Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

RoboVerse: Towards a Unified Platform, Dataset and Benchmark for Scalable and Generalizable Robot Learning

Haoran Geng[1*], Feishi Wang[1,2,3*], Songlin Wei[2*], Yuyang Li[2,9*], Bangjun Wang[3*], Boshi An[2*], Charlie Tianyue Cheng[1*], Haozhe Lou[3], Peihao Li[1,4], Yen-Jen Wang[1], Yutong Liang[2], Dylan Goetting[1], Chaoyi Xu[2], Haozhe Chen[5], Yuxi Qian[6], Yiran Geng[2], Jiageng Mao[3], Weikang Wan[2], Mingtong Zhang[3], Jiangran Lyu[2], Siheng Zhao[3], Jiazhao Zhang[2], Jialiang Zhang[1,2], Chengyang Zhao[7], Haoran Lu[2], Yufei Ding[1,2], Ran Gong[8], Yuran Wang[2], Yuxuan Kuang[2,3], Ruihai Wu[2], Baoxiong Jia[9], Carlo Sferrazza[1], Hao Dong[2], Siyuan Huang[9†], Yue Wang[3†], Jitendra Malik[1†], Pieter Abbeel[1†]

[1]UC Berkeley [2]PKU [3]USC [4]UMich [5]UIUC [6]Stanford [7]CMU [8]UCLA [9]BIGAI
* equal contribution † equal advising Correspondence to: Haoran Geng <ghr@berkeley.edu>

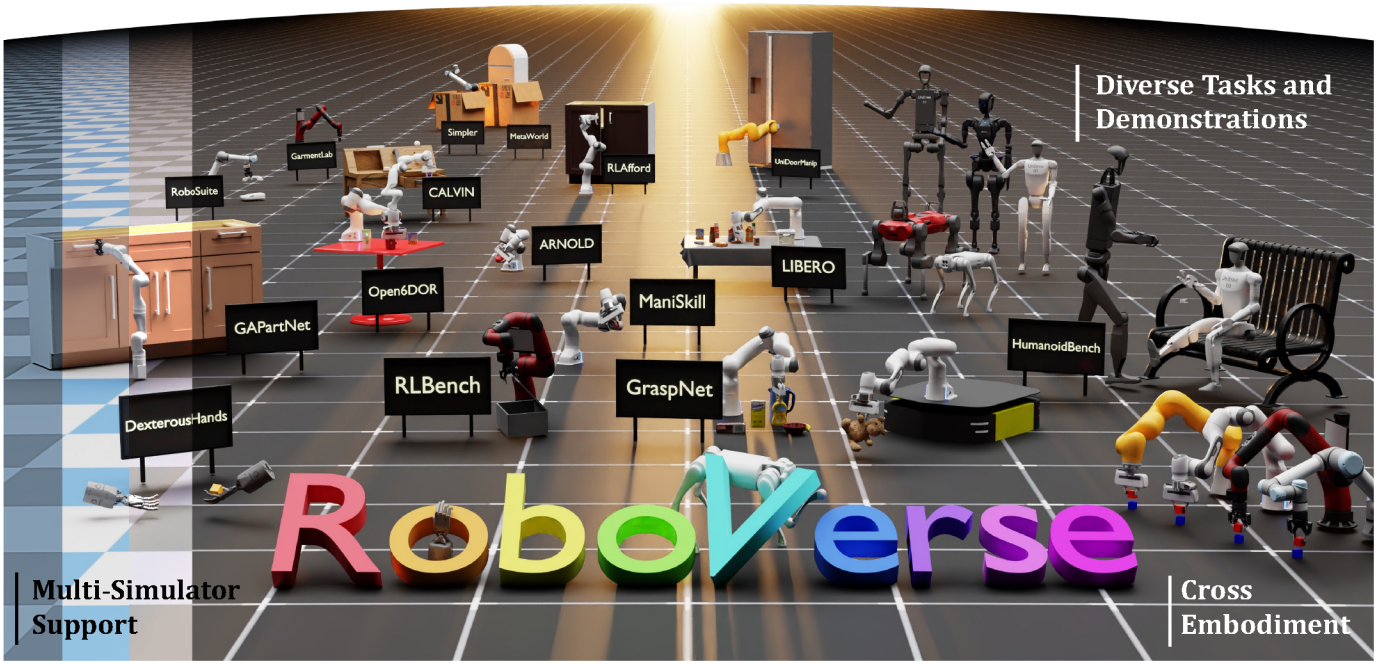


Fig. 1: RoboVerse comprises a scalable simulation platform, a large-scale synthetic dataset, and unified benchmarks. The simulation platform supports seamless integration of new tasks and demonstrations through unified protocols, ensuring flexibility and extensibility. The dataset includes over 1,000 diverse tasks and more than 10 million transitions, constructed through large-scale data migration, cross-embodiment transfer, and robust augmentation and randomization.

