PerAct^2: Benchmarking and Learning for Robotic Bimanual Manipulation Tasks

Markus Grotz
University of Washington
grotz@cs.washington.edu

Mohit Shridhar
University of Washington
mshr@cs.washington.edu

Yu-Wei Chao
NVIDIA
ychao@nvidia.com

Tamim Asfour
Karlsruhe Institute of Technology
asfour@kit.edu

Dieter Fox
University of Washington, NVIDIA
fox@cs.washington.edu

Abstract: Bimanual manipulation is challenging due to precise spatial and temporal coordination required between two arms. While there exist several real-world bimanual systems, there is a lack of simulated benchmarks with a large task diversity for systematically studying bimanual capabilities across a wide range of tabletop tasks. This paper addresses the gap by extending RLBench [1] to bimanual manipulation. We open-source our code and benchmark, which comprises 18 new tasks with 41 unique task variations, each requiring a high degree of coordination and adaptability. To initiate the benchmark, we extended several state-of-the-art methods to bimanual manipulation and also present a language-conditioned behavioral cloning agent – PerAct^2, an extension of the PerAct [2] framework. This method enables the learning and execution of bimanual 6-DoF manipulation tasks. Our novel network architecture efficiently integrates language processing with action prediction, allowing robots to understand and perform complex bimanual tasks in response to user-specified goals.

Keywords: Bimanual Benchmarking, Imitation Learning, Bimanual Manipulation

1 Introduction

Humans seamlessly manipulate and interact with their environment using both hands. With both hands, humans achieve greater efficiency through enhanced reachability and can solve more sophisticated tasks. Despite the recent advances in grasping and manipulation planning [3, 4] the investigation of bimanual manipulation remains an under-explored area, especially in terms of learning a manipulation policy. Unlike tasks that require grasping or manipulation with a single hand, bimanual manipulation introduces a layer of complexity due to the need for spatial and temporal coordination and a deep understanding of the task at hand. This complexity is compounded by the dynamic nature of real-world tasks, where the state of the environment and the objects within it are constantly changing, demanding continuous adjustment and coordination between both arms.

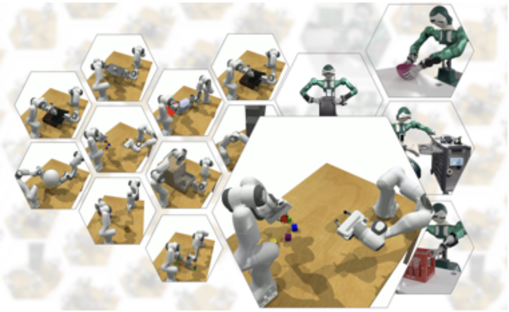


Figure 1: Selected bimanual tasks from the benchmark as well as real-world examples. Due to the architecture design the method can easily be transferred to other robots as the policy outputs a 6-D pose and is agnostic to the underlying controller.

CoRL 2024 Workshop on Whole-Body Control and Bimanual Manipulation (CoRL 2024 WCBM).

With the recent advent of complex bimanual systems such as the Boston Dynamics' Atlas, Tesla's Optimus or Figure AI's Humanoid, experiments investigating bimanual manipulation in real-world tasks are rising. Notably, the work by Zhao et al. [5, 6] presents sophisticated and fine-grained real-world tasks learned from demos collected by teleoperation. While real-world tasks provide a rich context for understanding the challenges of bimanual manipulation, they suffer from issues of reproducibility and variability that make systematic assessment difficult. To advance research in bimanual manipulation, there is a critical need for a dedicated and rich bimanual benchmark that allows for the reproducible and systematic evaluation of new methods and models. To fill this gap, we expand the robot learning benchmark RLBench [1] to bimanual manipulation. A significant benefit is the capacity to autonomously generate training data without the necessity of human demonstrations to the robot. Moreover, we extend two existing unimanual learning-based methods, namely PerAct and RVT, to bimanual manipulation and compare those with ACT. While benchmarking, we found that running two separate agents is insufficient and that coordination is a crucial aspect. Hence, we present a method to learn bimanual actions as well coordination implicitly using language-conditioned behavior-cloning agent within a single network. Overall, our contributions can be summarized as follows:

1.) A benchmark with 18 bimanual manipulation tasks and 41 unique tasks variations. RLBench is used as a basis preserving its functionality and its key properties.
2.) A novel network architecture, called PerAct^2, based on the PerAct framework to predict bimanual manipulation actions, and
3.) Qualitative experiments in real world.

We acknowledge the complexity of the tasks included in the benchmarks and look forward to the research community to embrace these challenges. We also hope that our method and evaluation will greatly enhance the benchmarking and generalization of skill learning in bimanual robots, including humanoids.

2 Related Work

Benchmark Name	# Bimanual Tasks[1]	Task Variation	Dataset Generation
Robotsuite	3	✓	✗[2]
ManiSkill	2	✓	✗
RLBench	–	✓	✓
Orbit	1	✓	✗[2]
HumanoidBench	8	✓	✗
ours	18	✓	✓

Table 1: Overview of bimanual benchmarks.

Benchmarking Benchmark protocols and frameworks for robotic manipulation are designed around reproducibility and extensibility. For reinforcement learning, Fan et al. [7] introduce *SURREAL* to foster reproducibility for learning robotic manipulation tasks. Similarly, *robosuite* [8], *bulletarm* [9] and *ManiSkill2* [10] provide a standardized benchmark and learning-environment for robotic manipulation. While some of them have three (*robosuite*) or two (*ManiSkill2*) bimanual tasks, those are not sufficient to efficiently evaluate methods for bimanual manipulation. James et al. [1] present RLBench, a large scale benchmark and learning-environment for robot learning alongside with baseline algorithms. A crucial aspect here is the automated waypoint-based dataset generation, removing the need of human demonstrations or by another baseline such as in [11]. However, no bimanual manipulation tasks were considered in RLBench. An issue that arises is the comparability, especially, in real world scenarios. RB2 [12] aims to provide rankings for robotic manipulation tasks by pooling data across labs. Mittal et al. [13] introduce *Orbit*, a framework with GPU acceleration and photo realistic scenes, to tackle real-to-sim gap. Their work also includes a bimanual manipulation task, but this was not the main focus of the work. Other works have a focus on dexterous bimanual hand-object manipulation, such as [14] or [15]. However, these works focus on the hand and neglect the use of a robotic arm. Last

[1] We only count tasks were both arms are required.

[2] Robomimic allows for dataset generation either through human demonstrations or with a baseline.

but not least, instead of providing a set of standardized benchmarks in simulation, another way is to establish protocols for real-world robot experiments. Chatzilygeroudis et al. [16] outline a protocol for bimanual manipulation of two challenging tasks for semi-deformable objects. Recently, [17] introduced *HumanoidBench*, a benchmark for whole-body manipulation and locomotion for reinforcement learning. Tab. 1 overviews different robotic benchmarks.

