Robotics: Science and Systems 2024
Delft, Netherlands, July 15-July 19, 2024

HumanoidBench: Simulated Humanoid Benchmark for Whole-Body Locomotion and Manipulation

Carmelo Sferrazza[1] Dun-Ming Huang[1] Xingyu Lin[1] Youngwoon Lee[1,2] Pieter Abbeel[1]
UC Berkeley[1] Yonsei University[2]

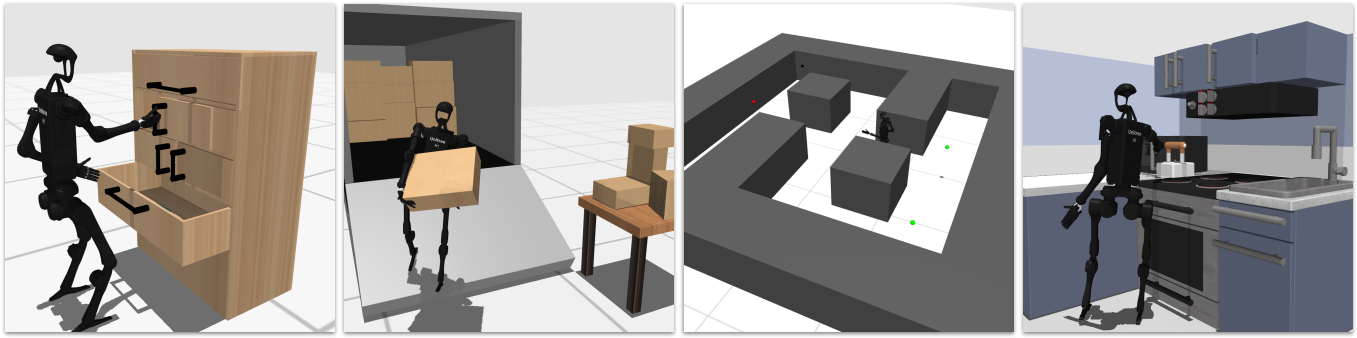


Fig. 1: Humanoid robots equipped with dexterous hands hold immense promise for integration into real-world human environments. Nonetheless, harnessing the full potential of humanoid robots presents numerous challenges, such as the intricate control of robots with complex dynamics, sophisticated coordination among various body parts, and addressing long-horizon complex tasks envisioned for these robots. We present **HumanoidBench**, a simulated humanoid robot benchmark consisting of 15 whole-body manipulation and 12 locomotion tasks, such as shelf rearrangement, package unloading, and maze navigation.

***Abstract*—Humanoid robots hold great promise in assisting humans in diverse environments and tasks, due to their flexibility and adaptability leveraging human-like morphology. However, research in humanoid robots is often bottlenecked by the costly and fragile hardware setups. To accelerate algorithmic research in humanoid robots, we present a high-dimensional, simulated robot learning benchmark, HumanoidBench, featuring a humanoid robot equipped with dexterous hands and a variety of challenging whole-body manipulation and locomotion tasks. Our findings reveal that state-of-the-art reinforcement learning algorithms struggle with most tasks, whereas a hierarchical learning baseline achieves superior performance when supported by robust low-level policies, such as walking or reaching. With HumanoidBench, we provide the robotics community with a platform to identify the challenges arising when solving diverse tasks with humanoid robots, facilitating prompt verification of algorithms and ideas. The open-source code is available at https://humanoid-bench.github.io.**

I. Introduction

Humanoid robots have long held promise to be seamlessly deployed in our daily lives. Despite the rapid progress in humanoid robots' hardware (e.g., Boston Dynamics Atlas, Tesla Optimus, Unitree H1), their controllers are fully or partially hand-designed for specific tasks, which requires significant engineering efforts for each new task and environment, and often demonstrates only limited whole-body control capabilities.

In recent years, robot learning has shown steady progress in both robotic manipulation [12, 69, 15] and locomotion [27, 71]. However, scaling learning algorithms to humanoid robots is still challenging and has been delayed mainly due to such robots' costly and unsafe real-world experimental setups.

To accelerate the progress of research for humanoid robots, we present the first-of-its-kind humanoid robot benchmark, **HumanoidBench**, with a diverse set of locomotion and manipulation tasks. Our simulated humanoid benchmark demonstrates a variety of challenges in addressing learning for autonomous humanoid robots, such as the intricate control of robots with complex dynamics, sophisticated coordination among various body parts, and addressing long-horizon complex tasks, while providing an accessible, fast, safe, and inexpensive testbed to robot learning researchers.

HumanoidBench provides (1) a simulation environment comprising a humanoid robot with two dexterous hands, as illustrated in Figure 1; (2) a variety of tasks, spanning locomotion, manipulation, and whole-body control, incorporating humans' everyday tasks; (3) a standardized benchmark to evaluate the progress of the community on high-dimensional humanoid robot learning and control. In fact, HumanoidBench supports generic controller structures, including both learning and model-based approaches [14, 26]. In this paper, we present extensive benchmarking results of the state-of-the-art reinforcement learning (RL) algorithms, which do not require extensive domain knowledge, and a hierarchical RL approach.

The simulation environment of HumanoidBench uses the MuJoCo [60] physics engine. For the simulated humanoid robot,

Benchmark	Dexterous hands	Action dim.	DoF	Task horizon	# Tasks	Skills[1]
MyoHand [8]	✔	39	23D	50-2000	9	PnP, R, Po, IR, H, Ro
Adroit [49]	✔	24	24D	200	4	PnP, P, R, Po, IR, H, L, Ro
MyoLeg [8]	✖	80	20D	1000	1	Lo, St
LocoMujoco [3] (Unitree-H1)	✖	19	6D	100-500	27	L, Lo, St, BM
DMControl [58] (Humanoid)	✖	24-56	22D	1000	6	Lo, St
FurnitureSim [22]	✖	8	6D	2300	8	PnP, P, I, IR, H, L, Ro
robosuite [70]	✖	6-24	6-7D	500	9	PnP, P, I, R, IR, H, L, Ro
rlbench [24]	✖	6-7	6-7D	100-1000	106	PnP, P, I, R, Po, IR, H, L, Ro
metaworld [64]	✖	6	7D	500	50	PnP, P, I, R, Po, IR, H, L, Ro
HumanoidBench (Ours)	✔	61	75D	500-1000	27	PnP, P, I, R, Po, IR, H, L, Ro, Lo, BM, St

[1]PnP: Pick-and-place / P: Push / I: Insert / R: Reach / Po: Pose / IR: In-hand re-orientation / H: Hold / L: Lift / Ro: Rotate / Lo: Locomotion / BM: Whole-body (humanoid) Manipulation / St: Stabilization

TABLE I: **Comparison of simulated robot benchmarks.** Our humanoid robot benchmark tests a variety of complex, long-horizon task with a large action space.

we mainly opt for a Unitree H1 humanoid robot[1], which is relatively affordable and offers accurate simulation models [66], with two dexterous Shadow Hands[2] attached to its arms. Our environment can easily incorporate any humanoid robots and end effectors; thus, we provide other models, including Unitree G1[3], Agility Robotics Digit[4], the Robotiq 2F-85 gripper, and the Unitree H1 hand.