***Abstract*—Data scaling and standardized evaluation benchmarks have driven significant advances in natural language processing and computer vision. However, robotics faces unique challenges in scaling data and establishing reliable evaluation protocols. Collecting real-world robotic data is resource-intensive and inefficient, while benchmarking in real-world scenarios remains highly complex. Synthetic data and simulation offer promising alternatives, yet existing efforts often fall short in data quality, diversity, and benchmark standardization. To address these challenges, we introduce RoboVerse, a comprehensive framework comprising a *simulation platform*, a *synthetic dataset*, and *unified benchmarks*. Our simulation platform supports multiple simulators and robotic embodiments, enabling seamless transitions between different environments. The synthetic dataset, featuring high-fidelity physics and photorealistic rendering, is constructed through multiple approaches including migration from public datasets, policy rollout, and motion planning, *etc*. enhanced by data augmentation. Additionally, we propose unified benchmarks for imitation learning and reinforcement learning, enabling consistent evaluation across different levels of generalization. At the core of the *simulation platform* is MetaSim, an infrastructure that abstracts diverse simulation environments into a universal interface. It restructures existing simulation environments into a simulator-agnostic configuration system, as well as an API aligning different simulator functionalities, such as launching simulation environments, loading assets with initial states, stepping the physics engine, *etc*. This abstraction ensures interoperability and extensibility. Comprehensive experiments demonstrate that RoboVerse enhances the performance of imitation learning, reinforcement learning, and world model learning,**

improving sim-to-real transfer. These results validate the reliability of our dataset and benchmarks, establishing RoboVerse as a robust solution for advancing simulation-assisted robot learning. Code and dataset can be found at: https://roboverseorg.github.io/.

I. Introduction

Large-scale datasets, combined with well-established benchmarks, have fueled rapid advancements in natural language processing (NLP) [75, 5] and computer vision (CV) [19, 45, 43, 77, 56, 33]. Specifically, large-scale data provides ample training examples that bolster learning, while uniform benchmarks enable standardized evaluation and fair comparison across different methods. However, replicating these successes in robotics remains challenging due to the difficulty of collecting high-quality, diverse data and the lack of widely recognized evaluation protocols.

Real-world approaches [14, 41] to constructing datasets and benchmarks, though authentically reflecting the complexities of operational environments, face significant practical constraints. First, collecting demonstrations is time-consuming and resource-intensive, and the resulting data is often hardware-dependent or modality-specific, limiting its adaptability to new scenarios. Additionally, establishing standardized and widely applicable benchmarks is inherently challenging since reproducing identical conditions for fair comparisons is nearly impossible. For instance, object placements can vary across rollouts, ambient lighting fluctuates under natural sunlight, and background environments may change. Consequently, scaling real-world datasets, evaluating policies, and iterating development in real-world scenarios remain cost-prohibitive and difficult to standardize.

Simulators, on the other hand, present a promising alternative for large-scale dataset and benchmark construction. By providing efficient computation, synthetic assets, and omniscient information in reproducible settings, they enable cost-effective dataset construction and consistent performance evaluation. Recent works, exemplified by [104, 37, 9, 27, 80, 98, 59, 48, 97, 94, 50, 51, 73], have demonstrated the potential of simulation-based methods in various robotic tasks. Despite these advantages, several challenges impede the broader adoption of synthetic datasets and benchmarks. First, utilizing simulators often demands considerable expertise due to both the complexity of simulator design and the relative immaturity of many platforms, which complicates the data construction process. Second, simulators vary widely in their internal architectures and external interfaces, making it laborious to transfer data and models or adapt workflows from one to another. Consequently, reusing existing synthetic datasets and benchmarks is difficult, resulting in a fragmented ecosystem that further hinders convenient construction and effective use of large-scale data in simulation environments.

To fully harness the potential of simulation in robotics, we introduce RoboVerse, a scalable simulation platform that unifies existing simulators under a standardized format and a single infrastructure, a large-scale synthetic dataset, and unified benchmarks. To achieve this, we first propose MetaSim, the core infrastructure of the RoboVerse. Through careful design, MetaSim establishes a universal configuration system for agents, objects, sensors, tasks, and physics parameters while exposing a simulator-agnostic interface for simulation setup and control. This architecture enables seamless integration of tasks, assets and robot trajectories from diverse simulation environments with minimal adaptation effort. MetaSim provides three key capabilities: (1) *Cross-Simulator Integration*: Enables seamless switching between different simulators, fostering unified benchmarking and facilitating the transfer of environments and demonstrations across platforms. (2) *Hybrid Simulation*: Combines the strengths of multiple simulators—such as pairing advanced physics engines with superior renderers—to generate scalable and high-quality synthetic data. (3) *Cross-Embodiment Transfer*: Allows the retargeting of trajectories across various robot arms with parallel grippers, maximizing dataset reuse from heterogeneous sources.

MetaSim enables RoboVerse to systematically enhance the workflow for building and scaling simulation environments and datasets. Our method features:

- *Scalable and Diverse Data Generation*: By aligning multiple benchmarks and task trajectories and leveraging a robust multi-source integration and data filtering pipeline, we generate large-scale, high-quality datasets. Additionally, our data randomization and augmentation pipeline enhances data diversity and volume, further enriching the dataset for comprehensive model training;
- *Realistic Simulation and Rendering*: With MetaSim's hybrid simulation capability, we enable the fusion of advanced physics engines and rendering systems across multiple simulators and renderers. Combined with carefully curated scenes, materials, and lighting assets, RoboVerse enhances realism in physical interactions and sensory observations;
- *Unified Benchmarking and Evaluation*: We unify widely used benchmarks into a cohesive system, streamlining algorithm development and performance comparison within a structured evaluation framework. Additionally, we introduce a standardized benchmarking protocol to assess varying levels of generalization and sim-to-real transferability.
- *Highly Extensibility and Scalability:* The aligned APIs and infrastructure streamline development and enable efficient algorithm integration, testing, and deployment across diverse simulation environments. Additionally, we develop real-to-sim frameworks, multiple teleoperation methods, and AI-generative systems for scalable task and data creation.

Leveraging these workflows in RoboVerse, we construct the largest and most diverse high-quality synthetic dataset and benchmark to date, all in a unified format. This dataset includes ~500k unique, high-fidelity trajectories covering 276 task categories and ~5.5k assets. Additionally, we generate over 50 million high-quality state transitions to support policy learning.

