VRGym: A Virtual Testbed for Physical and Interactive AI

Xu Xie   Hangxin Liu   Zhenliang Zhang   Yuxing Qiu   Feng Gao   Yixin Zhu   Song-Chun Zhu

UCLA Center for Vision, Cognition, Learning, and Autonomy

Los Angeles, CA 90095

{xuxie,hx.liu,zlz,yxqiu,f.gao,yixin.zhu}@ucla.edu,sczhu@stat.ucla.edu

ABSTRACT

We propose VRGym, a virtual reality (VR) testbed for realistic human-robot interaction. Different from existing toolkits and VR environments, the VRGym emphasizes on building and training both physical and interactive agents for robotics, machine learning, and cognitive science. VRGym leverages mechanisms that can generate diverse 3D scenes with high realism through physics-based simulation. We demonstrate that VRGym is able to (i) collect human interactions and fine manipulations, (ii) accommodate various robots with a ROS bridge, (iii) support experiments for human-robot interaction, and (iv) provide toolkits for training the state-of-the-art machine learning algorithms. We hope VRGym can help to advance general-purpose robotics and machine learning agents, as well as assisting human studies in the field of cognitive science.[1]

CCS CONCEPTS

• **Computing methodologies Simulation environments**; **Real-time simulation**; • **Computer systems organization External interfaces for robotics**; • **Software and its engineering Virtual worlds training simulations**.

KEYWORDS

Virtual reality, simulation, training, benchmark, ROS

1 INTRODUCTION

The past decade has witnessed a rapid development of categorical classification for objects, scenes, and actions, fueled by large datasets and benchmarks, discriminative features, and machine learning methods. Similarly, successes have also been achieved in many other domain-specific tasks, largely due to the ever-growing vast amount of labeled data and rapidly increasing computing power, combined with supervised learning methods (in particular, deep learning [9]).

---

[1]The code has been made publicly available at GitLab.

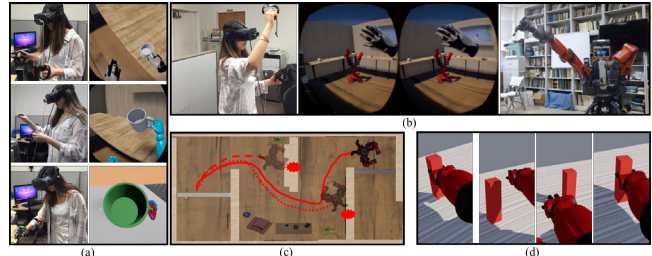Figure 1: (a) VRGym integrates three types of input devices, providing human manipulation in an increasing resolution using Oculus Touch, LeapMotion, and a data glove, from top to bottom. (b) The VRGym-ROS bridge allows physical human/robot agent meet virtual agents inside a virtual world, providing the capability of social interactions. (c) The training of the robot navigation using reinforcement learning (RL) inside VRGym. The robot successfully navigates to the goal without collisions after about 10,000 episodes. (d) The learning of object manipulation using human demonstrations (leftmost) and inverse reinforcement learning (IRL) (right three) inside VRGym.

The performance of certain tasks has reached a remarkable level, even arguably better than human in control [7, 18], grasp [13, 16], object recognition [8], learning from demonstration (LfD) [1], and playing the game of go [26] and poker [3, 19].

Despite the impressive progress, these data-driven feed-forward classification methods have well-known limitations, hindering the advancement towards a more general artificial intelligence (AI) that can interact with human: (i) needing *large labeled training datasets*; (ii) often *task-specific* and view-dependent, which makes it difficult to generalize; (iii) lacking an *explicit representation and structure* to handle large variations exhibited in and outside of the training data.

In contrast, the hallmark of machine intelligence is the capability to rapidly adapt to new tasks and "achieve goals in a wide range of environments [12]". To achieve such intelligence, recent years have seen the increasing use of synthetic data and simulation platforms[2]. Advantages include: (i) the structure of the data is efficiently encoded *without the need for human labeling* as the simulation inherently comes with the ground truth; (ii) can accommodate different embodied agents (*e.g.*, humans, humanoid robots, or turtle-bots); and (iii) benchmark *generalization* in various tasks at a low cost.