The HumanoidBench task suite includes 15 distinct whole-body manipulation tasks involving a variety of interactions, e.g., unloading packages from a truck, wiping windows using a tool, catching and shooting a basketball. In addition, we provide 12 locomotion tasks (not requiring hands' dexterity), which can serve as primitive skills for whole-body manipulation tasks and provide a set of easier tasks to verify algorithms. The benchmarking results on this task suite show how the state-of-the-art RL algorithms struggle with controlling the complex humanoid robot dynamics and solving the most challenging tasks, illustrating ample opportunities for future research.

II. Related Work

Deep reinforcement learning (RL) has made rapid progress with the advent of standardized, simulated benchmarks, such as Atari [5] and continuous control [7, 58] benchmarks. In robotic manipulation, most existing simulated environments are limited to quasi-static, short-horizon skills, having focused on tasks like picking and placing [7, 24, 70, 64, 37], in-hand manipulation [49, 44, 8], and screwing [43].

Complex manipulation tasks, such as block stacking [13], kitchen tasks [17], and table-top manipulation [25, 39, 34], have been introduced but are still limited to a combination of pushing, picking, and placing. On the other hand, the IKEA furniture assembly environment [31], BEHAVIOR [55, 32], and Habitat [57] present diverse long-horizon (mobile) manipulation tasks, with their main focus being on high-level planning by abstracting complex low-level control problems, while FurnitureBench [22] introduces a simulated benchmark for complex long-horizon furniture assembly tasks with sophisticated low-level control. However, most of these benchmarks use a single-arm manipulation setup with either a parallel gripper or a dexterous hand [9, 49], limiting the types of object interactions and not addressing the challenges of coordinating multiple parts of a body [30], e.g., multiple fingers, arms, and legs.

Robosuite [70] includes a handful of bimanual manipulation tasks, while more recently [10] and [67] have introduced additional benchmarks that require coordinating two floating robot hands, i.e., not attached to any arm base. While bimanual manipulation is one of the key objectives of humanoid robots, most benchmarks in humanoid research have so far focused on the locomotion challenges of such platforms [8, 28, 45, 3]. In this regard, such simulations have accelerated research on control algorithms [6, 46, 47, 40], ultimately leading to achieve robust humanoid locomotion in the real world [1, 50, 11].

Recent works have extended humanoid simulations to different domains involving a certain degree of manipulation, i.e., tennis [68], soccer [19], ball manipulation [61] and catching [38], and box moving [63]. However, all these works focus on demonstrating their approaches on specific humanoid tasks and lack a diversity of tasks. In addition, most of the previous work focuses on simplistic humanoid models [38, 61], leading to inaccurate physics and collision handling. This motivates us to implement a *comprehensive* simulated humanoid benchmark based on real-world hardware and consisting of a diverse set of whole-body control tasks with careful design choices for diversity and usability.

In contrast to prior robotic simulation benchmarks, HumanoidBench presents a broader set of challenges, featuring high-dimensional action spaces and DoFs resulting from humanoid robots and dexterous hands, and a variety of long-horizon tasks, which cover a comprehensive set of robotic locomotion and manipulation skills, as summarized in Table I.

Finally, we note how in the literature, tasks that require long-term planning with a high-dimensional action space have been addressed with hierarchical reinforcement learning (HRL), which decouples low-level and high-level planning in a reinforcement learning paradigm setting [33, 4, 42, 29, 17, 30, 48]. In the context of humanoids, we propose an HRL paradigm

[1]https://www.unitree.com/h1
[2]https://www.shadowrobot.com/dexterous-hand-series/
[3]https://www.unitree.com/g1
[4]https://agilityrobotics.com/robots

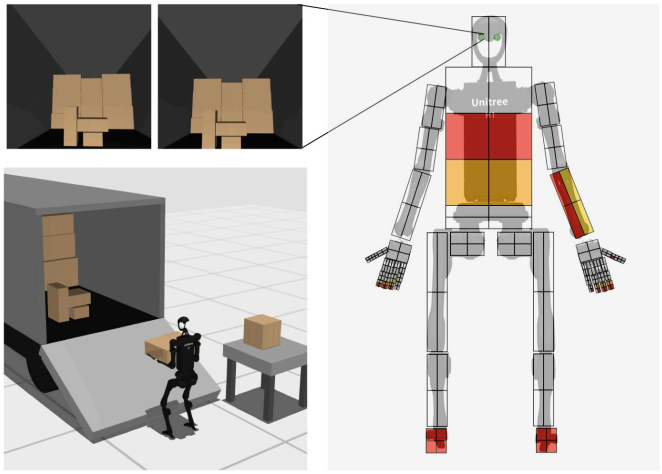


Fig. 2: Example egocentric visual (top-left) and whole-body tactile (right) observations when the humanoid interacts with a package in the `truck` environment. In the right figure, the two cameras on the robot head are highlighted in green, while continuous tactile pressure readings are indicated in shades of red (strong pressure) and yellow (mild pressure). Note that here, for ease of visualization, we are not showing shear forces and tactile readings on the back of the robot, which are also implemented in our environment.

to show how a specific set of low-level skills (e.g., standing, walking) facilitates learning of higher level tasks.

III. Simulated Humanoid Robot Environment

In this section, we describe our simulated environment and discuss relevant design choices for the simulated humanoid robot. As illustrated in Figure 2, we use the Unitree H1 humanoid robot[1] with two dexterous Shadow Hands[2] as the primary robotic agent of our benchmark. We simulate this humanoid robot using MuJoCo [60] adapting the Unitree H1 model provided by Unitree[5] and the dexterous Shadow Hand models available through MuJoCo Menagerie.[6]

Humanoid Body. We implement Unitree H1[1], Unitree G1[3], and Agility Robotics Digit[4], which are well-known humanoid robots with their model files freely available [66, 1]. Unitree H1 is primarily used in our benchmark as it is a full-size humanoid compared to the smaller Unitree G1, and as we observed faster learning compared to the Agility Robotics Digit, which we ascribe to a simpler mechanical design compared to Digit, which features passive joints actuated through a four-bar linkage.

Dexterous Hands. We use two dexterous Shadow Hands[2], which also have model files freely available[6], and have shown impressive manipulation capabilities both in simulation [67] and in the real world [2]. To make the simulated robot have more human-like morphology, we remove the cumbersome forearms of the dexterous Shadow Hands in HumanoidBench.

