 17–29.

[15] Igor Mordatch, Zoran Popović, and Emanuel Todorov. "Contact-invariant optimization for hand manipulation". In: *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*. 2012, pp. 137–144.

[16] Van-Duc Nguyen. "Constructing force-closure grasps". In: *The International Journal of Robotics Research* 7.3 (1988), pp. 3–16.

[17] Yuzhe Qin, Hao Su, and Xiaolong Wang. "From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation". In: *arXiv preprint arXiv:2204.12490* (2022).

[18] Yuzhe Qin et al. "DexMV: Imitation learning for dexterous manipulation from human videos". In: *arXiv preprint arXiv:2108.05877* (2021).

[19] Deirdre Quillen et al. "Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods". In: *International Conference on Robotics and Automation (ICRA)*. 2018, pp. 6284–6291.

[20] Aravind Rajeswaran et al. "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations". In: *arXiv preprint arXiv:1709.10087* (2017).

[21] Grégoire Richard et al. "Studying the role of haptic feedback on virtual embodiment in a drawing task". In: *Frontiers in Virtual Reality* 1 (2021), p. 573167.

[22] Shadow Robot. *Shadow Teleoperation System*. 2022. URL: `https://www.shadowrobot.com/teleoperation/`.

[23] Nikita Rudin et al. "Learning to walk in minutes using massively parallel deep reinforcement learning". In: *Conference on Robot Learning (CoRL)*. PMLR. 2022, pp. 91–100.

[24] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[25] Andy Zeng et al. "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning". In: *International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 4238–4245.

[26] Hanbo Zhang et al. "Robotic Grasping from Classical to Modern: A Survey". In: *arXiv preprint arXiv:2202.03631* (2022).

[27] Tianhao Zhang et al. "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation". In: *International Conference on Robotics and Automation (ICRA)*. 2018, pp. 5628–5635.