Empowered by the gaming industries, tremendous amount of game contents, including scenes and objects, are made available for the virtual environment. Meanwhile, more sophisticated physics-based simulation engines and rendering techniques have enabled more realistic simulations. These characteristics allow a growing number of tasks to be performed using synthetic data in simulation platforms. Furthermore, some simulation platforms also become publicly available, such as AirSim [24], AI2THOR [11], Gibson [31], *etc.*, promoting the further explorations and applications. In short, it

---

[2]See a brief review in the supplementary.
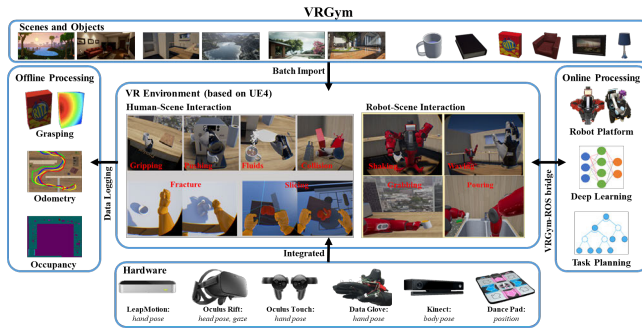
Figure 2: System architecture of VRGym, consist of three major components: (i) Hardware modules for human data input. (ii) Scene modules batch import various category of scenes as well as diverse objects, derived from different resources such as 3D modeling tools, scanned models, and automatically generated synthetic data. (iii) VR environment serves as an ideal testbed, where both a human and a robot can perform diverse tasks. The inherent physics-simulation engine enables realistic human-scene interactions and robot-scene interactions.

is both the research and the engineering efforts that make it possible to achieve considerable successes in some AI tasks and applications.

However, prior work often lacks the human involvement, especially in high-level tasks. For instance, although some virtual platforms (*e.g.*, OpenAI Gym [2] and Mujoco [28]) allow to train a virtual robot to perform many manipulation tasks, they lack a human in-the-loop, thus cannot handle critical tasks like intention prediction and social interaction. Hence, having a simulation environment where a robot can interact realistically with a human and evolve incrementally could facilitate the robotics developments.

In this paper, we propose VRGym—a virtual reality testbed, which combines VR with virtual training for both physical and interactive AI agents. By putting human in-the-loop, VRGym goes beyond the traditional synthetic data and simulation platforms by simulating a human-robot co-existing environment.

Specifically, VRGym tries to fill in the gap between the new advancement of VR and the need for training virtual agents to collaborate with human. In particular, we hope to address three critical issues. First, what is the best way to reflect human embodiment in VR; *i.e.*, how humans can genuinely interact with robots and how the robots can perceive related data that are sufficiently close to those in real life? Second, how to take advantages of current well-developed algorithms and models? Third, to which level of unique interactions the VR simulations can afford? To answer these questions, VRGym is designed to push the limits of current akin simulators by offering the following characteristics.

Fine-grained human embodiment representation   Adding a real human in the simulation is not a trivial task. Most of the current simulation platforms only support either scripted or limited remote-controlled human models. In VRGym, we integrate a multi-sensor setup as alternatives to traditional VR input devices. Our setup is capable of providing a whole-body sensing and reflecting the measured data on a detailed human avatar. As a result, the simulation can account for both body and hand poses during interactions. Figure 1a shows different resolutions of manipulations in VRGym.

High compatibility with existing robotics systems and algorithms   In VRGym, we build an efficient bi-directional communication interface with the Robot Operating System (ROS). Figure 1b

depicts an example of how a person interacts with a robot in VR-Gym, supported by the VRGym-ROS bridge. As a result, all ROS-compatible resources can be used in VRGym with little effort, which allows easy setups, training, evaluations, and benchmark.

Multiple levels of interactions   By providing the fine-grained human embodiment representation and the ROS integration, various interactions between humans and autonomous agents are made possible in different resolutions. VRGym supports interactions as simple as only providing visual/perception information and as sophisticated as learning complex robot grasping from human demonstrations. Figure 1c shows how an agent obtains a navigation policy using RL, and Figure 1d shows learning a grasp policy using IRL.