[5]https://github.com/unitreerobotics/unitree_ros
[6]https://github.com/google-deepmind/mujoco_menagerie

	Without hand	With 2 hands
Observation space	51	151
Action space	19	61
DoF (body)	25	25
DoF (two hands)	0	50

TABLE II: **Humanoid robot specifications with and without hands.** Both the humanoid body (including its floating base) and one Shadow Hand present action spaces (19 and 21, respectively) smaller than their DoFs (25), making them under-actuated systems. In this table, the observation spaces solely comprise generalized positions and velocities of the robots and do not take into account any environment observations. We use quaternions for the robot floating base orientation, which adds an additional position coordinate compared to the velocity components, which match the DoFs. In the appendix, Table III shows an exhaustive overview of all the robot configurations available in HumanoidBench.

While this is not currently a realistic model, we anticipate the trend in the industry towards developing slimmer, human-like hands (e.g., Tesla Optimus, Figure 01) so that our design choice aligns better with next-generation humanoid robots. In addition, we also provide models for the Robotiq 2F-85 parallel-jaw gripper and the 13-DoF Unitree hands available in the Unitree collection[5] (see Appendix, Section A for more details).

The observation and action spaces, and degrees of freedom of the robot system with or without the dexterous hands are summarized in Table II.

Observations. Our simulated environment supports the following observations:

- Proprioceptive robot state (i.e., joint angles and velocities) and task-relevant environment observations (i.e., object poses and velocities).
- Egocentric visual observations from two cameras placed on the robot head (see Figure 2).
- Whole-body tactile sensing using the MuJoCo tactile grid sensor (see Figure 2). We design tactile sensing at the hands with high resolution and in other body parts with low resolution, similar to humans, with a total of 448 taxels spread over the entire body, each providing three-dimensional contact force readings. Similar distributed force readings have been captured on real-world systems both on humanoid bodies [41] and end-effectors [53]. The implementation of such spatially distributed contact sensing required non-trivial mesh adaptations and refinements, which we detail in the appendix.

Although other sensory inputs are available from the environment, to investigate challenges in whole-body control of humanoid robots, we first focus on the state-based environment setup, where proprioceptive robot states and object states are used as the agent's input in HumanoidBench. In our state-based environment, we maintain the robot observations the same across tasks to minimize domain knowledge, in contrast to tailoring it to the specific tasks [59]. We leave extending our environment to benchmarking multimodal perception

Fig. 3: **HumanoidBench manipulation task suite.** We devise 15 benchmarking whole-body manipulation tasks that cover a wide variety of interactions and difficulties. This figure illustrates an initial state for each task (left) and examples of the robot performing such tasks (right).

capabilities [65, 54] of humanoid robots as future work.

Actions. In HumanoidBench, the humanoid robot is controlled via position control (i.e., specifying the target joint positions). Torque-based control is also supported but we found that position control is generally more stable and allows for lower control frequency than torque control. For both position and torque control, the action space is 61-dimensional including the two hands, and controlled at 50 Hz.

IV. HumanoidBench

Humanoid robots promise to solve human-like tasks in human-tailored environments, possibly using human tools. However, their form factor and hardware challenges make real-world research challenging, making simulation a crucial tool to advance algorithmic research in the field.

To this end, we present HumanoidBench, a humanoid benchmark for robot learning and control, which features a

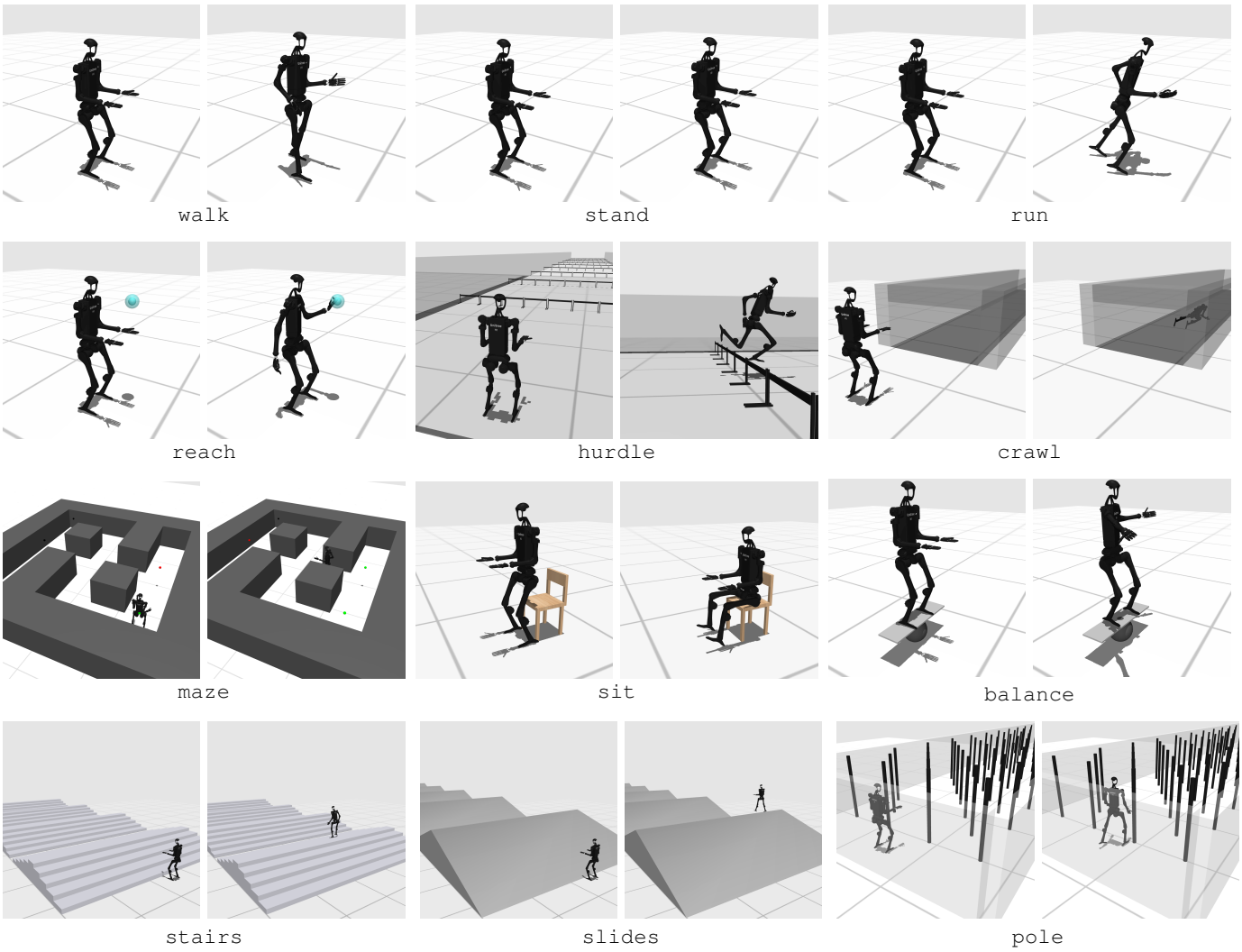


Fig. 4: **HumanoidBench locomotion task suite.** We devise 12 benchmarking locomotion tasks that cover a wide variety of interactions and difficulties. This figure illustrates an initial state for each task (left) and examples of the robot performing such tasks (right).