Bimanual Manipulation Bimanual manipulation offers several advantages, such as increased reachability and enhanced dexterity [18]. In general approaches vary depending on the domain. Key challenges for bimanual manipulation are the coordination and the state complexity, i.e., how to orchestrate the arms with respect to each other. In the following, we want to specifically discuss work that addresses those issues.
Coordination is central aspect for bimanual manipulation, for example when a robot is playing the piano [19]. Coordination can be achieved by modeling coordination constraints explicitly besides task constraints [20] to reach a moving target simultaneously. With TAMP this robotic assembly planning can be solved explicitly [21]. Other areas include object carrying [22]. Indeed, this requires knowledge about the environment and the physical parameters, which can be cumbersome when generalizing to other tasks and scenarios. With an explicit leader-follower assumption and when a DMP [23] for the leader is known, the coordination can be also learned using a structured-transformer to generate DMPs for the follower arm [24]. Other work [25] learns a separate coordination module for each gripper and coordination is done by a separate module. Concurrently, research by Grannen et al. investigates using one arm to stabilize an object while manipulating it with the other [26]. The symmetry-aware context has been studied for multi-object handover and rearrangement tasks [27]. This work has also been evaluated in real-world. Notably, for real-world robotics notably Zhao et al. [5] learn bimanual manipulation from teleoperation. The work was later extended to mobile manipulation [6] and also revised [28]. In independent and concurrent work, [29] propose VoxActB, a voxel-based, language-conditioned method for bimanual manipulation using VLMs to focus on the most important regions. This work is complementary to ours, and future work could merge techniques from both works.

3 Method

To benchmark, we extend RLBench [1] to the complex bimanual case by adding functionality and tasks for bimanual manipulation. We choose RLBench for several key advantages over other frameworks, mainly its ability to generate training data with variations as well as the widespread acceptance in the learning community. To initiate the benchmark, we also present a bimanual behavioral cloning agent. Our method, called `PerAct`^{2}, extends `PerAct` [2], which learns a single language-conditioned policy for unimanual manipulation actions. Fig. 3 illustrates the system architecture, which has been modified to accommodate the intricate coordination required between two robotic arms while responding to language instructions.

3.1 RLBench2

RLBench is a robot learning benchmark suite featuring over more than 100 tasks to facilitate robot learning, which is widely used in the community. Among task diversity, other key properties include reproducibility and the ability to adapt to different learning strategies. We extend RLBench to bimanual manipulation, while keeping the functionality and its key properties. This allows us to quantify the success of our method and compare it with other baselines. Compared to unimanual manipulation, bimanual manipulation is more challenging as it requires different kinds of coordination and orchestration of the two arms. Therefore, we also provide task descriptions along with metrics to benchmark performance and outline the key challenges of each task in Appendix A. For the implementation side this makes it much more complex since synchronization is required when controlling both arms at the same time.

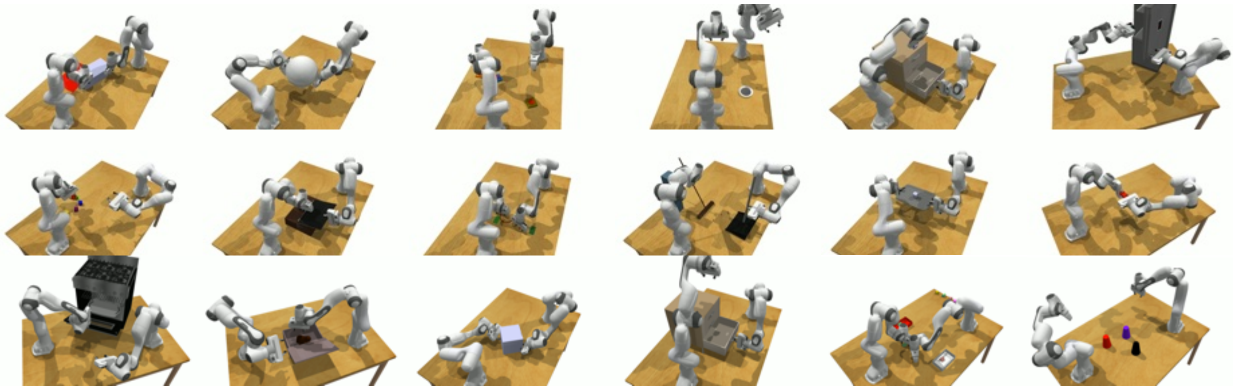


Figure 2: Overview of the different tasks. Task names from left to right and top to bottom as in Tab. 5. For example, the task visualized (g) includes a handover of a specific item.

Task	Coupled			Coordination	
	temporal	spatial	physical	sym.	sync.
(a) push box	✓	✓	✗	✓	✓
(b) lift a ball	✓	✓	✓	✓	✓
(c) push two buttons	✓	✗	✗	✓	✗
(d) pick up a plate	✓	✓	✓	✗	✗
(e) put item in drawer	✓	✗	✗	✗	✗
(f) put bottle in fridge	✓	✗	✗	✗	✗
(g) handover an item	✓	✓	✓	✗	✓
(h) pick up notebook	✓	✓	✓	✗	✗
(i) straighten rope	✓	✓	✓	✗	✓
(j) sweep dustpan	✓	✓	✓	✗	✗
(k) lift tray	✓	✓	✓	✓	✓
(l) handover item (easy)	✓	✓	✓	✗	✗
(m) take tray out of oven	✓	✗	✗	✗	✗
(n) take shoes out of box	✓	✗	✗	✗	✗
(o) lift a cube	✓	✓	✓	✓	✓
(p) open drawer	✗	✓	✗	✗	✗
(q) sort shapes	✗	✗	✗	✗	✗
(r) shell game	✓	✓	✗	✗	✗

Table 2: Classification of the bimanual tasks. Coupling can be temporal, spatial, or physical. Coordination can be symmetric (sym.) or synchronous (sync.)

3.2 Task and Challenges

We introduce 18 bimanual manipulation tasks with different coupling, coordination, language instructions and manipulation skills. Fig. 2 illustrates these tasks, which range from instructions like *"push the box to the target area"* to *"put the bottle into the fridge"*. Out of the tasks, eight are prehensile manipulation, three are non-prehensile manipulation, and two involve both. These tasks also show different kind of coupling and coordination. For example, the task *"lift the tray"* has to be executed synchronously and both arms must be coordinated. Tab. 2 classifies the tasks according to the bimanual taxonomy of [30]. Here, key distinguishing factors are the coupling as well as the required coordination between the two arms. We extended the classification in that we also distinguish between physical coupling, i.e., if one arm exerts a force that could be measured by the other arm. The benchmarking tasks differ in terms of complexity and the coordination required between two arms. Other attributes, such as the number of objects and the variation count, also affect the complexity of the task. All of these attributes influence the task complexity, and a rich and diverse set of tasks is required for both qualitative and quantitative benchmarking.