VRGym makes the three contributions:

- A comprehensive simulation platform that integrates UE4 built-in functions, *e.g.*, scene, physics-based simulation, rendering, basic human inputs, with customized developments, aiming to facilitate a variety of AI researches.
- A multi-sensor hardware and software setup that allows the whole body sensing and reflects human subjects to virtual embodiments with great details. The generated data can be seamlessly logged for online and offline training purposes.
- VRGym-ROS bridge enables a bi-directional data communication. Through this interface, AI researchers can take advantages of the existing robotics models and algorithms. Similarly, robotics researchers can utilize more sophisticated physics-based simulation.

2 VRGYM SYSTEM ARCHITECTURE

Figure 2 illustrates the system architecture of the VRGym. VRGym offers a variety of realistic scenes and tasks for both humans and robots, and provide automatic logging of the data during agents performing tasks. This capability is provided by the integration of three main modules: (i) scene module which renders user-specified 3D scenes and objects, (ii) VR environment based on UE4 with physics-simulation engine, introducing various physical properties that enrich tasks and data, and (iii) VR hardware module that imports a human agent's state and command to the VRGym. We now further elaborate each module in the following subsections.

2.1 Scene Module

Scenes and objects are the building blocks for a simulation environment. In order to increase the variety of environments for VRGym, we develop several pipelines to import or create scenes into VRGym based on the users' specifications. The scene module enriches static environments for VRGym. Note that the ground truth of RGB image, depth image, surface normal, and object label come automatically with the scene module in real-time, enabling the training for machine learning models and robotics applications.

Specifically, VRGym can directly import the entire 3D scenes provided in large open-source datasets, either collected from the web [5, 27] or automatically generated from a given set of objects [10, 22, 32] (see top of Figure 2). Additionally, VRGym also supports manually constructed scenes (see Figure 4) for more specific tasks, where neither the open-source scene dataset or the automatically generated scenes could satisfy such constraints.

Similarly, individual objects can be imported to VRGym from mesh files, which can be obtained from open-source CAD datasets

Figure 3: Examples of various physics-based simulation for diverse tasks in VRGym beyond merely rigid-body simulation in other 3D virtual environments. (Top) Pouring water. (Bottom) Folding clothes.

(*e.g.*, [4, 6]). Customized or complex objects can be manually created or scanned using a RGB-D sensor to import to VRGym for specific tasks. After the import, users can further adjust static meshes, textures, materials, and collision boundaries of the objects.

## 2.2 Real-time Physics-based Simulation

We choose UE4 as the simulation engine for VRGym for its advanced real-time physics-based simulation. Unlike previous 3D virtual environments that mostly focus on rigid body simulation or symbolic-level event simulation, VRGym integrates the advanced simulation provided by UE4 to enable a large set of various simulations, including rigid body, soft body, collision, fluid, cloth, slicing, and fracture. Some examples are shown in Figure 3 and the center of Figure 2. As a result, subtle object state or fluent [20] changes due to the virtual agent's actions are realistic and diversify. Integrating with such sophisticated physics-based simulations, VRGym not only increases the task complexity and improves the visual experience of human agents, but also affords more complicated task simulations for both virtual and physical robots.

2.3 Human Embodiment in VRGym

Compared to other similar 3D virtual environments, VRGym has another distinct feature; *i.e.*, introducing the capability to represent the physical human agent's embodiment in real-time as an avatar in the virtual environment. To reflect human movements and manipulations accurately, the physical human agent is tracked in real-time, resulting in a humanoid mesh that can deform accordingly based on the underlying tracked body skeleton and the hand poses.

Specifically, the setup includes: (i) A Kinect One RGB-D sensor to map human skeleton to the avatar in real-time through a customized-built Kinect plugin developed in UE4, (ii) an Oculus headset to record the head pose, (iii) a dance pad to navigate the avatar inside a large virtual world, and (iv) three types of input devices that provide manipulation information in different resolutions. Compared to other platforms, VRGym emphasizes the capability for users to interact with virtual environments. Depending on the needs, the user can use one of the three input devices for manipulation:

- Oculus Touch Controller offers an attachment-based approach; *i.e.*, the virtual object will automatically attach to the virtual controller/hand once the user triggers the grasp event. It enables a firm-grip manipulation, providing a firm but the least realistic grasp during the human-object interaction. Such manipulation is effective in the event-level tasks where the fine-grained hand pose is not required; *e.g.*, pick and place.