Beyond dataset and benchmark construction, we explore the potential of ROBOVERSE through extensive experiments on imitation learning (Sec. VI-B), reinforcement learning (Sec. VI-C), and world model learning (Sec. VI-E). Our results demonstrate that ROBOVERSE enables reliable policy learning and evaluation, supports strong sim-to-sim and (Sec. VI-G) sim-to-real transfer (Sec. VI-F) via high-fidelity physics and rendering, and facilitates efficient data expansion through teleoperation (Sec. IV-C), trajectory augmentation (Sec. IV-D1), domain randomization (Sec. IV-D2) and generative models (Sec. IV-C). These findings highlight the framework's robustness, scalability, and real-world applicability.

II. RELATED WORK

A. Robotics Simulators

Advancements in computer graphics have contributed to the development of high-fidelity simulators, which are widely used in robotics research and development. CoppeliaSim [79], Bullet [15], and MuJoCo [90] provide accurate physics simulations and are extensively utilized in applications such as reinforcement learning and robotic benchmarking [3, 99, 71, 13]. More simulators have been developed to fully exploit parallelism for better efficiency. Isaac Gym [60], Isaac Sim [69], SAPIEN [30, 88], MuJoCo MJX [90, 103], and Genesis [2] utilize GPU power for enhanced performance, enabling large-scale reinforcement learning and efficient data collection, significantly improving training speed and scalability. Some simulators focus on bridging the simulation-reality gap (Sim-to-Real Gap), incorporating technologies including ray-tracing and customized renderers for photo-realistic rendering [69, 88]. Furthermore, Isaac Sim [69] and Genesis [2] offer high-fidelity soft-body and liquid simulation, expanding the scope of realistic robotic interactions. ROBOVERSE proposes a unified platform that supports multiple simulators, facilitating seamless transitions between them and enabling hybrid integration to utilize the strengths of each simulator.

B. Large-Scale Robotics Dataset

The scarcity of large-scale, high-quality, and diverse datasets in the robotics community has long been recognized. Several works have shown the possibility of collecting demonstration data directly on real robots. RoboNet [18] is a large-scale manipulation dataset containing roughly 162k trajectories from multiple robot platforms. DROID [41] has collected over 76k contact-rich robotic manipulation demonstrations across 86 tasks. RH20T [24] proposed a dataset with over 100k demonstrations and 147 tasks. At the same time, RT-1 [4] set the record further to 130k demonstrations on over 700 tasks. Recently, Open X-Embodiment [14] has demonstrated a promising approach to unite the community's efforts, collecting over 1M trajectories on 160,266 tasks with 22 different embodiments. At this stage, real-world datasets became difficult to scale up due to the proportional effort and cost required to collect more demonstrative trajectories.

Simulation-based data collection provides a promising solution to the high cost and inefficiencies of real-world datasets. Hussing *et al.* [35] proposed a dataset containing 256M transitions on 256 tasks for offline compositional reinforcement learning. RoboCasa [67] introduced a dataset of 100 tasks and over 100k trajectories for generalist robots. DexGraspNet-2.0 [104] has collected over 400M demonstrations for dexterous grasping. Despite these efforts, synthetic datasets often exist in disparate simulators, leading to a fragmented ecosystem with limited diversity and quality. Moreover, simulation-based data often fails to capture complex physics and diverse task variations found in the real world [52, 22], potentially causing overfitting to specific simulators and hampering generalization to real-world scenarios.

ROBOVERSE provides a unified solution for large-scale, high-quality, and diverse synthetic data. It enables agents to train on a large set of environments and simulators to reduce overfitting, thereby improving the robustness of the learned policies.

C. Benchmarking in Robotics

Benchmarking remains a critical yet highly challenging problem in the robotics community. Compared to supervised learning tasks, it is relatively difficult to evaluate the performance of a robotics model. Meta-World [102] is an early attempt in multi-task benchmarking. This is followed by RLBench [36], BEHAVIOR-1K [49], Habitat [87], and ManiSkill [66, 30, 88, 85], covering a large variety of robotic tasks. Grutopia [95] and InfiniteWorld [78] make a leap toward general-purpose robot benchmarking.

Despite significant efforts dedicated to these benchmarks, it is not guaranteed that the results are reproducible across different benchmarks. The uncertainty comes from multiple aspects including simulation accuracy, rendering style and asset properties [52, 22]. To address these challenges, ROBOVERSE enables researchers to evaluate their policies across multiple benchmarks and simulators seamlessly, without familiarizing themselves with each one individually.

III. INFRASTRUCTURE: METASIM

A. METASIM *Overview*

We present METASIM, a high-level interface above specific simulation environment implementations. It is also the core infrastructure of ROBOVERSE. As illustrated in Fig. 2, METASIM empowers the ROBOVERSE simulation platform, allowing for the generation of a large-scale high-quality dataset, as well as the construction of a unified benchmark.