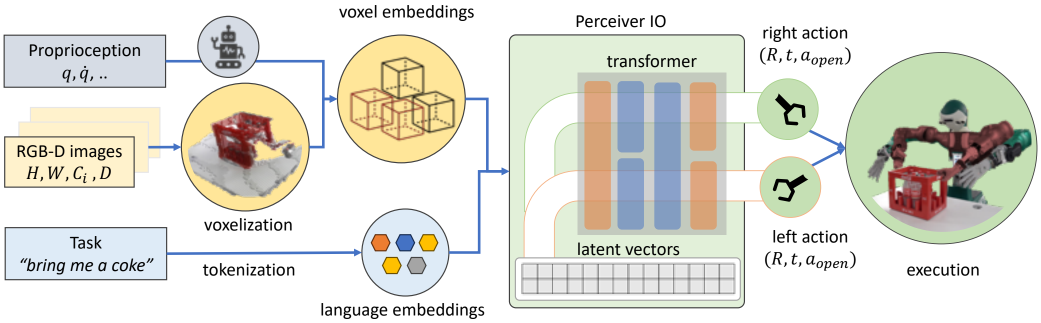


Figure 3: The system architecture. PerAct^2 takes proprioception, RGB-D camera images as well as a task description as input. The voxel grid is constructed by merging data from multiple RGB-D cameras. A PerceiverIO transformer is utilized to learn features at both the voxel and language levels. The output for each robot arm includes a discretized action, which comprises a six-dimensional end-effector pose, the state of the gripper, and an extra indicator whether the motion-planner should use collision avoidance.

3.3 PerAct^2

To address the challenges of bimanual tasks, we present a method for predicting bimanual actions, following the framework established by [2]. Our model takes as input a 3D voxel grid, proprioception data, and a language goal. The voxel grid is assembled by combining sensory data streams from several RGB-D cameras. A PerceiverIO transformer learns per-voxel and language features. For each robot arm, the output is a discretized action consisting of a 6-DoF end-effector pose, a gripper state and an additional flag for collision-aware motion planning. We choose PerAct as the basis for our application because the voxel-based representation makes it robust to changes in the view pose. Unlike RVT, which suffers from occlusion issues due to its reliance on rendered virtual images, PerAct, directly handles raw input data, avoiding these problems. Additionally, unlike ACT, which relies on joint angles and may struggle with adaptability due to the need for similar demonstrations in the joint space, PerAct and RVT are robot-agnostic and can be transferred to other robots with different degrees of freedom. Dealing with the question to get bimanual control, a naive approach to bimanual control would be to instantiate two independent instances of PerAct as separate agents, each of them controlling one robot arm. We will refer to this as two *independent* agents. Coordination between these two agents is only possible with visual perception. Another drawback is that the voxel representation is stored twice, resulting in an increased memory usage. Another approach, is to adopt a leader-follower based architecture. Here, the predicted output from one agent is passed to the second network. Hence, the prediction of the second network is based on the prediction of the first network. Once both action predictions are inferred, they are executed simultaneously. While this offers the advantage of communication, this approach yet suffers from large memory consumption and fixed roles.

In order to address the limitations of traditional voxel representations and facilitate communication without fixed roles, we introduce a novel transformer module. Our approach is distinctive in that it divides the latent space between the two actions while employs combined self-attention. Hence, our approach utilizes a single Perceiver IO as backbone to predict both actions simultaneously. Besides sharing the voxel representation, another advantage is that the proprioception of the two agents can be shared.

3.4 Expert Demonstrations

Demonstrations consist of a set of action tuples for each robot arm that are executed simultaneously. Following previous works [2], we assume a dataset $\mathcal{D} = \{\zeta_1, \zeta_2, \ldots, \zeta_n\}$ of n expert demonstrations, each paired with language goals $\mathcal{G} = \{\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_n\}$. For the bimanual setup we assume each demonstration contains two actions. Thus, each demonstration ζ_i is a sequence of continuous actions $\mathcal{A} = \{(a_1^r, a_1^l), (a_2^r, a_2^l), ..., (a_t^r, a_t^l)\}$, where the superscripts r and l denote the right or the left robot arm. Each action a contains the 6-DoF pose, gripper open state, and whether the motion-planner used collision avoidance to reach an intermediate pose: $a = \{a_{\text{pose}}, a_{\text{open}}, a_{\text{collide}}\}$. Additionally, we capture visual observations for $\mathcal{O} = \{\tilde{o}_1, \tilde{o}_2, \ldots \tilde{o}_t\}$. An observation $\tilde{o}$ consists of RGB-D images from any number of cameras. For our simulated experiments, we use a total of five cameras and thus $\tilde{o}_{\text{sim}} = \{o_{\text{front}}, o_{\text{left}}, o_{\text{right}}, o_{\text{wrist left}}, o_{\text{wrist right}}\}$. Each demonstration ζ is a sequence of continuous actions $\mathcal{A}$ paired with observations $\mathcal{O}$.

3.5 Keyframe Extraction

During the demonstrations salient keyframes are identified among the recorded visual and proprioceptive sensor data, which are used for training. Similar to prior work [2] and [31] for any given demonstration ζ_i of length m we identify keyframes $k_1 < k_2 < \ldots k_n$. We extend the heuristic to the bimanual case and define a keyframe if

1. The gripper state of one of the robot has changed, or
2. A robot reached the end of its executed trajectory, i.e., the pose of an end-effector is no longer changing.

The discretisation of keyframe actions $\mathbf{k}$ facilitates the conceptualization of our BC agent's training paradigm as a classification task, specifically focusing on the identification of the 'next best action'.

3.6 Action Inference

Our goal is to learn bimanual action-centric representations [32] and retrieve a 6-DoF pose from a voxel for each arm. Hence the 6-DoF pose is split into a translation, a rotation and a gripper state. To this end, we utilize a 3-D voxel grid [33, 34] to represent both the observation and action space. The advantage is that such a representation is view-point independent compared to ACT. The voxel grid $\mathbf{v}$ is reconstructed from several RGB-D images $\tilde{o}$ and fused through triangulation $\tilde{o} \Rightarrow \mathbf{v}$ from known camera extrinsics and intrinsics. To allow for a fine-grained action representation we use 100^3 voxels with size of $0.01\,\mathrm{m}$ to cover a workspace area of $1.0\,\mathrm{m}^3$.