- The commercial hand pose sensing products (*e.g.*, LeapMotion) provide the vision-based gesture recognition. It is a low-cost and
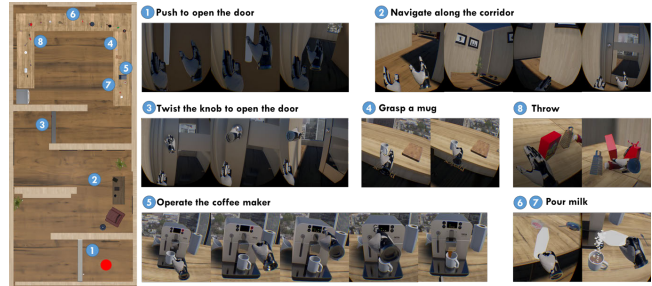
Figure 4: A human agent performs a series of actions in a virtual scene using Oculus Touch controllers. (Left) Action sequence from a top view of a virtual indoor environment. (Right) Sequences of the performed actions. Specifically, the human agent starts at the red dot as shown in the left, (1) pushes a door, (2) navigates along the hall, (3) twists a door to enter the kitchen, and (4)-(7) makes a cup of coffee. This process involves (i) large movements using the human embodiment provided in VRGym (navigating along the hallway), (ii) complex operations (operating the coffee maker), (iii) fine-grained manipulations (twisting the doorknob), and (iv) physics-rich controls (pouring milk).

off-the-shelf solution that can be easily set up by mounting the sensor on the head-mounted display. However, it is difficult to have a firm grasp due to occlusions and sensor noises. Note that the hand tracking will fail if the hand is not within the view.

- An open-sourced glove-based device [15] is also compatible with VRGym to provide the finest-grained manipulation. It requires a Vive Tracker to provide global positioning of the hand, and an IMU network in the glove to measure the rotation of each phalanx and calculate the hand poses using forward kinematics. Although glove-based devices are costly compared to other alternatives, they allow reliable hand pose sensing, which is vital for the tasks with detailed, complex and subtle hand manipulations.

3 SOFTWARE INTERFACE DESIGN

VRGym has two major software interfaces developed to enable training and benchmarking both physical and interactive AI agents. The first interface is the human data logging system that builds on top of the hardware setups to collect the data generated during the interactions between the avatar and the environment. Another interface, a VRGym-ROS bridge, is introduced to allow seamlessly import of robot models and robotics algorithms from ROS. The collected data together with the VRGym-ROS bridge could be used for a variety of AI applications; see examples in Section 4.

To demonstrate the functions of these two interfaces, we consider a task-rich environment built for the VRGym. Figure 4 depicts an environment in VRGym that provides semantically-diverse tasks to the agent. Note that although such environment could be constructed in the real world to perform the demonstrated tasks, sensing and logging the detailed data generated during the interactions between the agent and objects would be extremely difficult in practice.

In such a typical virtual environment in VRGym, an agent (a human as an avatar or a virtual robot) is initially placed on the starting point, indicated as the red dot in Figure 4. The final goal for the agent is to reach the kitchen located at the far-end, and accomplish several sub-tasks. At the beginning, the agent has to push to open the first door and navigate along the corridor, requiring *large*

(a)   (b)

(c)
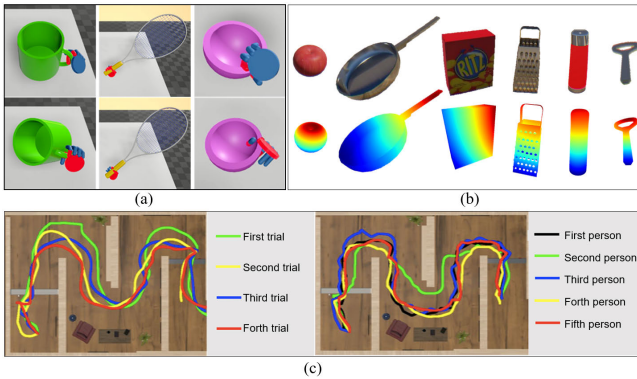
Figure 5: (a) Grasp a mug, a tennis racket, and a bowl. The red area indicates the contact force between the virtual hands and the object. (b) Visualization of the collected human grasp data. Top: a set of 3D objects. Bottom: the average grasp heat map generated by multiple subjects. (c) Visualization of footprint from different subjects.