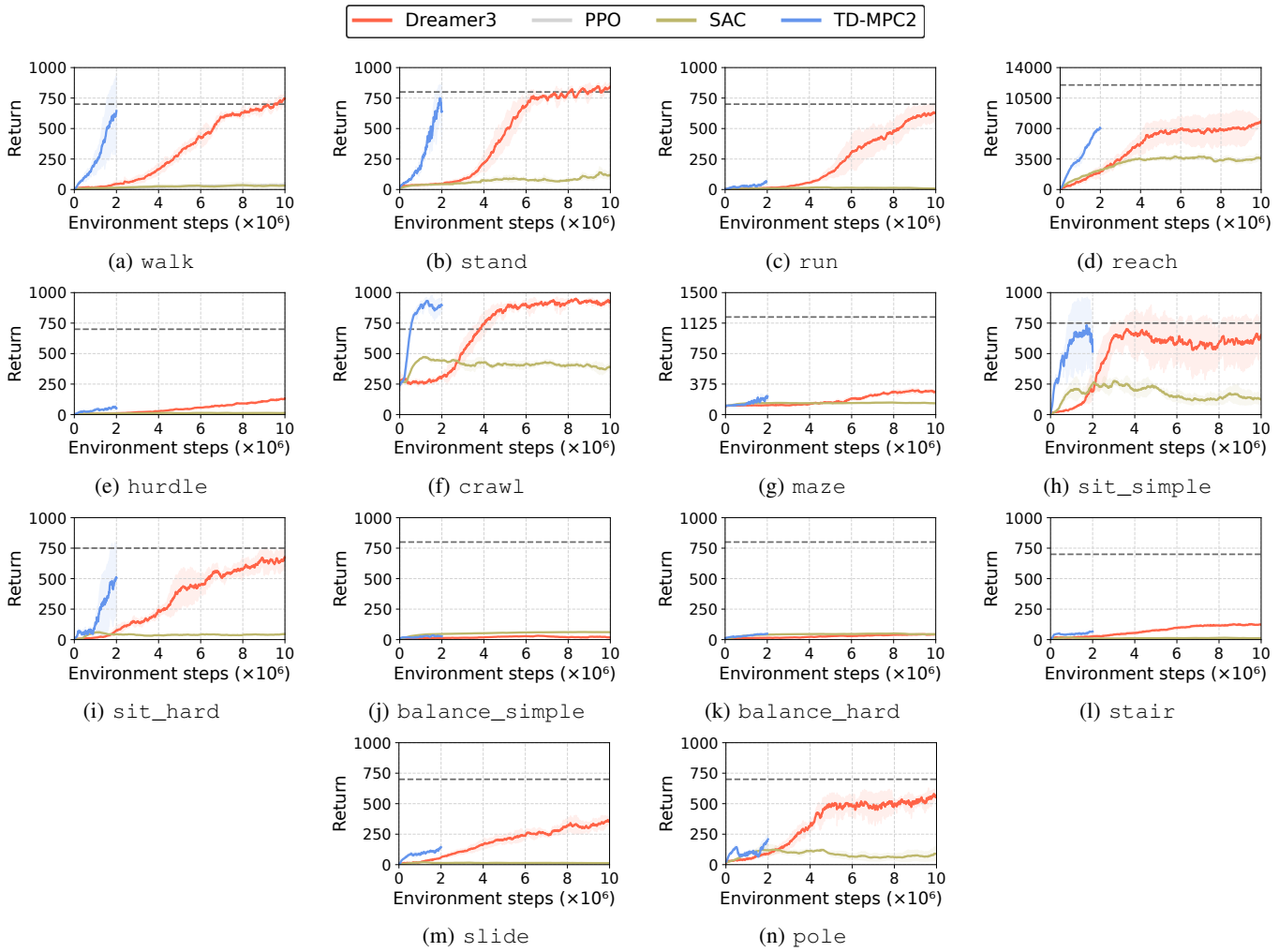
high-dimensional action space (up to 61 different actuators) and enables research in complex whole-body coordination.

We benchmark 27 tasks, consisting of 12 locomotion tasks and 15 distinct manipulation tasks, as illustrated in Figure 4 and Figure 3. A set of locomotion tasks aim to provide interesting but simpler humanoid control scenarios, bypassing intricate dexterous hand control. On the other hand, whole-body manipulation tasks render a comprehensive evaluation of the state-of-the-art algorithms on challenging tasks with unique challenges that require coordination across the entire robot body, ranging from toy examples (e.g., pushing a box on a table) to practical applications (e.g., truck unloading, shelf rearrangement).

Below we briefly describe the tasks that are part of the benchmark. Further details about each of the tasks, including task initialization and reward functions, are provided in Appendix, Section B-E.

A. *Locomotion Tasks*

- `walk`: Keep forward velocity close to $1\,\mathrm{m/s}$ without falling to the ground.
- `stand`: Maintain a standing pose throughout the provided amount of time.
- `run`: Run forward (in the global x-direction) at a speed of $5\,\mathrm{m/s}$.
- `reach`: Reach a randomly initialized 3D point with the left hand.
- `hurdle`: Keep forward velocity close to $5\,\mathrm{m/s}$ while successfully overcoming hurdles.
- `crawl`: Keep forward velocity close to $1\,\mathrm{m/s}$ while passing inside a tunnel.
- `maze`: Reach the goal position in a maze by taking multiple turns at the intersections.
- `sit`: Sit onto a chair situated closely behind the robot.
- `balance`: Stay balanced on the unstable board.



(a) walk (b) stand (c) run (d) reach

(e) hurdle (f) crawl (g) maze (h) sit_simple

(i) sit_hard (j) balance_simple (k) balance_hard (l) stair

(m) slide (n) pole

Fig. 5: **Learning curves of RL algorithms (locomotion).** The curves are averaged over three random seeds and the shaded regions represent the standard deviation. Returns are computed by summing the rewards at all timesteps of an episode. The dashed lines qualitatively indicate task success. We run PPO on the `walk` task but it is not visible in the plot since it only achieves very low returns.

- `stair`: Traverse an iterating sequence of upward and downward stairs at $1\,\mathrm{m/s}$.
- `slide`: Walk over an iterating sequence of upward and downward slides at $1\,\mathrm{m/s}$.
- `pole`: Travel in forward direction over a dense forest of high thin poles, without colliding with them.