The translation is identified as the voxel nearest to the gripper fingers' center. Rotation is quantified into discrete $5°$ intervals for each rotational axis. The state of the gripper a_{open}, either open or closed, is represented through a binary value. Similarly, the 'collide' parameter $a_{collide}$ is binary, and indicates if the motion-planner should avoid the voxel grid. This binary mechanism is pivotal for enabling tasks that require both contact-based actions, like opening a drawer, and non-contact maneuvers, such as reaching a handle without physical contact.

3.7 Training

We extend the loss function as in [2] to the bimanual setup and thus the loss function results in

$$\mathcal{L}_{\text{total}} = \sum_{i \in \mathcal{X}} \mathcal{L}_{\text{i}}^{\text{right}} + \sum_{i \in \mathcal{X}} \mathcal{L}_{\text{i}}^{\text{left}}$$

with $\mathcal{X} \in \{\text{trans}, \text{rot}, \text{open}, \text{collide}\}$ and $\mathcal{L}_{\text{i}} = \mathbb{E}_{Y_{\text{i}}}[\log \mathcal{V}_{\text{i}}]$, where

$$\mathcal{V}_{\text{trans}} = \text{softmax}(\mathcal{Q}_{\text{trans}}((x, y, z)|\mathbf{v}, \mathbf{l})) \qquad \mathcal{V}_{\text{rot}} = \text{softmax}(\mathcal{Q}_{\text{rot}}((\psi, \theta, \phi)|\mathbf{v}, \mathbf{l}))$$

$$\mathcal{V}_{\text{open}} = \text{softmax}(\mathcal{Q}_{\text{open}}(\omega|\mathbf{v}, \mathbf{l})) \qquad \mathcal{V}_{\text{collide}} = \text{softmax}(\mathcal{Q}_{\text{collide}}(\kappa|\mathbf{v}, \mathbf{l}))$$

Similar to PerAct, we also augment $\mathbf{v}$ and $\mathbf{k}$ with translation and rotation perturbations for robustness, keeping other parameters, such as the optimizer, the same.

4 Evaluation

We study the efficacy of robotic bimanual manipulation. To this end, we compare our method against the following baselines. a.) ACT: A transformer network with action chunking that outputs joint positions from camera inputs. b.) RVT-LF: Two Robotic View Transformer (RVT) as a leader-follower architecture. c.) `PerAct-LF`: Two Perceiver Actor networks as a leader-follower architecture. d.) `PerAct`^{2}: A single bimanual Perceiver Actor network as described in Section 3.

The leader-follower architecture consists of two networks, where the output of one network is fed as a prediction to the other, and then both actions are executed. For `PerAct` we updated the network architecture as well as reduced floating point precision[1] resulting in a significant reduction of the training time. We report on the task success rate as well as on the training time. For simulated experiments, we use two Franka Panda robots with parallel grippers. To demonstrate that our method is robotic agnostic, we also test with the humanoid robot ARMAR-6 [35] in real world. For `ACT` we set $\tilde{o}_{\text{sim}} = \{o_{\text{front}}, o_{\text{wrist left}}, o_{\text{wrist right}}\}$ to minimize the network input.

4.1 Simulation

Method	Task success ↑						
	(a) box	(b) ball	(c) buttons	(d) plate	(e) drawer	(f) fridge	(g) handover
`ACT`	0 %	36 %	4 %	0 %	13 %	0 %	0 %
`RVT-LF`	52 %	17 %	39 %	3 %	10 %	0 %	0 %
`PerAct-LF`	★ 57 %	40 %	10 %	2 %	★ 27 %	0 %	0 %
`PerAct`^{2}	6 %	★ 50 %	★ 47 %	★ 4 %	10 %	★ 3 %	★ 11 %
	(h) laptop	(i) rope	(j) dust	(k) tray	(l) handover easy	(m) oven	
`ACT`	0 %	16 %	0 %	6 %	0 %	2 %	
`RVT-LF`	3 %	3 %	0 %	6 %	0 %	3 %	
`PerAct-LF`	11 %	21 %	★ 28 %	★ 14 %	9 %	8 %	
`PerAct`^{2}	★ 12 %	★ 24 %	0 %	1 %	★ 41 %	★ 9 %	

Table 3: Performance of different methods on various tasks.

We conduct our primary experiments in simulation for reproducibility and benchmarking using the tasks (a) to (m). The environment is similar to [2]. RGB-D sensors are positioned at the front, left shoulder, right shoulder, and on the wrist. All cameras are noiseless and have a resolution of $256\,\text{px} \times 256\,\text{px}$. The increase in image resolution ensures future comparability with other methods that require it.

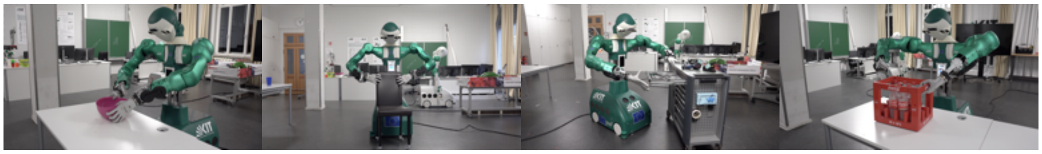
Method	avg. task success ↑	avg. training time ↓
`ACT`	5.9 %	80 h
`RVT-LF`	10.5 %	231 h
`PerAct-LF`	★ 17.5 %	89 h
`PerAct`^{2}	16.8 %	★ 54 h

Table 4: Overview of the average task success rate and average training time with respect to different input image size for 100 demonstrations.

We trained all tasks individually, as this allows for a more refined analysis and different coordination types can be distinguished. We also note that while it is possible to train multi-task agents, not all methods accommodate this setting. We used 100 demonstrations for each task. All single tasks were trained on a NVIDIA A40 GPUs for up to 100k iterations. For each method, the batch size was maximized to fit into the GPU memory. Every 10k-th checkpoint was evaluated with 100 episodes and the best checkpoint was finally evaluated on a separate test set. Tab. 3 lists the task success rate for a single-task agents for each individual task.

The difference in the success rate of image-based methods, such as `ACT` and `RVT` can be explained by the symmetry in the tasks. For example the both robot arms are the exact same model making it difficult to distinguish between them. Furthermore, other challenges for `ACT` include that the demonstrations can have high variations due to the randomization of the spawned objects or the generated motion planning paths. Both aspects are challenging because `ACT` predicts joint angles and not a 6-DoF pose.

[1] https://github.com/ishikasingh/YARR/commit/875f636



(a) lifting a bowl (b) pushing a chair (c) storing away a tool (d) getting a coke

Figure 4: Selected snapshots of the real world experiments showing different tasks.