movements. Then the agent must go through another door to enter into the kitchen, and the only solution is to twist the doorknob using complex *manipulations*. Inside the kitchen, the agent is required to make a cup of coffee with milk, which needs to grasp and move a mug, operate the coffee maker by pushing several buttons in a certain order. The entire procedure requires the *task planning* empowered by the *physics-based simulation*.

3.1 Human Data Logging

When a user performs a task, data generated by the interactions between an avatar and the environment can be directly logged with ground-truth labels in VRGym. In this section, we showcase two scenarios where the data is logged and used in other applications.

Grasping Finer-grained manipulation is made feasible in VR-Gym using a glove-based device [15]; see Figure 5a for some results. By collecting a set of subjects' grasp data on a variety of objects, we can merge all the collected grasp data to form heat maps on different objects to visualize the likelihood of grasp points on man-made objects. Specifically, the grasp data shown in Figure 5b is the averaging data of heat maps collected from 10 human subjects, where the hotter the area is, the denser the grasp points are, and the more likely a human agent would grasp around that area.

Footprints VRGym provides the function to log an agent's footprints or the odometry data. Figure 5c shows the recorded odometry data from 5 human subjects who have limited VR experience. Each of the participants navigates from the starting point to the kitchen room along the corridor using Oculus Touch controllers.

3.2 ROS Interface

The VRGym is compatible with the popular ROS framework through a customized VRGym-ROS communication bridge. This bridge allows the off-the-shelf ROS robot models to communicate with the simulations and human agents in VRGym with minimal efforts; *e.g.*, the diverse scenes rendered in VRGym can also be exported to the Gazebo simulator, which is highly compatible with ROS.

Implementation We develop a ROS interface, VRGym-ROS bridge, based on the TCP/IP protocol in order to enable VRGym to communicate with the existing popular robotics platforms. Through

(a)   (b)

Figure 6: VRGym-ROS bridge. (a) The robot navigation in the scene imported into the Gazebo, exported from the VRGym. The red curve indicates the path planned by the robot's global planner. The black curve is the actual trajectory executed by the robot. (b) A Husky-UR5 robot is imported into VRGym from ROS to guide the way and open the door for a human agent.

this interface, robot body parts can be easily imported to VR environments as mesh files and control signals, and a data stream can be seamlessly transferred between the VRGym and the robot platforms using ROS to communicate with either physical or virtual robots. We organize all data types (*i.e.*, ROS topics) in a unified JSON format and construct JSON parsers in both VRGym and ROS to further improve the compatibility. Each port in the protocol supports a stream of data, making it possible to present multiple agents from ROS into the VRGym. With the VRGym-ROS bridge, we present two examples of training and evaluating human robot interactions (HRI) inside VRGym in Section 4, which incorporates direct human reactions and involvements. Such capability is largely missing in the current robotics simulators such as Gazebo or V-Rep. The benchmark in Section 4 is also supported by this VRGym-ROS bridge.

Evaluation We evaluate the VRGym-ROS bridge on a navigation task (see Figure 4) using a Clearpath Husky robot. This navigation task is performed in VRGym, whereas the robot model is imported from ROS, making it possible to evaluate a number of SLAM algorithms and path planning approaches. In 6a, the mapping result is obtained using the conventional GMapping package in ROS. The red curve indicates the planned path, whereas the black curve is the actual odometry of the Husky robot. 6b shows the user's view when the robot is moving. This VRGym-ROS bridge fills in the gap between the diverse scenes in VRGym and the existing fine-tuned algorithms provided in ROS.

Communication Bandwidth To evaluate the reliability and efficiency of the VRGym-ROS bridge, we conduct an experiment by sending packages with the size of 512Kb[3].