B. Whole-Body Manipulation Tasks

- `push`: Move a box to a randomly initialized 3D point on a table.
- `cabinet`: Open four different types of cabinet doors (e.g., hinge doors, sliding door, drawer).
- `highbar`: Athletically swing while staying attached to a horizontal high bar until reaching a vertical upside-down position.
- `door`: Pull a door and traverse it while keeping the door open.
- `truck`: Unload packages from a truck by moving them onto a platform.
- `cube`: Manipulate two cubes in-hand until they both reach a randomly initialized target orientation.
- `bookshelf`: Pick and place several items across shelves in a given order.
- `basketball`: Catch a ball coming from random directions and throw it into the basket.
- `window`: Grab a window wiping tool and keep its tip parallel to a window by following a prescribed vertical velocity.
- `spoon`: Grab a spoon and use it to follow a circular pattern inside a pot.
- `kitchen` [17]: Execute a sequence of actions in a kitchen environment, namely, open a microwave door, move a kettle, and turning burner and light switches.
- `package`: Move a box to a randomly initialized target position.
- `powerlift`: Lift a barbell shaped object of a designated

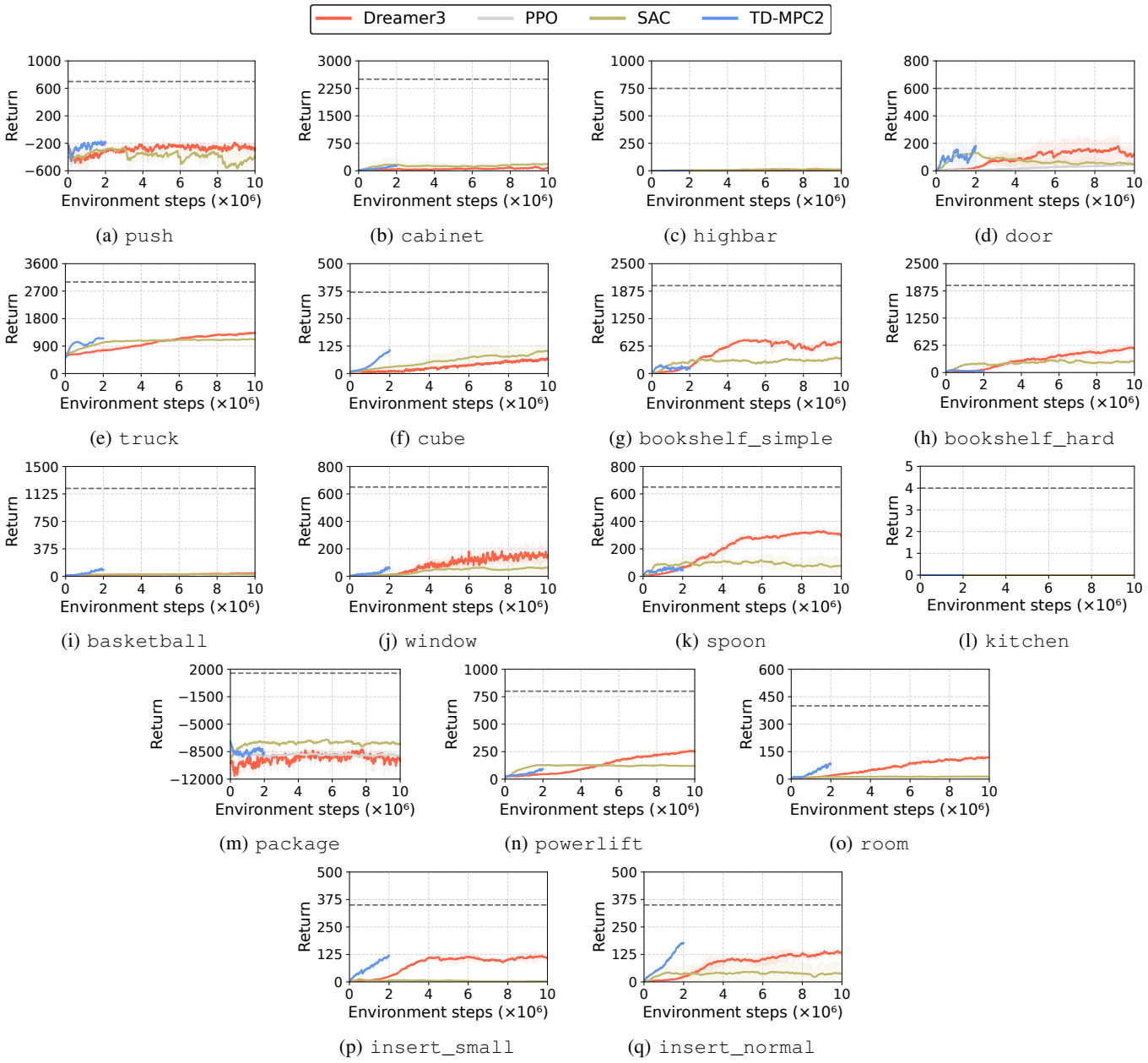


Fig. 6: **Learning curves of RL algorithms (manipulation).** The curves are averaged over three random seeds and the shaded regions represent the standard deviation. The dashed lines qualitatively indicate task success. Note that `kitchen` is the only environment with a purely discrete, sparse reward, with a maximum of 4.

mass.

- `room`: Organize a 5 m by 5 m space populated with randomly scattered object to minimize the variance of scattered objects' locations in x, y-axis directions.
- `insert`: Insert the ends of a rectangular peg into two tight target blocks.

V. Benchmarking Results

To identify the challenges in learning with humanoid robots, we benchmark reinforcement learning (RL) algorithms on HumanoidBench, which promises for robots to learn from their own experience. Remarkably, this class of algorithms requires limited domain expertise and does not necessarily rely on expert demonstrations, which are not only expensive but also challenging to collect for humanoid robots.[7]

A. Baselines

We evaluate all tasks in our benchmark with four RL methods (DreamerV3, TD-MPC2, SAC, PPO). Please refer to Appendix, Section C for implementation details.

[7]While we do not benchmark classical model-based control approaches [14, 26] in this work, our environments support actions obtained by using any type of controllers.

- **DreamerV3** [20]: the state-of-the-art model-based RL algorithm, learning from imaginary model rollouts.
- **TD-MPC2** [21]: the state-of-the-art model-based RL algorithm with online planning.
- **SAC** (Soft Actor-Critic [18]): the state-of-the-art off-policy model-free RL algorithm.
- **PPO** (Proximal Policy Optimization [52]): the state-of-the-art on-policy model-free RL algorithm.

B. Results

We report benchmarking results in Figure 5 and Figure 6, where we ran each of the algorithms for approximately 48 hours, resulting in the visible differences in environment steps (e.g., 2M steps for TD-MPC2, 10M steps for DreamerV3). We only run PPO on a subset of tasks (`walk`, `kitchen`, `door`, `package`), given its inferior performance without massive parallelization. Each of the environments is evaluated with a combination of dense rewards and sparse subtask completion rewards, and for each of these we provide qualitative measures of task success (see dashed lines in Figure 5 and Figure 6). A detailed description of the reward functions used for each environment is available in Appendix, Section B.