B. METASIM *Implementation*

As illustrated in Fig. 3, METASIM employs a three-layer architecture including a universal configuration system, a simulator-agnostic interface, and a user-friendly environment wrapper. The universal configuration system unifies specifications for a simulation scenario and ensures consistent format across simulators. The simulator-agnostic interface interprets these specifications, translates them into simulator-specific commands, and therefore aligns different simulator backends. In

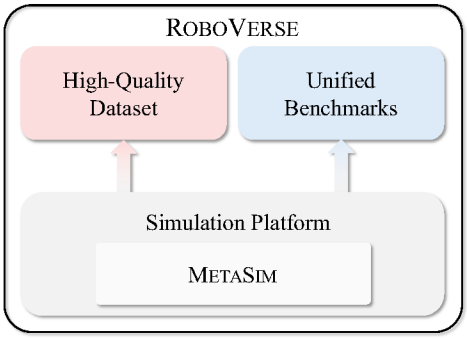


Fig. 2: ROBOVERSE consists of a simulation platform, a large-scale, high-quality dataset, and unified benchmarks. At the core of the simulation platform is METASIM, the infrastructure of ROBOVERSE. Powered by METASIM, the simulation platform facilitates dataset creation and benchmark construction.

addition, the environment wrappers encapsulate the simulator-agnostic interface into a standarized learning environment, such as a Gym [91] environment. We describe each layer with more details in the following sections.

1) Universal Configuration System: A typical simulation environment comprises agents, objects, tasks, sensors, and physics parameters. They collectively define who performs the actions (agents), what the environment looks like (objects), what the agents should do (tasks, including instructions, success metrics, and rewards), how the environment is perceived and measured (sensors), and the governing physical laws (physics parameters). Ideally, these components should be simulator-agnostic, requiring a unified standard of simulation scenarios. Such a standard would enable researchers to work across different simulators seamlessly and integrate existing efforts from the community through cross-simulation.

Based on such a principle, we design a configuration system, `MetaConfig`, to abstract simulation scenarios in a simulator-agnostic way. As illustrated in Fig. 4, `MetaConfig` is a nested class that contains the above-mentioned core components. It can be interpreted by different simulator backends to build the corresponding simulation. Additionally, `MetaConfig` supports optional simulator-specific hyperparameters (*e.g.*, solver type), allowing fully leveraging the unique features of different simulators through customization.

2) Aligned Simulator Backends: Different simulators have their own implementations and specializations. However, routine operations – such as initializing a scene, loading objects, stepping the physics engine, retrieving observations, time management, and determining success states – tend to follow similar patterns. To standardize these shared operations, we create a unified interface through a `Handler` class. Each simulator has its own handler instance implementing this interface. The handler class implements the common methods including `launch()`, `get_states()`, and `set_states()`, *etc.*, spanning the whole lifecycle of simulating a task. The usage of the APIs is illustrated in Code 1. More information is provided in the supplementary materials.

```
class Env:
    def __init__(self, handler):
        self.handler = handler
        handler.launch()

    def reset(self):
        handler.set_states()
        states = handler.get_states()
        return get_observation(states), \
               handler.get_extra()

    def step(self, action):
        handler.set_states(action=action)
        handler.step()
        states = handler.get_states()
        return get_observation(states), \
               get_reward(states), \
               get_success(states) \
               get_termination(states), \
               get_time_out(states), \
               handler.get_extra()

    def render(self):
        return handler.render()

    def close(self):
        handler.close()
```

Code 1: Pseudocode for `gym.Env` implementation. Each method of `gym.Env` is implemented by calling the corresponding methods of the `Handler` class.

3) User-Friendly Environment Wrapper: Gym [91] is a widely adopted paradigm in reinforcement learning and robotics, in which the `gym.Env` class is fundamental to building learning environments. We define a wrapper to easily transform a `Handler` into an environment equipped with Gym APIs (`step()`, `reset()`, `render()`, and `close()`). As shown in Code 1, these methods are implemented by leveraging the underlying `Handler` methods.

C. METASIM *Capabilities*

METASIM offers the following three key capabilities.

1) Cross-Simulator Integration: Seamlessly switching between different simulators, allowing tasks and trajectories from one simulator to be utilized in other simulators. This capability enables efficient task and trajectory integration, unified benchmark construction, and sim-to-sim transfer for reinforcement learning training. For example, tasks from Meta-World [102] can be used by Isaac Gym [60] for fast parallel training, after which the generated trajectories can be deployed in Isaac Sim [69] for rendering.

2) Hybrid Simulation: METASIM supports combining the physics engine of one simulator and the renderer of another simulator at the same time, allowing users to benefit from advantages owned by different simulators. Specifically, using a single command, one could launch a simulator with a powerful renderer (*e.g.*, Isaac Sim [69]) with a simulator that has an accurate physics engine (*e.g.*, MuJoCo [90]) to form an