4 EXPERIMENTS

In this section, we demonstrate the performance and capability of the VRGym from four different perspectives[4]. Two human robot interaction (HRI) applications are conducted, including a human intention prediction task and a social interaction task. Like other testbeds, we also benchmark the performance popular machine learning algorithms (*e.g.*, reinforcement learning and IRL) in the VRGym.

4.1 Experiment 1: Intention Prediction

Predicting human intention is difficult when training on a physical robot since this task has very small error tolerance; wrong predictions may endanger both the human and the robot. It is particularly

---

[3]See a detailed evaluation in supplementary.

[4]See a video demo at Vimeo.
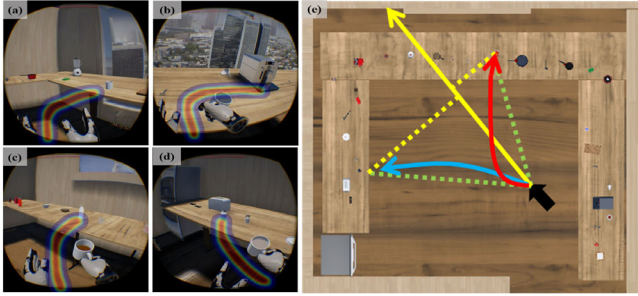
Figure 7: Intention predictions in a coffee-making task. (a) Grab a cup. (b) Use the coffee machine. (c) Pour milk. (d) Add sugar. (Right) Visualization of three intention prediction algorithms. Blue and Red: sampled paths from the grammar model [21]. Green: straight-line distance. Yellow: prediction by shortest perpendicular distance (dashed lines) from objects to the ray direction (solid arrow) based on avatar's location.

interesting to study human intention prediction in VRGym, since this problem involves complicated inference process that many types of data can be useful: human trajectories, human poses, object positions, object states, and first/third-person vision inputs, *etc.*. Predicting intention is made possible in VRGym as our unique multi-sensor setup reflects human poses, and the odometry data provided by the data logging system indicates human's trajectories.

In the experiment, we analyze different human intention prediction algorithms to demonstrate the potential of VRGym as a testbed for both physical and virtual AI agents. Additionally, we show the unification of both the learning and the inference enabled by the VRGym. 20 subjects are recruited. The virtual environment is set up as a virtual kitchen, in which more than 20 objects are placed on top of three long tables. The layout of the kitchen is shown in Figure 7, where the agent starts from the entrance of the room (red dot) and performs the task with at least 4 steps: grasp a mug, operate the coffee maker, add milk, and add sugar. Note these tasks can perform in different orders. The resulting footprint from one subject is plotted in Figure 7. All subjects are required to perform a coffee-making task—making a cup of coffee using the available objects.

Figure 7e illustrates the comparisons among these three methods. The qualitative results are shown in Figure 7a-d to reveal the intention of the agent as the heat maps during the process of making coffee, where hotter color (red) indicates higher probability. This high-level semantic prediction is inferred given multiple human demonstrations as logged navigation and grasp data collected from the agent using VRGym.

4.2 Experiment 2: Social Interaction

Social interactions or social HRI is a vital topic enabling human-robot co-existing environment, since the robot needs to understand and respond properly to human's social behaviors, such as waving and hand-shaking. Although the current robot simulators (*e.g.*, Gazebo and V-Rep) provide a suite of features, one key element these simulation platforms still largely missing is the direct human involvement which is crucial for human-robot interaction studies.

Participants A total of 10 subjects were recruited. We implemented the algorithm proposed in [25] for robot learning social affordance. The algorithm is briefly described as follows; we refer the readers to the original paper for more technical details.

Figure 8: Human robot interactions in VRGym. A Baxter robot (a) waves hands and (b) shake hands with a virtual human agent.

Results Qualitative results are shown in Figure 8. Concretely, the robot starts hand-waving in response to the agent's hand waving (Figure 8a), illustrated by a virtual hand model. The robot stretches out its manipulator to make a handshake with the virtual agent (Figure 8b). Technically, when the virtual Baxter inside the VRGym perceives the action signals from a virtual human such as *hand shaking* or *hand stretching out*, it sends the action signals to ROS through the VRGym-ROS bridge. In ROS, the motion planning will generate corresponding body parts transformations and send the computed transformation data back to the virtual Baxter inside VRGym, such that it will then act with the appropriate responses to the virtual human agent. In this sense, the proposed VRGym enables a new approach to study social human-robot interaction without using a costly physical robot or having a physical contact between a subject and robots, which in some cases could be dangerous.