4.2 Discussion of Failure Types

In the following, we will briefly discuss common failures. During the handover task, a robot may experience a collision with another robot due to inadequate spatial awareness or timing errors. Grasping failures are also common, often due to misalignment of the gripper, leading to an inability to securely grasp the object. For the picking up the plate or the tray tasks, a robot arm may miss the object entirely. This could be a result of insufficient demonstrations or errors in motion planning. Lastly, inserting an item into a drawer introduces complexities such as the requirement for a temporal dependencies: A robot may fail to open the drawer. These failures underscore the challenges in bimanual robotic manipulation when interacting with the scene and the need for more sophisticated motion planning strategies.

4.3 Real-World

To also demonstrated that the framework is robot-agnostic, the method has been integrated into the humanoid robot ARMAR-6 [35, 36]. The robot is equipped with an Azure Kinect RGB-D sensor with a resolution of 1920×1080 and thus for the experiments only a single camera is used, i.e., $\tilde{o}_{\text{real}} = \{o_{\text{front}}\}$. The voxel size is set to $50 \times 50 \times 50$ to speed up training, but reduces accuracy. A Cartesian waypoint controller as used in [37] moves the end-effector to the predicted targets. For each task a single demonstration ζ_i is recorded using Kinesthetic teaching instead of VR. Overall, four different tasks have been demonstrated to the robot. Three of the tasks require a synchronous coordination of both arms, such as *"lifting a bowl"* or *"pushing a chair"*. The fourth task, *"put away the tool"*, requires spatial coordination. While it is possible to quantify results in real world, reproducing them is challenging due to a lack of hardware as well as to object poses and sensor noise. Fig. 4 shows the tasks for the real-world experiments.

5 Conclusion

In this work, we presented a robotic manipulation benchmark specifically designed for bimanual robotic manipulation tasks. We open-source 18 new tasks with 41 unique task variations, each requiring a high degree of coordination and adaptability. We extended two existing methods to bimanual manipulation and run them with another method on the benchmark. Additionally, we presented `PerAct`^{2}– a perceiver-actor agent for bimanual manipulation. Due to the architecture design the method can easily be transferred to other robots, such as a humanoid, as the policy outputs a 6-DoF pose and control is separated. Our investigation reveals that our method is the most successful in 9 out of 13 tasks and performs effectively in real-world settings while also having the fastest training time. For the average task success rate, both `PerAct-LF` and `PerAct`^{2} outperformed image-based methods, achieving average task success rates of $17.5\,\%$ and $16.8\,\%$, respectively.

Limitations and Future Work None of the methods is able to achieve a sufficiently high success rate, which can be explained by the complexity of bimanual manipulation. Another limiting factor is that methods using discretized actions rely on a sampling-based motion planner for successful execution. We acknowledge that tasks in the benchmarks are challenging, and we are looking forward for the community to pick up on this challenge. Future work will focus on extending the benchmark by adding more state-of-the-art methods and including mobile manipulation.

Acknowledgments

This work has been supported by Amazon under the research project ”Robotic Manipulation in Densely Packed Containers”. We would further like to thank Yi Li, Bernie Zhu, Jonas Frey, Aaron Walsman for the insightful suggestions and providing feedback. Moreover, Christian Dreher, Felix Hundhausen and Timo Birr for making the real-world robot experiments possible.

References

[1] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):3019–3026, 2020. doi:10.1109/LRA.2020.2974707.

[2] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation, 2022.

[3] A. Billard and D. Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446): 1–8, 2019. doi:10.1126/science.aat8414.

[4] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic, D. Fox, and A. Cosgun. Deep learning approaches to grasp synthesis: A review, 2022.

[5] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023.

[6] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation, 2024.

[7] L. Fan, Y. Zhu, J. Zhu, Z. Liu, O. Zeng, A. Gupta, J. Creus-Costa, S. Savarese, and L. Fei-Fei. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In A. Billard, A. Dragan, J. Peters, and J. Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 767–782. PMLR, 29–31 Oct 2018. URL `https://proceedings.mlr.press/v87/fan18a.html`.

[8] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning, 2022.

[9] D. Wang, C. Kohler, X. Zhu, M. Jia, and R. Platt. Bulletarm: An open-source robotic manipulation benchmark and learning framework, 2022.

[10] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

[11] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation, 2021.

[12] S. Dasari, J. Wang, J. Hong, S. Bahl, Y. Lin, A. Wang, A. Thankaraj, K. Chahal, B. Calli, S. Gupta, D. Held, L. Pinto, D. Pathak, V. Kumar, and A. Gupta. Rb2: Robotic manipulation benchmarking with a twist. 2022.

[13] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters (RA-L)*, 8(6):3740–3747, 2023. doi:10.1109/LRA.2023.3270034.

[14] Z. Fan, O. Taheri, D. Tzionas, M. Kocabas, M. Kaufmann, M. J. Black, and O. Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[15] Y. Chen, T. Wu, S. Wang, X. Feng, J. Jiang, S. M. McAleer, Y. Geng, H. Dong, Z. Lu, S.-C. Zhu, and Y. Yang. Towards Human-Level Bimanual Dexterous Manipulation with Reinforcement Learning, 2022. URL http://arxiv.org/abs/2206.08686.

[16] K. Chatzilygeroudis, B. Fichera, I. Lauzana, F. Bu, K. Yao, F. Khadivar, and A. Billard. Benchmark for bimanual robotic manipulation of semi-deformable objects. *IEEE Robotics and Automation Letters*, 5(2):2443–2450, 2020. doi:10.1109/LRA.2020.2972837.

[17] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation, 2024.

[18] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic. Dual arm manipulation—A survey. *Robotics and Autonomous Systems*, 60(10):1340–1353, Oct. 2012. ISSN 09218890. doi:10.1016/j.robot.2012.07.005.

[19] K. Zakka, P. Wu, L. Smith, N. Gileadi, T. Howell, X. B. Peng, S. Singh, Y. Tassa, P. Florence, A. Zeng, and P. Abbeel. Robopianist: Dexterous piano playing with deep reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2023.

[20] S. Sina Mirrazavi Salehian, N. Figueroa, and A. Billard. Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty. In *Robotics: Science and Systems XII*. Robotics: Science and Systems Foundation, 2016. doi:10.15607/RSS.2016.XII.019.

[21] V. N. Hartmann, A. Orthey, D. Driess, O. S. Oguz, and M. Toussaint. Long-Horizon Multi-Robot Rearrangement Planning for Construction Assembly. *IEEE Transactions on Robotics*, 39(1):239–252, 2023. ISSN 1552-3098, 1941-0468. doi:10.1109/TRO.2022.3198020.

[22] D. Sirintuna, I. Ozdamar, and A. Ajoudani. Carrying the uncarriable: a deformation-agnostic and human-cooperative framework for unwieldy objects using multiple robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7497–7503, 2023. doi:10.1109/ICRA48891.2023.10160677.