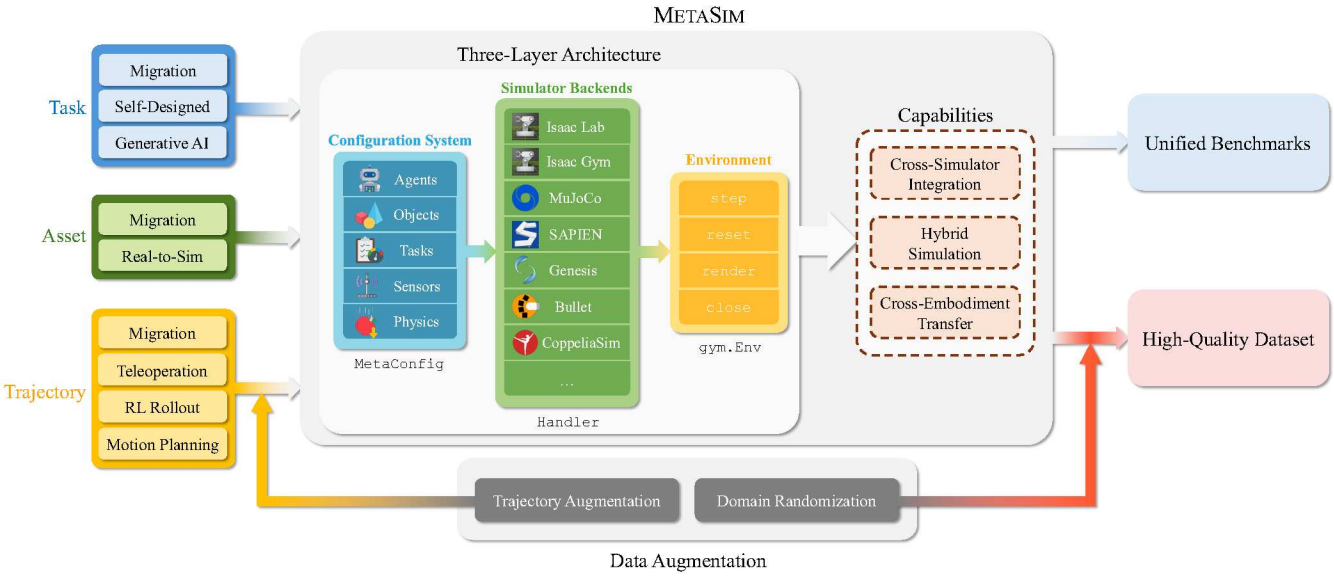


Fig. 3: METASIM provides a universal configuration system, aligned simulator backends, and a Gym [91] environment wrapper. This three-layer architecture abstracts simulation environments into simulator-agnostic specifications and aligns simulator backends, enabling three key capabilities: cross-simulator integration, hybrid simulation and cross-embodiment transfer. Based on METASIM, we build a pipeline to collect tasks, assets and trajectories from diverse public sources in a unified format, employ data augmentation methods, and ultimately generate a large-scale high-quality dataset along with unified benchmarks. This data pipeline forms the foundation of ROBOVERSE, facilitating the generation of large-scale datasets and construction of unified benchmarks.

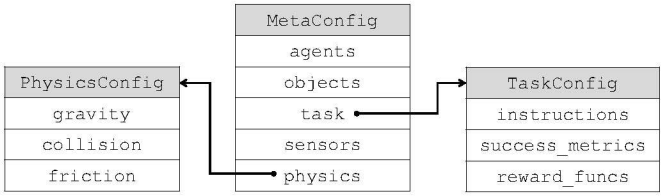


Fig. 4: The `MetaConfig` is a nested dataclass that abstracts the core components in any simulation environment in a simulator-agnostic way.

even more powerful simulation, enabling high-quality data generation.

3) Cross-Embodiment Transfer: Reusing the trajectories across different gripper-based robot morphologies by retargeting the end-effector pose, which allows the integration of data collected from diverse robots into a unified format.

IV. ROBOVERSE DATASET

A. Dataset Overview

On top of METASIM, we generate large-scale high quality dataset by incorporating multiple data collection methods. Overall, there are three key data types to collect: tasks, assets, and robot trajectories. The main source of these data is migration from existing simulation environments. Beyond migration, we explore various methods to collect these data, such as using large language models to generate new tasks, leveraging the real-to-sim toolset [57] to reconstruct assets from the real world, using teleoperation to collect new trajectories, *etc*. Additionally, we leverage data augmentation methods for both trajectories and visual observations. Finally, we report the statistics for current progress of data migration in ROBOVERSE.

B. Tasks, Assets and Trajectories Collection: Migration

Leveraging the ROBOVERSE format and infrastructure, we seamlessly integrate a wide range of benchmarks and datasets into our system with a unified format and clean codebase. We apply the following approaches to collect tasks and demonstrations.

- **Direct Migration from Other Simulation Environments** Some benchmarks provide essential components integration into ROBOVERSE. We define environment configurations for task initialization and evaluation, then convert trajectory data and asset formats for seamless compatibility. Notably, ROBOVERSE streamlines this migration process by first aligning formats in the original simulator and automatically ensuring compatibility across all simulators.
- **Motion Planning and RL Rollout** When benchmarks provide only partial manipulation data, such as keypoint trajectories or grasping poses, we use motion planning to generate complete trajectories. If no explicit manipulation data is available but pre-existing policies or reinforcement learning frameworks exist, we either utilize these policies or train new ones to collect demonstration data through rollouts. To ensure high data quality and consistency with

our system standards, we carefully adapt the success checker and rigorously filter both planned and collected trajectories.

With the techniques mentioned above, we migrated multiple existing manipulation datasets into RoboVerse. Currently, we support ManiSkill [66, 30, 88], RLBench [36], CALVIN [64], Meta-World [102], `robosuite` [109], MimicGen [61], GAPartNet [26], Open6DOR [20], ARNOLD [29], LIBERO [54], SIMPLER [52], GraspNet [23], GarmentLab [58], and UniDoorManip [53].

We also integrated datasets from a wider range of embodiments, including dexterous hands, quadrupeds, and humanoids, covering tasks such as dexterous manipulation, locomotion, navigation, and whole-body control. Currently, we have migrated VLN-CE R2R [44] and RxR [46] for navigation, as well as HumanoidBench [84] and Humanoid-X [62] for locomotion and whole-body control.

RoboVerse simplifies and standardizes the migration process, and we will continue to maintain and expand it.