4.3 Experiment 3: RL Algorithms Benchmark

We introduce a playground as a sub-module (Figure 9) inside the VRGym, aiming to train robots to navigate in a 3D maze-like indoor corridor. The overall goal is to teach the robot agent itself by trial and error to obtain a navigation policy, reaching the final goal of the maze. The learning strategy applied on the virtual robot follows the standard RL framework. A Baxter robot is integrated into the VRGym and controlled by off-the-shelf ROS packages.

Compared to other virtual playgrounds (*e.g.*, OpenAI Gym), the proposed VRGym differs in two primary aspects.

- *Sophisticated Interactions*. With the advanced physics-based simulator, the VRGym offers realistic interactions between the virtual agent and the virtual environment.
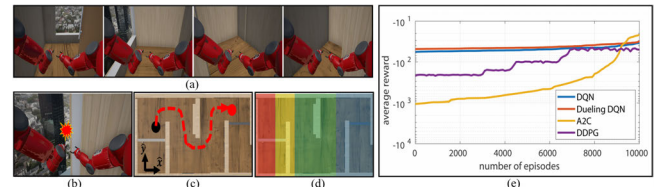
Figure 9: Settings for the RL training inside VRGym environment for an indoor maze navigation task. (a) First-person view of a virtual robot. (b) The robot collides with a wall, triggering negative rewards. (c) An eagle view of the indoor navigation task. (d) Rewards assigned in different color zones (red, yellow, green and blue) from low to high. (e) The performances of the tested RL algorithms.
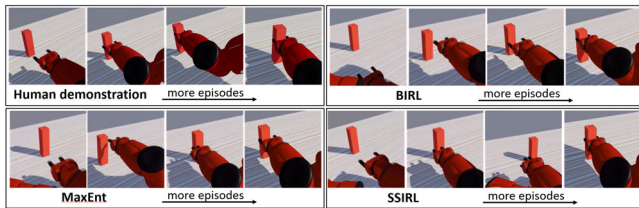
Figure 10: Learning human grasping demonstration with different IRL frameworks.

- *Physical RL Agent*. Since the VRGym is capable of importing both the physical and the virtual robot model to the virtual scene, it is feasible to transfer RL model trained inside the virtual environment directly to a physical robot agent.

We conduct four state-of-the-art deep RL algorithms to demonstrate the VRGym's capability in RL related tasks. These algorithms are **DDPG** [14], **DQN** [18], **Actor-Critic** [17], and **Dueling DQN** [30]. All four algorithms use the pixel-input from the first-person camera view. The quantitative comparison of the above four algorithms in VRGym is plotted in Figure 9e.

4.4 Experiment 4: IRL for Learning Grasp

Grasp is an imperative capability for an interactive agent. In this experiment, we adopt an inverse reinforcement learning (IRL) framework to enable a virtual robot learning to grasp from human demonstrations. This task primarily involves both the data logging function in VRGym and a ROS motion planer communicated by the VRGym-ROS bridge. The robot is expected to learn how to successfully grasp an unknown object based on the hand trajectories demonstrated by the human subjects, collected through tele-operations using the Oculus Touch Controller inside the VRGym.

The trajectories of the human demonstrations are logged and used to infer the model and its parameters. Later, with the learned model and its parameters, the robot can be executed using the motion planner in ROS to grasp an unknown objects in the virtual environment.

Three IRL algorithms are implemented in the VRGym: Bayesian-IRL [23], Maximum Entropy-IRL [33], and Semi-supervised-IRL [29]. Qualitative results are shown in Figure 10.