[23] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[24] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen. Robot Cooking With Stir-Fry: Bimanual Non-Prehensile Manipulation of Semi-Fluid Objects. *Robotics and Automation Letters (RA-L)*, 7(2):5159–5166, 2022. ISSN 2377-3766, 2377-3774. doi:10.1109/LRA.2022.3153728.

[25] Y. Zhao, R. Wu, Z. Chen, Y. Zhang, Q. Fan, K. Mo, and H. Dong. Dualafford: Learning collaborative visual affordance for dual-gripper manipulation, 2023.

[26] J. Grannen, Y. Wu, B. Vu, and D. Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation, 2023.

[27] Y. Li, C. Pan, H. Xu, X. Wang, and Y. Wu. Efficient bimanual handover and rearrangement via symmetry-aware actor-critic learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3867–3874, 2023. doi:10.1109/ICRA48891.2023.10160739.

[28] A. . Team, J. Aldaco, T. Armstrong, R. Baruch, J. Bingham, S. Chan, K. Draper, D. Dwibedi, C. Finn, P. Florence, S. Goodrich, W. Gramlich, T. Hage, A. Herzog, J. Hoech, T. Nguyen, I. Storz, B. Tabanpour, L. Takayama, J. Tompson, A. Wahid, T. Wahrburg, S. Xu, S. Yaroshenko, K. Zakka, and T. Z. Zhao. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation, 2024.

[29] I.-C. A. Liu, S. He, D. Seita, and G. Sukhatme. Voxact-b: Voxel-based acting and stabilizing policy for bimanual manipulation, 2024. URL https://arxiv.org/abs/2407.04152.

[30] F. Krebs and T. Asfour. A bimanual manipulation taxonomy. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):11031–11038, 2022. doi:10.1109/LRA.2022.3196158.

[31] S. James and P. Abbeel. Coarse-to-fine q-attention with tree expansion, 2022.

[32] J. J. Gibson. *The ecological approach to visual perception: classic edition*. Psychology Press, 2014. URL https://books.google.com/?id=QReLBQAAQBAJ.

[33] H. P. Moravec. Robot spatial perceptionby stereoscopic vision and 3d evidence grids. *Perception*, 1996. URL https://api.semanticscholar.org/CorpusID:958210.

[34] Y. Roth-Tabak and R. Jain. Building an environment model using depth information. *Computer*, 22(6):85–90, 1989. doi:10.1109/2.30724.

[35] T. Asfour, M. Wächter, L. Kaul, S. Rader, P. Weiner, S. Ottenhaus, R. Grimm, Y. Zhou, M. Grotz, and F. Paus. Armar-6: A high-performance humanoid for human-robot collaboration in real world scenarios. *IEEE Robotics & Automation Magazine*, 26(4):108–121, 2019. doi:10.1109/MRA.2019.2941246.

[36] F. Peller-Konrad, R. Kartmann, C. R. G. Dreher, A. Meixner, F. Reister, M. Grotz, and T. Asfour. A memory system of a robot cognitive architecture and its implementation in armarx. *Robotics and Autonomous Systems*, 164:1–20, 2023. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2023.104415.

[37] R. Grimm, M. Grotz, S. Ottenhaus, and T. Asfour. Vision-based robotic pushing and grasping for stone sample collection under computing resource constraints. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6498–6504, 2021. doi: 10.1109/ICRA48506.2021.9560889.

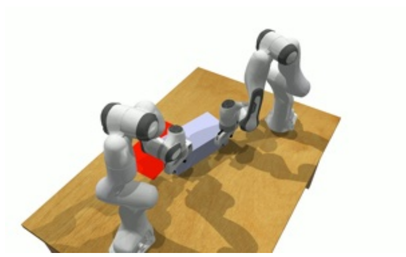
Task	duration	# keyframes	# items	# variations
(a) push box	4.33 s	2.1	1	1
(b) lift a ball	4.40 s	4.0	1	1
(c) push two buttons	3.47 s	4.0	3	5
(d) pick up a plate	6.47 s	6.6	1	1
(e) put item in drawer	5.57 s	8.4	5	3
(f) put bottle in fridge	9.70 s	7.8	2	1
(g) handover an item	7.63 s	7.6	5	5
(h) pick up notebook	3.97 s	7.2	1	1
(i) straighten rope	3.83 s	5.9	1	1
(j) sweep dust pan	4.93 s	7.3	1	1
(k) lift tray	3.77 s	5.1	1	1
(l) handover item (easy)	7.17 s	7.5	1	1
(m) take tray out of oven	10.13 s	8.7	2	1
(n) take shoes out of box	14.67 s	18.8	3	1
(o) lift a cube	4.2 s	6.0	1	1
(p) open drawer	3.63 s	4.0	4	3
(q) sort shapes	7.9 s	6.0	7	10
(r) shell game	6.3 s	7.0	4	3

Table 5: Properties of the bimanual tasks. We report on the average length of the task demonstration in seconds. The average number of extracted keyframes of the task, the number of items that the robot can interact with and the task variations.

A Description of Tasks

To model the complexity of a task we report in Tab. 5 on the average time length of the demonstrations, the number of identified keyframes, number of items and the number of task variations Keyframes are discrete frames in a continuous stream of data, and the number of keyframes is a measure of the number of actions necessary to complete a task.

(a) push box



Task Description: The robot's task is to push a heavy box using both arms to move it to a designated target area.

Success Metric: The task is considered successfully completed when the box reaches the targetarea.

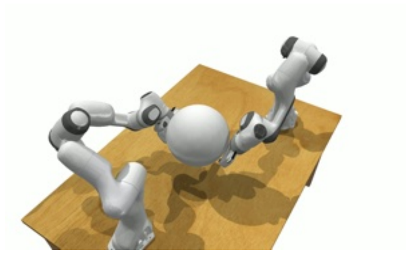
Objects: The task involves a large box and a target area.

Coordination Challenges: The primary challenge lies in the weight of the box, which is set to $50\,\mathrm{kg}$, making it very difficult for a single arm to push.

NB: This task cannot be solved with one robot due to the weight of the box.

Language Instructions: *Push the box to the red area.*

(b) lift a ball



Task Description: The robot's task is to grasp and lift a large ball using both arms.

Success Metric: The task is considered successfully completed when the ball is lifted to a height above $0.95\,\mathrm{m}$.

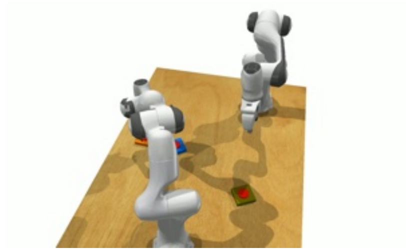
Objects: The task involves a large ball.