C. Tasks, Assets and Trajectories Collection: Teleoperation and Generation

- **Teleoperation System for Trajectory Collection** . As shown in Fig. 5, RoboVerse integrates teleoperation systems within the MetaSim infrastructure, offering a flexible and efficient solution for high-quality data collection. It supports various robotic systems, including arms, dexterous hands [72], and bimanual setups, enabling seamless teleoperation across different simulators. To mitigate the high cost and complexity of professional equipment, we introduce an interactive motion control system utilizing accessible devices such as keyboards, joysticks, mobile apps (we developed a new app for Android and iOS to control robotic arms; see supplementary materials for more details.), motion capture (Mocap) [93], and VR systems [11, 74]. These devices' integrated sensors capture motion data, allowing natural, gesture-based control along with real-time, high-frequency communication for precise, low-cost remote operation. Further details are provided in the supplementary materials.
- **AI-Assisted Task Generation**. Leveraging the generalization capability of large generative models, AI-assisted task generation provides a mechanism to diversify task varieties and scenario distribution. By learning from example placements, it acquires a sense of spatial and semantic constraints [1] (*e.g.* by demonstrating specific constraints, it can learn to spread out objects to avoid potential overlap *etc.*). It can arrange objects originally from different benchmarks into a physically plausible scenes based on MetaSim, as shown in Fig. 6. Incorporating randomization in robot and object selection [39] with their initial poses, large generative models can generate various initial states. The system can automatically output all the required configuration files in unified format for instant visualization and user-friendly editing. After task generation, we will process a two-step filtering to avoid errors and hallucinations: (1) *Format Validation*: Tasks that fail to meet RoboVerse format standards are discarded. (2) *Feasibility Check*: Since trajectory data is collected via human teleoperation, tasks deemed unreasonable by the teleoperator are removed. By unleashing the extrapolative and few-shot learning abilities of large generative models, we integrate assets under a uniform schema automatically, driving task generation that spans multiple simulators and benchmarks.
- **Real-to-Sim for Asset Construction**. Video-based reconstruction proves to be a valuable source for data and asset creation by leveraging Real-to-Sim techniques. Our approach integrates multiple reconstruction pipelines to extract high-fidelity assets from video data. First, we initialize the structure using COLMAP [81, 82] and employ Gaussian Splatting [40] for high-quality rendering. Next, we infer physical properties by feeding both semantic and original images into a Vision-Language Model (VLM) [108]. For geometry reconstruction, we estimate surface normals from video [101], apply surfel splatting [34], and utilize TSDF-based methods with dynamic filtering to reconstruct detailed meshes [100]. By leveraging semantic masks [77], we selectively extract components from both Gaussian and mesh representations. To further enhance realism, we infer and learn object kinematics directly from video [55], ensuring accurate motion representations. Finally, we formulate URDF models by refining key attributes such as coordinate frames, orientation, axis alignment, scale, relative 6-DoF poses, and PD control parameters [57]. This pipeline effectively bridges the gap between real-world video data and simulation-ready assets, enhancing robotic learning and simulation fidelity. We also present comparative experiments in the supplementary materials, demonstrating that our methods significantly enhance real-world policy performance.

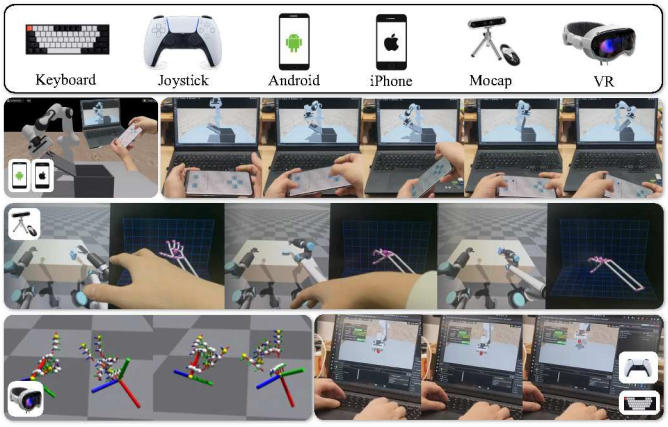


Fig. 5: **Teleoperation System.** RoboVerse supports various user-friendly teleoperation approaches. Currently, it enables teleoperation via a phone app (second row), motion capture (middle), VR devices (bottom left), as well as keyboard and joystick (bottom right). These methods allow control of robotic arms, dexterous hands, and bimanual systems across different simulators.

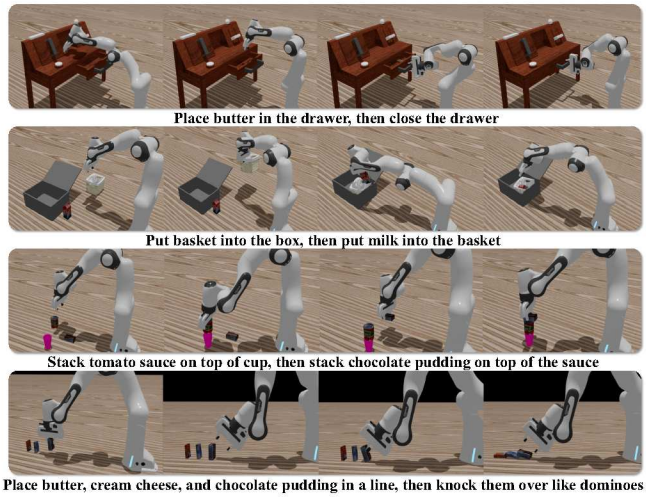


Fig. 6: **AI-Assisted Task Generation.** RoboVerse supports an AI-assisted task generation framework that leverages large generative models' extrapolation capabilities to generate non-trivial and semantically rich tasks. Combined with our teleoperation system, it enables the generation of diverse and high-quality data.