5 CONCLUSION

In this paper, we introduce the VRGym as a promising simulation platform for training and evaluating autonomous agents to build the physical and interactive AI. VRGym can represent a fine-grained human embodiment as a virtual avatar using a range of hardware setups for body and manipulation sensing. Existing robotics systems and algorithms developed in ROS can also be integrated to VRGym through a customized VRGym-ROS bridge. Multiple evaluations indicate that the VRGym has a robust performance at the system level and in the communication with ROS. Our experiments have demonstrated that four different robotics interactive tasks can be successfully trained using RL and IRL inside VRGym. Specifically, we showcase how the data logged from the VRGym is useful in several interaction tasks, combining with the functions (*e.g.*, motion planners, robot models) provided by ROS through the VRGym-ROS bridge. The successful implements of RL and IRL for robotics interactive tasks in VRGym also support the training capability offered by VRGym in training robots with advanced machine learning methods.

We believe VRGym could have further potential applications and it will benefit future research on the physical and interactive AI.

REFERENCES

[1] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 5 (2009), 469–483.
[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym.
[3] Noam Brown and Tuomas Sandholm. 2018. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* 359, 6374 (2018), 418–424.
[4] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. 2015. Benchmarking in Manipulation Research. *IEEE Robotics & Automation Magazine* 1070, 9932/15 (2015), 36.
[5] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *International Conference on 3D Vision (3DV)*.
[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository.
[7] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. 2016. Benchmarking deep reinforcement learning for continuous control. In *ICML*.
[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *CVPR*.
[9] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.
[10] Chenfanfu Jiang, Siyuan Qi, Yixin Zhu, Siyuan Huang, Jenny Lin, Lap-Fai Yu, Demetri Terzopoulos, and Song-Chun Zhu. 2018. Configurable 3D Scene Synthesis and 2D Image Rendering with Per-pixel Ground Truth Using Stochastic Grammars. *IJCV* 126, 9 (2018), 920–941.
[11] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. 2017. AI2-THOR: An interactive 3d environment for visual AI.
[12] Shane Legg and Marcus Hutter. 2007. Universal intelligence: A definition of machine intelligence. *Minds and Machines* 17, 4 (2007), 391–444.
[13] Ian Lenz, Honglak Lee, and Ashutosh Saxena. 2015. Deep learning for detecting robotic grasps. *IJRR* 34, 4-5 (2015), 705–724.
[14] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *ICLR*.
[15] Hangxin Liu, Zhenliang Zhang, Xie Xu, Yixin Zhu, Yue Liu, Yongtian Wang, and Song-Chun Zhu. 2019. High-Fidelity Grasping in Virtual Reality using a Glove-based System. In *ICRA*.
[16] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. 2017. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. In *RSS*.
[17] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *ICML*.
[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.
[19] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. 2017. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356, 6337 (2017), 508–513.
[20] Isaac Newton and John Colson. 1736. *The Method of Fluxions and Infinite Series; with Its Application to the Geometry of Curve-lines*. Henry Woodfall.
[21] Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. 2017. Predicting Human Activities Using Stochastic Grammar. In *ICCV*.
[22] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. 2018. Human-centric Indoor Scene Synthesis Using Stochastic Grammar. In *CVPR*.
[23] Deepak Ramachandran and Eyal Amir. 2007. Bayesian inverse reinforcement learning. *Urbana* 51, 61801 (2007), 1–4.
[24] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*. Springer, 621–635.
[25] Tianmin Shu, Xiaofeng Gao, Michael S Ryoo, and Song-Chun Zhu. 2017. Learning Social Affordance Grammar from Videos: Transferring Human Interactions to Human-Robot Interactions. In *ICRA*.
[26] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484.
[27] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. 2017. Semantic Scene Completion From a Single Depth Image. In *CVPR*.
[28] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *IROS*.
[29] Michal Valko, Mohammad Ghavamzadeh, and Alessandro Lazaric. 2012. Semi-Supervised Apprenticeship Learning.. In *EWRL*.
[30] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. 2016. Dueling Network Architectures for Deep Reinforcement Learning. In *ICML*.
[31] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. 2018. Gibson Env: Real-World Perception for Embodied Agents. In *CVPR*.
[32] Lap-Fai Yu, Sai-Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. 2011. Make it home: automatic optimization of furniture arrangement. *TOG* 30, 4 (2011), 86.
[33] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum Entropy Inverse Reinforcement Learning. In *AAAI*.