Coordination Challenges: The primary challenge involves coordinated non-prehensile manipulation, as the ball cannot be grasped by the gripper. This requires careful coordination during the lifting motion.

NB: This task is impossible to solve with one robot due to the size of the object.

Language Instructions: *Lift the ball.*

(c) push two buttons



Task Description: The robot's task is to push two out of three buttons in an environment where the colors of the buttons are randomized. The goal is to press two specified buttons at the same time.

Success Metric: The task is considered successfully completed when both specified buttons are pressed simultaneously.

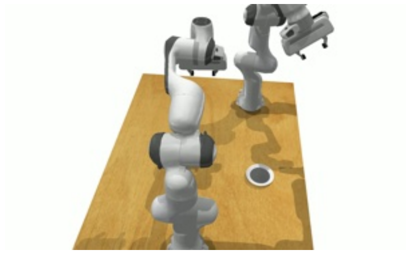
Objects: The task involves three buttons with different colors and a differently colored base.

Coordination Challenges: The primary challenge is the synchronous button press. The randomization of the button colors adds complexity compared to standard tasks.

NB: This task is impossible to solve with one robot as two buttons need to be pressed simultaneously.

Language Instructions: *Push the (color A) and the (color B) button.*

(d) pick up a plate



Task Description: The robot's task is to pick up a plate that is placed on a table. This involves grasping the plate and lifting it.

Success Metric: The task is considered successfully completed when the robot has securely grasped the plate and lifted it.

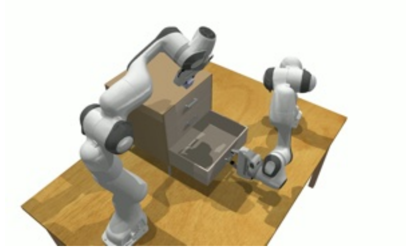
Objects: The task involves a single plate.

Coordination Challenges: The main challenges involve non-prehensile manipulation, as well as coordination during the lifting motion. The plate must be handled delicately to avoid slipping or tilting, which requires precise control.

NB: This task is difficult to solve with one robot because the plate's flat and smooth surface makes it hard to grasp securely with a single gripper. The coordination required to lift the plate without tilting or dropping it is challenging for one robot arm.

Language Instructions: *Pick up the plate.*

(e) put item in drawer



Task Description: The robot's task is to open a specific drawer in a cupboard and place an item into it. The correct drawer must be identified and opened before the item can be placed inside.

Success Metric: The task is considered successfully completed when the item is placed inside the specified drawer.

Objects: The task involves an item and a cupboard with three drawers.

Coordination Challenges: The primary challenge involves identifying the correct drawer and ensuring it is open before attempting to place the item inside. This requires coordination between the actions of opening the drawer and placing the item.

Objects: A cupboard with three drawers and an item on top

Language Instructions: *Put the item into the (top, middle, bottom) drawer.*

(f) put bottle in fridge

Task Description: The robot's task is to put a bottle into the fridge. This requires opening the fridge door, grasping the bottle, and placing it inside the fridge.

Success Metric: The task is considered successfully completed when the bottle is placed inside the fridge.

Objects: The task involves a bottle and a fridge.

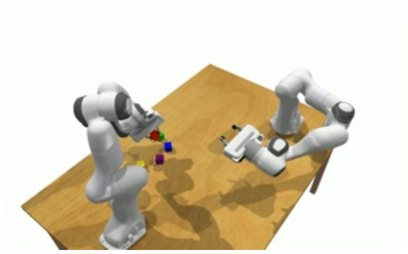
Coordination Challenges: The primary challenges include: - The fridge needs to be opened first. - The bottle is difficult to grasp. - Collision with the fridge needs to be avoided. - Reachability is an issue as either the bottle or the fridge door can only be reached by one robot.

Objects: A bottle and a fridge

NB: This task requires two robots due to reachability issues.

Language Instructions: *Put the bottle into the fridge.*

(g) handover an item



Task Description: The robot's task is to hand over the *color* item. This involves identifying the correct item based on its color, grasping it, and lifting it to the required height.

Success Metric: The task is considered successfully completed when the robot has securely grasped the correct item and lifted it to a height of $80\,\mathrm{cm}$, while the other arm remains idle.

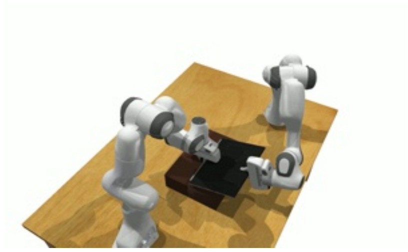
Objects: The task involves five items with different colors.

Coordination Challenges: The main challenge lies in correctly identifying the item based on its color as specified in the task description, and then coordinating the handover process.

NB: There are variations of this task: one with only three items instead of five, and a simpler task that involves only a block instead of colored cubes.

Language Instructions: *Hand over the (red, green, blue, yellow) item.*

(h) pick up notebook



Task Description: The robot's task is to pick up a notebook that is placed on top of a block. This requires the robot to first manipulate the notebook into a position where it can be grasped.

Success Metric: The task is considered successfully completed when the robot has securely grasped the notebook and lifted it off the block.

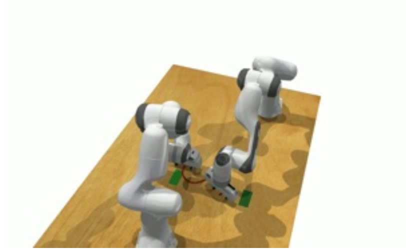
Coordination Challenges: Since the notebook is resting on a block, the robot must perform non-prehensile manipulation, such as pushing or sliding, to reposition the notebook into a graspable orientation.

Objects: The task involves two primary objects: a notebook and a block.

NB: This task can be accomplished with a single robotic arm, though coordination is crucial for successful manipulation.

Language Instructions: *pick up the notebook*

(i) straighten rope



Task Description: The robot's task is to straighten a rope by manipulating it so that both ends are placed into distinct target areas.

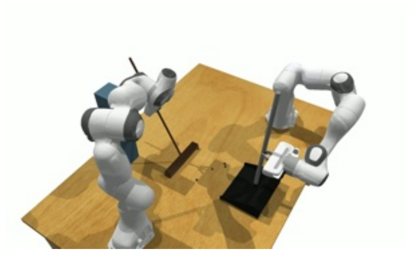
Success Metric: The task is considered successfully completed when both ends of the rope are positioned within their respective target areas.

Objects: The task involves a single object: a rope.

Coordination Challenges: The main challenge involves handling a deformable object, which requires the robot to grasp and manipulate the rope simultaneously at different points to achieve the desired straightening.

Language Instructions: *Straighten the rope.*

(j) sweep dust pan



Task Description: The robot's task is to sweep the dust into the dust pan using a broom. This involves coordinating the sweeping motion to ensure the dust is effectively collected.