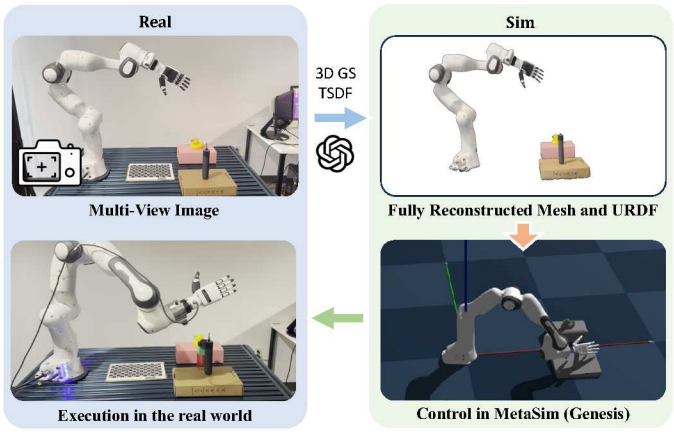


Fig. 7: **Real-to-Sim Tools.** We use a mobile device to capture multi-view images, reconstruct a high-quality mesh, build a URDF using VLM, and then perform actions in both RoboVerse and the real world.

D. Data Augmentation

1) Trajectory Augmentation: With the unified simulation interface and data format, RoboVerse enables significantly more efficient data augmentation and supports advanced augmentation techniques. Beyond the visual randomization detailed in Benchmark Protocol [7], we also provide robust trajectory space augmentation. We offer an API to generate large-scale robot trajectory datasets from a limited number of source demonstrations. Following the MimicGen [61] framework, for most tasks, we can decompose them into a sequence of object-centric subtasks $(S_1(o_{S_1}), S_2(o_{S_2}), \ldots, S_M(o_{S_M}))$, where the robot's trajectory within each subtask $S_i(o_{S_i})$ is relative to a single object's coordinate frame ($o_{S_i} \in \mathcal{O}$, $\mathcal{O}$ is the set of objects in the task $\mathcal{M}$). Additionally, we assume that the sequence of subtasks in each task is predefined. By leveraging this minimal human annotation regarding the order of subtasks, we can efficiently divide each source demo into contiguous object-centric manipulation segments $\{\tau_i\}_{i=1}^{M}$ (each of which corresponds to a subtask $S_i(o_i)$) using a simulator, and then generate extensive trajectory datasets for various task variants (in our case: variations in the initial and goal state distributions of objects (D) and robots (R)) using MimicGen [61]. This approach has been shown to significantly benefit generalization in imitation learning [61, 37, 96, 25, 67], particularly in scenarios where the number of source demonstrations is limited. For further details, please refer to the supplementary materials.

2) Domain Randomization: We implement domain randomization in the Isaac Sim [69] handler of MetaSim. This involves four types of randomization:

- **Table, Ground, and Wall.** Walls (and ceilings) can be added for tasks that lack a predefined scene. Customizable tables can also be included for tasks that are performed on tabletops. The visual materials for these elements are randomly selected from a curated subset of ARNOLD [29] and vMaterials [68]. The table has ∼300 material options, while the wall and ground each have around ∼150 material options.
- **Lighting Condition.** Two types of lighting scenarios can be specified: distant light and cylinder light arrays. For distant light, the light's polar angles are randomized. For cylinder light, a random $n \times m$ matrix of cylinder lights with random size is added at a fixed height above the agents. In both scenarios, the intensity and color temperature of the lights are randomized within a reasonable range.
- **Camera Poses.** We carefully select 59 candidate camera poses, with the majority positioned to face the robot directly and a smaller subset placed at side-facing angles.
- **Reflection Properties.** The roughness, specular, and metallic properties of each surface are randomized within reasonable ranges.

These randomization options can be freely combined. For example, a scene can include a customized table, walls with a ceiling, and a set of cylinder lights to simulate an indoor environment. For details, please refer to the supplementary materials.

E. RoboVerse Dataset

1) Dataset Statistics:

a) Manipulation Dataset: We migrate diverse manipulation datasets from existing source benchmarks [66, 30, 88, 36, 64, 102, 109, 61, 26, 20, 29, 54, 52, 28, 23, 58, 53, 16] into RoboVerse. The number of task categories, trajectories and assets contributed by each source benchmarks is summarized in Tab. I. In total, this migration results in 276 task categories, 510.5k trajectories, and 5.5k assets. Representitive tasks with rich domain randomization are shown in Fig. 8.

b) Navigation Dataset: We migrate vision-and-language navigation (VLN) tasks into RoboVerse. Note that there exists various VLN tasks with different settings; here, we particularly