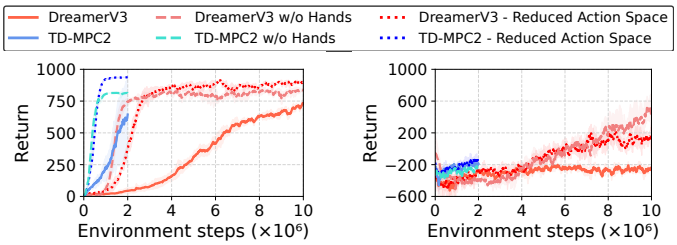
All the baseline algorithms perform below the success threshold on most tasks, particularly struggling on tasks that require long-horizon planning and intricate whole-body coordination in a high-dimensional action space. Surprisingly, these state-of-the-art RL algorithms require a large number of steps to learn even simple locomotion tasks, such as `walk`, which has been extensively studied with a simplified humanoid agent in the DeepMind Control Suite [59].

This poor performance is mainly attributed to *the high-dimensionality of the state and action spaces* of our humanoid robot agent with dexterous hands. Although the hands of the humanoid robot are barely used for most locomotion tasks, the RL algorithms fail to ignore this information, which makes policy learning challenging. In addition, these high-dimensional state and action spaces result in a much larger exploration space, which makes exploration slow or infeasible with simple maximum entropy approaches. This implies the need for incorporating behavioral priors or common sense knowledge about the world that can ease the exploration problem, when it comes to learning on more complex agents, like humanoid robots. We investigate this problem further in Section V-C.

This problem becomes even more severe in manipulation tasks, resulting in particularly low rewards in all such tasks. Before learning any manipulation skills, an agent must learn locomotion skills to balance and move towards an object or the world to interact. All the policies barely learn to stabilize using the dense reward, but struggle to learn any complex manipulation skills.

C. With Hands vs. Alternative Configurations

With Hands vs. Without Hands. As discussed in Section V-B, controlling humanoid robots with dexterous hands is challenging due to their high degrees of freedom and complex dynamics. Thus, we investigate the difficulty of RL



(a) `walk` (b) `push`

Fig. 7: **Performance with and without dexterous hands.** The curves are averaged over three random seeds and the shaded regions represent the standard deviation.

training with a large action space (i.e., additional 42 dimensions with two dexterous Shadow Hands) on `walk` that does not necessarily require to control dexterous hands. The results in Figure 7 show that the presence of hands, with their additional joints and actuators, leads to a large decrease in performance compared to training the same task without the dexterous hands (see differences in observation and action space in Table II).

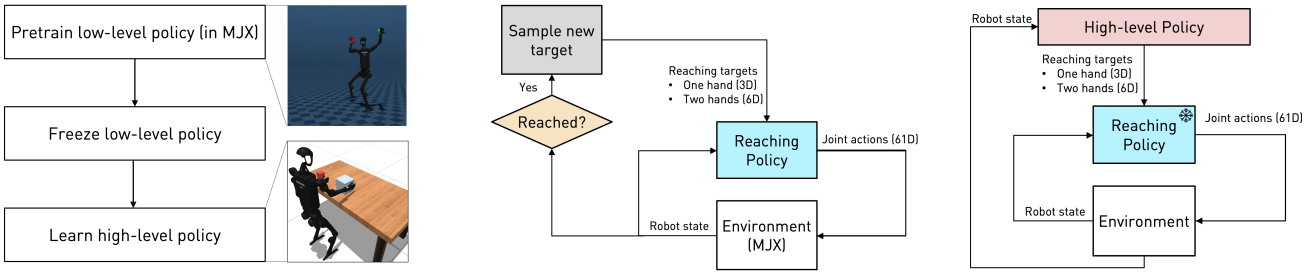
Reduced Action Space. To verify whether such difficulties stem from the dimensionality of the action space, we benchmark our full robot model, but fix the actuation of the hands (42D), which we set to zero. In this way, the action dimensionality is reduced from 61D in the original model to 19D. Note that the observations and masses induced by the presence of the hands are retained (i.e., observation space remains 151D). Figure 7 shows that the RL algorithms learn significantly faster in the reduced action space setup than the ones trained with the full action space. This confirms that most of the performance drop is indeed due to the increased action dimensionality.

We observe similar trends in the more complex manipulation task, `push`, wihch presents substantially different dynamics in the task approach (e.g., pushing with and without hands).

D. Flat vs. Hierarchical Reinforcement Learning

As shown in the previous subsection, flat, end-to-end RL approaches fail to learn most of the tasks in HumanoidBench. Many of such tasks require long-horizon planning and necessitate acquiring a diverse set of skills (e.g., balancing, walking, reaching). These issues can be mitigated by introducing additional structure into the learning problem. In particular, we explore a hierarchical learning paradigm, where one or multiple low-level skill policies are provided to a high-level planning policy that sends setpoints to lower-level policies. In practice, such setpoints comprise the action space of the high-level policy. This framework is very general, and there are no constraints on how to obtain both low-level and high-level policies. However, here we focus on training both of these through reinforcement learning [56].

Hierarchical RL Implementation. We implement a hierarchical RL approach on two manipulation tasks, namely, the `push` and `package` tasks. As a low-level skill, `push` uses a *one-hand reaching policy*, which allows the robot to reach a 3D point in space with its left hand, while `package` uses a



(a) Hierarchical learning pipeline (b) Low-level policy pretraining (c) High-level policy training

Fig. 8: **Our hierarchical RL pipeline (a).** (b) A robust low-level reaching policy is pretrained using PPO in a MuJoCo MJX-based reaching environment, as shown in the top snapshot in (a). (c) The high-level policy then leverages the pretrained reaching policy to move to a desired position and learns to solve a downstream task, shown in the bottom snapshot in (a). Note that the reaching policy weights are frozen during the high-level policy training.

two-hand reaching policy, where both hands are commanded to reach different 3D targets. Figure 8 illustrates the overview of our hierarchical RL implementation.

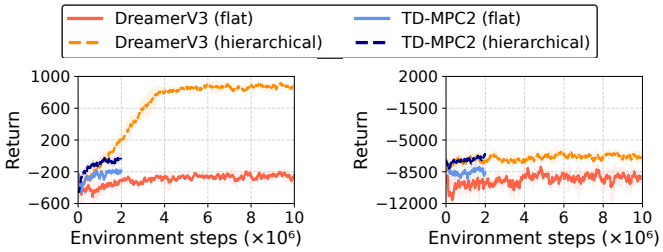
Low-level Reaching Policy Pretraining. We treat the low-level reaching policy as a pretrained frozen block that can be reused across tasks. Since this policy does not improve during training of the high-level policy, it needs to be very robust to cope with the continually shifting reaching targets that the high-level policy sets during exploration. However, the results in the previous section show that even a one-hand reaching task is hard to learn.

On a separate note, while our experiments above confirm that the on-policy PPO exhibits poor sample efficiency compared to the other off-policy algorithms, it is worth noting that PPO has achieved significant success in robotic locomotion by exploiting large-scale parallelization of environments on GPUs [36]. We exploit hardware acceleration by pretraining the low-level reaching policies in the recently released MuJoCo MJX[8], which enables training PPO on thousands of parallel environments.