Success Metric: The task is considered successfully completed when all the dust is inside the dust pan.

Objects: The task involves several objects: a broom, a dust pan, supporting objects, and dust.

Coordination Challenges: The main challenge lies in executing the sweeping motion accurately to ensure that the dust is directed into the dust pan.

Language Instructions: *Sweep the dust to the pan.*

(k) lift tray

Task Description: The robot's task is to lift a tray that is placed on a holder. An item is on top of the tray and must be balanced while both arms lift the tray.

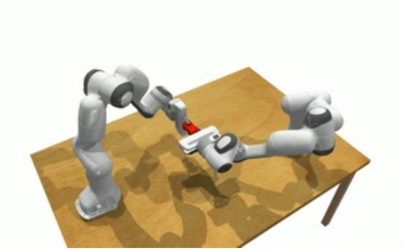
Success Metric: The task is considered successfully completed when both the tray and the item on top reach a height above $1.2\,\mathrm{m}$.

Objects: The task involves a tray, a holder, and an item.

Coordination Challenges: The primary challenge lies in coordinating the lifting motion with both arms to maintain the balance of the item on the tray. This task cannot be accomplished with only one arm.

Language Instructions: *Lift the tray*

(l) handover item (easy)



Task Description: The robot's task is to hand over a red item. One robotic arm must grasp the red item while the other arm remains free and wait for the handover

Success Metric: The task is considered successfully completed when the robot has securely grasped the correct item and lifted it to a height of $80\,\mathrm{cm}$, while the other arm remains idle and has not grasped anything.

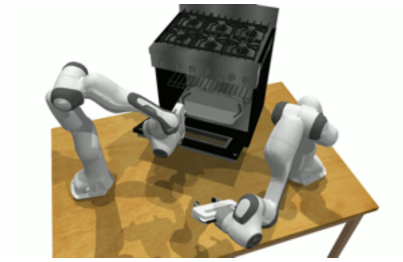
Objects: The task involves a red block.

Coordination Challenges: The primary challenge lies in coordinating the handover process.

NB: There is also a more complex variant of this task that involves handling multiple objects of different shapes and sizes. Refer to the "handover item" task for details.

Language Instructions: *Handover the item.*

(m) take tray out of oven



Task Description: The robot's task is to remove a tray that is located inside an oven. This involves opening the oven door and then grasping the tray.

Success Metric: The task is considered successfully completed when the tray is lifted above the oven.

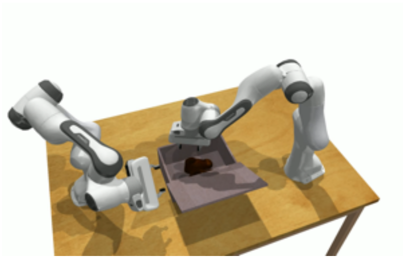
Objects: The task involves a tray inside an oven.

Coordination Challenges: The primary challenge lies in opening the oven door to make the tray graspable.

NB: This task can be solved with only one arm.

Language Instructions: *Take tray out of oven.*

(n) take shoes out of box



Task Description: The robot's task is to open a shoe box and take both shoes out of the box.

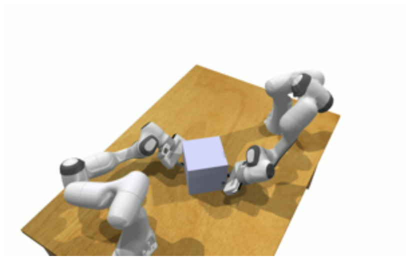
Success Metric: The task is considered successfully completed if both shoes have been placed in a target area.

Objects: The task involves a show box, containing two shoes. a.

Coordination Challenges: The primary challenge lies in the long horizon planning. Additionally there is a temporal dependency, as the lid of the shoe box needs to be opened first.

Language Instructions: *Take the shoes out of the box.*

(o) lift a cube



Task Description: The robot's task is to grasp and lift a cube using both arms.

Success Metric: The task is considered successfully completed when the cube is lifted to a height above $0.95\,\mathrm{m}$.

Objects: The task involves a large cube.

Coordination Challenges: The primary challenge is that the cube is not graspable with a single gripper. Hence the robot needs to exert force with both gripper towards the center in order to lift the cube. Morever, this requires careful coordination during the lifting motion.

NB: This task is impossible to solve with one robot due to the size of the object. Furthermore, see *(b) lift ball*.

Language Instructions: *Lift the cube.*

(p) open drawer

Task Description: The robot's task is to open a specific drawer using either the left or right arm based on the task.

Success Metric: The task is considered successfully completed when the drawer is fully opened using the specified arm.

Objects: The task involves a single drawer.

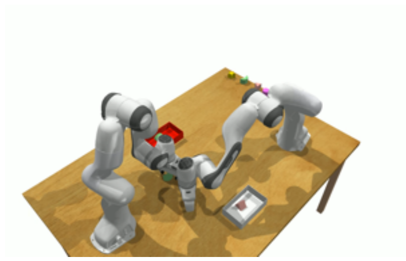
Coordination Challenges: The main challenge involves using the specified hand to open the drawer, which requires precise control and coordination. Depending on the drawer's position and the robot's configuration, reaching and applying the necessary force to open the drawer might be difficult.

Objects: A drawer

NB: Another variation of this task requires using the other hand to open the drawer.

Language Instructions: *Open the (top,middle,bottom) drawer.*

(q) Sort shapes



Task Description: The robot is tasked with sorting two distinct types of items. Two item categories are selected out of five. Two items from each category are randomly placed in front of the robot, and two sorting containers are positioned within its reach. The robot must correctly identify each item and place it into the appropriate container based on its category.

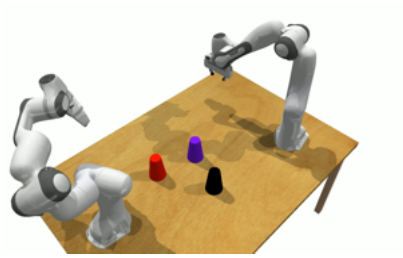
Success Metric: The task is deemed successful when the two types of items are correctly separated, with each sorting container holding only one category of items.

Objects: Five different types of items and two sorting containers.

Coordination Challenges: The main challenge is for the robot to effectively use both arms simultaneously to optimize the efficiency of the sorting process.

Language Instructions: *Sort the items according to their shape.*

(r) Shell game



Task Description: The robot has to hide an item under a cup woth one arm and the reveal it with the other arm.

Success Metric: The task is considered successful when the item was first hidden under a cup and then revealed.

Objects: Three cups and one item.

Coordination Challenges: The primary challenge is that the agent has to reason about occlusion.

Language Instructions: *Find the item.*