For low-level reaching policy training, we employ a simplified H1 model that only considers collisions between feet and ground in the MuJoCo MJX environments, as in our experience the advantages stemming from parallelization are largely reduced when considering all numerous humanoid geometries (hindering training of the more complex benchmark tasks via MJX). We also remove the hands from the model to further increase training efficiency. The simplified reaching task environments for pretraining reset the target once reached. To achieve robust low-level reaching policies, we apply force perturbations at each of the links during training. We train the one-hand reaching policy for 2 billion steps (36 hours) and the two-hand reaching policy for 4 billion steps (60 hours) on $32,768$ parallel environments. The pretrained reaching policies successfully transfer to the original (non-simplified, simulated in classical MuJoCo) humanoid environments.

High-level Policy Training. Then, we use the pretrained reaching policies (frozen) as low-level policies and only train a high-level policy using either DreamerV3 and TD-MPC2 on the `push` and `package` tasks. To facilitate exploration, we restrict the range of reaching targets to the robot workspace.

[8]https://mujoco.readthedocs.io/en/stable/mjx.html



(a) `push` (b) `package`

Fig. 9: **Comparison between flat policies and hierarchical policies.** The curves are averaged over three random seeds and the shaded regions represent the standard deviation.

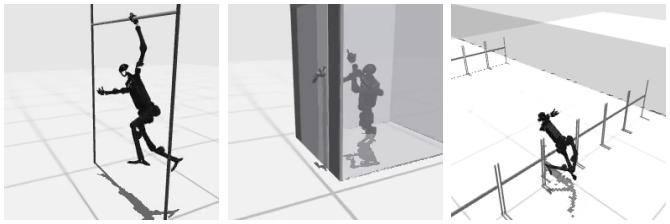
Hierarchical RL Results. In Figure 9, our hierarchical architecture significantly outperforms the flat, end-to-end baselines on the `push` task, achieving very high success rates with DreamerV3. While the low-level policy has undergone additional pretraining, this can be in principle reused across tasks. On the other hand, we note a less pronounced performance improvement in the more challenging `package` task. While getting closer to picking up the package with our hierarchical approach, the policy struggles in lifting it (having never experienced it during training).

These results confirm that the tasks in our benchmark present challenges that can be addressed with a more structured approach to the learning problem, and we hope this stimulates further directions for future research.

E. Common Failures

In this subsection, we remark on notable challenges and common failures for some representative tasks in our benchmark, which denote the challenge in learning with high-dimensional action spaces and limited planning horizon of the state-of-the-art RL algorithms.

Common Failure on `highbar`. In the `highbar` task, the Unitree H1 robot conservatively learns to maintain contact



(a) `highbar` (b) `door` (c) `hurdle`

Fig. 10: **Failure Scenarios.** This figure presents a selection of common failures that occur while training our benchmark tasks.

with the bar to avoid episode termination, but experiences difficulties in performing the whole-body rotation trajectory. This is indicative of short horizon planning and, despite the availability of dense rewards, is a recurrent challenge in many of the long-horizon benchmark tasks.

Common Failure on `door`. In the `door` task, the robot is well-guided to turn the door hatch to unlock the door, but it finds it challenging to learn the precise motion required to pull the door towards its opening position. This is mainly because pulling the door requires not only pulling its arm but also moving the whole body backwards. The coordination between multiple body parts and seamless interaction between manipulation and locomotion skills are common challenges in training humanoid robots.

Common Failure on `hurdle`. In the `hurdle` environment, the robot learns to run forward with the expected velocity but does not recognize the need to surpass the hurdle by jumping, which is a hard exploration problem. Previous work has shown that in OpenAI gym Walker2d, the forward-moving reward is sufficient to learn this behavior [29]. On the other hand, the humanoid robot finds conservative poses to collide with the hurdle such that it can stabilize without terminating the episode after hitting the obstacle, without further exploring high-reward jumping behaviors.

VI. Conclusion

We presented HumanoidBench, a high-dimensional humanoid robot control benchmark. Ours is the first example of a comprehensive humanoid environment with a diversity of locomotion and manipulation tasks, ranging from toy examples to practical humanoid applications. We set a high bar with our complex tasks, in the hope to stimulate the community to accelerate the development of whole-body algorithms for such robotic platforms.

Future work. HumanoidBench already includes multi-modal high-dimensional observations in the form of egocentric vision and whole-body tactile sensing. While our experiments only benchmarked the performance of state-based environments, studying the interplay between different modalities is a compelling direction for future work.

Extensions of the humanoid environment will also eventually include more realistic objects and environments with real-world diversity and higher-quality rendering. As for dexterous manipulation tasks, we envision screwing and furniture assembly tasks being part of our framework, given that they are particularly tailored for bimanual manipulation.

Here we have focused on reinforcement learning algorithms because humanoids are a setting where collecting physical demonstrations may be challenging. However, we believe that other means could be employed to bootstrap learning (e.g., learning from human videos).

Finally, while this was not the focus of our work, the impressive results obtained via domain randomization in the newly developed MuJoCo MJX show promise to study sim-to-real transfer in more depth, following the large success of the field in quadrupedal locomotion [23].

Acknowledgments

This work was supported in part by the SNSF Postdoc Mobility Fellowship 211086, ONR MURI N00014-22-1-2773, BAIR Industrial Consortium, Komatsu, InnoHK Centre for Logistics Robotics, an ONR DURIP grant, the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University) and RS-2024-00333634). We also thank Google TPU Research Cloud (TRC) for granting us access to TPUs for research.

References

[1] Alphonsus Adu-Bredu, Grant Gibson, and Jessy Grizzle. Exploring kinodynamic fabrics for reactive whole-body control of underactuated humanoid robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10397–10404. IEEE, 2023.

[2] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[3] Firas Al-Hafez, Guoping Zhao, Jan Peters, and Davide Tateo. Locomujoco: A comprehensive imitation learning benchmark for locomotion. *6th Robot Learning Workshop at NeurIPS*, 2023.

[4] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Association for the Advancement of Artificial Intelligence*, pages 1726–1734, 2017.

[5] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, jun 2013.

[6] Cameron H Berg, Vittorio Caggiano, and Vikash Kumar. SAR: Generalization of Physiological Dexterity via Synergistic Action Representation. In *Robotics: Science and Systems*, 2023.

[7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech

Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[8] Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. Myosuite: A contact-rich simulation suite for musculoskeletal motor control. In *Learning for Dynamics and Control*, pages 492–507. PMLR, 2022.

[9] Vittorio Caggiano, Sudeep Dasari, and Vikash Kumar. Myodex: a generalizable prior for dexterous manipulation. In *International Conference on Machine Learning*, pages 3327–3346. PMLR, 2023.

[10] Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuang Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[11] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

[12] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

[13] Yan Duan, Marcin Andrychowicz, Bradly Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in Neural Information Processing Systems*, pages 1087–1098, 2017.

[14] Siyuan Feng, Eric Whitman, X Xinjilefu, and Christopher G Atkeson. Optimization based full body control for the atlas robot. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 120–127. IEEE, 2014.

[15] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

[16] Dibya Ghosh. dibyaghosh/jaxrl_m, 2023. URL https://github.com/dibyaghosh/jaxrl_m.

[17] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *Conference on Robot Learning*, 2019.

[18] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1856–1865, 2018.

[19] Tuomas Haarnoja, Ben Moran, Guy Lever, Sandy H Huang, Dhruva Tirumala, Markus Wulfmeier, Jan Humplik, Saran Tunyasuvunakool, Noah Y Siegel, Roland Hafner, Michael Bloesch, Kristian Hartikainen, Arunkumar Byravan, Leonard Hasenclever, Yuval Tassa, Fereshteh Sadeghi, Nathan Batchelor, Federico Casarini, Stefano Saliceti, Charles Game, Neil Sreendra, Kushal Patel, Marlon Gwira, Andrea Huber, Nicole Hurley, Francesco Nori, Raia Hadsell, and Nicolas Heess. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *arXiv preprint arXiv:2304.13653*, 2023.

[20] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[21] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

[22] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.

[23] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[24] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020.

[25] Harini Kannan, Danijar Hafner, Chelsea Finn, and Dumitru Erhan. Robodesk: A multi-task reinforcement learning benchmark. https://github.com/google-research/robodesk, 2021.

[26] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 40:429–455, 2016.

[27] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021.

[28] Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. Scalable muscle-actuated human simulation and control. *ACM Transactions on Graphics*, 38(4):1–13, 2019.

[29] Youngwoon Lee, Shao-Hua Sun, Sriram Somasundaram, Edward S. Hu, and Joseph J. Lim. Composing complex skills by learning transition policies. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rygrBhC5tQ.

[30] Youngwoon Lee, Jingyun Yang, and Joseph J. Lim. Learning to coordinate manipulation skills via skill behavior diversification. In *International Conference on Learning Representations*, 2020.

[31] Youngwoon Lee, Edward S Hu, and Joseph J Lim. IKEA furniture assembly environment for long-horizon complex manipulation tasks. In *IEEE International Conference on Robotics and Automation*, 2021. URL https://clvrai.com/furniture.

[32] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gok-

men, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, 2022.

[33] L-J Lin. Hierarchical learning of robot skills by reinforcement. In *IEEE International Conference on Neural Networks*, pages 181–186. IEEE, 1993.

[34] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, 2020.

[35] Chris Lu, Jakub Kuba, Alistair Letcher, Luke Metz, Christian Schroeder de Witt, and Jakob Foerster. Discovered policy optimisation. *Advances in Neural Information Processing Systems*, 35:16455–16468, 2022.

[36] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu based physics simulation for robot learning. In *Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[37] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, 2021.

[38] Dominik Mattern, Pierre Schumacher, Francisco M López, Marcel C Raabe, Markus R Ernst, Arthur Aubret, and Jochen Triesch. Mimo: A multi-modal infant model for studying cognitive development. *IEEE Transactions on Cognitive and Developmental Systems*, 2024.

[39] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 2022.

[40] Josh Merel, Saran Tunyasuvunakool, Arun Ahuja, Yuval Tassa, Leonard Hasenclever, Vu Pham, Tom Erez, Greg Wayne, and Nicolas Heess. Catch & carry: reusable neural controllers for vision-guided whole-body tasks. *ACM Transactions on Graphics*, 39(4):39–1, 2020.

[41] Philipp Mittendorfer and Gordon Cheng. Humanoid multimodal tactile-sensing modules. *IEEE Transactions on robotics*, 27(3):401–410, 2011.

[42] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 3303–3313, 2018.

[43] Yashraj Narang, Kier Storey, Iretiayo Akinola, Miles Macklin, Philipp Reist, Lukasz Wawrzyniak, Yunrong Guo, Adam Moravanszky, Gavriel State, Michelle Lu, Ankur Handa, and Dieter Fox. Factory: Fast contact for robotic assembly. In *Robotics: Science and Systems*, 2022.

[44] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[45] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 37(4):1–14, 2018.

[46] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions on Graphics*, 37(6):1–14, 2018.

[47] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40(4):1–20, 2021.

[48] Karl Pertsch, Youngwoon Lee, and Joseph J. Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on Robot Learning*, 2020.

[49] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, Vikash Kumar, and Wojciech Zaremba. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

[50] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Learning humanoid locomotion with transformers. *arXiv preprint arXiv:2303.03381*, 2023.

[51] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3, 2019.

[52] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[53] Carmelo Sferrazza and Raffaello D'Andrea. Sim-to-real for high-resolution optical tactile sensing: From images to three-dimensional contact force distributions. *Soft Robotics*, 9(5):926–937, 2022.

[54] Carmelo Sferrazza, Younggyo Seo, Hao Liu, Youngwoon Lee, and Pieter Abbeel. The power of the senses: Generalizable manipulation from vision and touch through masked multimodal learning. *arXiv preprint arXiv:2311.00924*, 2023.

[55] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, Silvio Savarese, Hyowon Gweon, Jiajun Wu, and Li Fei-Fei.

Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on Robot Learning*, 2021.

[56] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[57] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Neural Information Processing Systems*, 2021.

[58] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[59] Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, and Nicolas Heess. dm_control: Software and tasks for continuous control. *arXiv preprint arXiv:2006.12983*, 2020.

[60] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[61] Yinhuai Wang, Jing Lin, Ailing Zeng, Zhengyi Luo, Jian Zhang, and Lei Zhang. Physhoi: Physics-based imitation of dynamic human-object interaction. *arXiv preprint arXiv:2312.04393*, 2023.

[62] Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022.

[63] Zhaoming Xie, Jonathan Tseng, Sebastian Starke, Michiel van de Panne, and C Karen Liu. Hierarchical planning and control for box loco-manipulation. *Symposium on Computer Animation*, 2023.

[64] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, 2019.

[65] Ying Yuan, Haichuan Che, Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Kang-Won Lee, Yi Wu, Soo-Chul Lim, and Xiaolong Wang. Robot synesthesia: In-hand manipulation with visuotactile sensing. *arXiv preprint arXiv:2312.01853*, 2023.

[66] Kevin Zakka, Yuval Tassa, and MuJoCo Menagerie Contributors. MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo, 2022. URL http://github.com/google-deepmind/mujoco_menagerie.

[67] Kevin Zakka, Philipp Wu, Laura Smith, Nimrod Gileadi, Taylor Howell, Xue Bin Peng, Sumeet Singh, Yuval Tassa, Pete Florence, Andy Zeng, and Pieter Abbeel. Robopianist: Dexterous piano playing with deep reinforcement learning. In *Conference on Robot Learning*, pages 2975–2994. PMLR, 2023.

[68] Haotian Zhang, Ye Yuan, Viktor Makoviychuk, Yunrong Guo, Sanja Fidler, Xue Bin Peng, and Kayvon Fatahalian. Learning physically simulated tennis skills from broadcast videos. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[69] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023.

[70] Yuke Zhu, Josiah Wong, Ajay Mandlekar, and Roberto Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

[71] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher G Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. In *Conference on Robot Learning*, 2023.