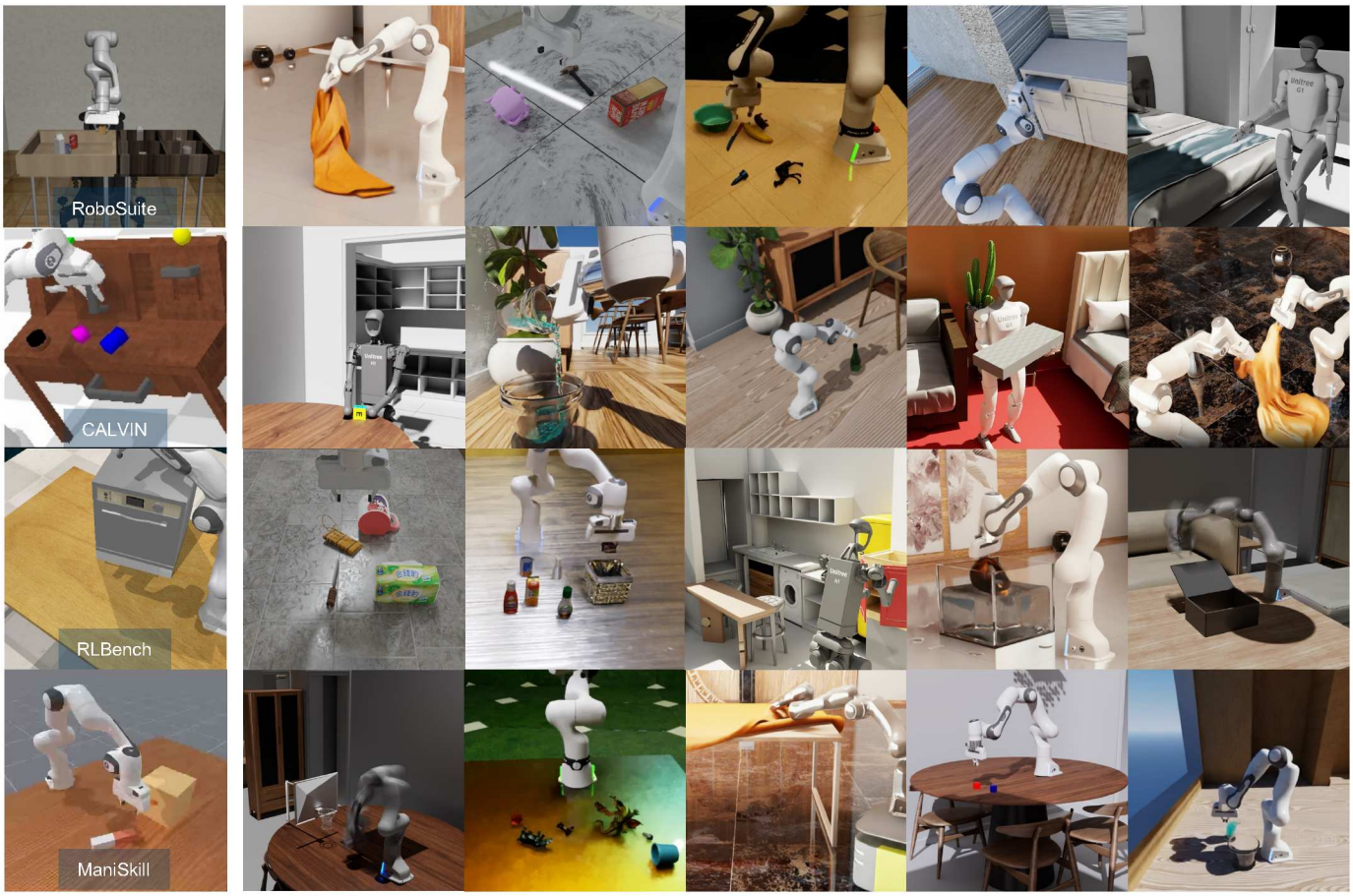


Fig. 8: **Dataset Comparison and Gallery**. Left: other representative synthetic robotics datasets. Right: the RoboVerse dataset.

Source Benchmark	Source Simulator	# Task Categories	# Trajectories	# Assets
ManiSkill [66, 30, 88]	SAPIEN	6	19k	1.7k
RLBench [36]	CoppeliaSim	80	150k	100
CALVIN [64]	Pybullet	7	20k	7
MetaWorld [102]	MuJoCo	5	5k	6
RoboSuite [109]&MimicGen [61]	MuJoCo	6	6k	12
GAPartNet [26]	IsaacGym	4	4k	151
Open6DOR [20]	IsaacGym	69	10k	207
ARNOLD [29]	IsaacSim	6	3k	30
LIBERO [54]	MuJoCo	10	15k	15
Simpler [52]	SAPIEN	6	30k	52
RLAfford [28]	IsaacGym	4	40k	40
GraspNet [23]	-	58	200k	42
GarmentLab [58]	IsaacSim	6	6k	3k
UniDoorManip [53]	IsaacGym	7	1k	140
GAPartManip [16]	IsaacSim	2	1.5k	42
Total	-	276	510.5k	5.5k

TABLE I: Migration progress statistics for manipulation tasks in RoboVerse

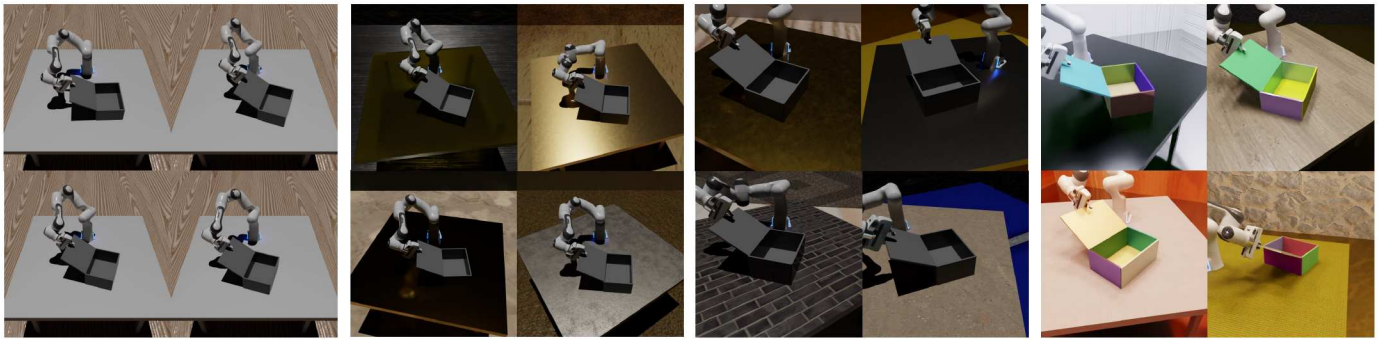
focus on VLN in continuous environments (VLN-CE) [44], as it more closely resembles real-world scenarios [10, 105, 106]. Specifically, we construct our dataset based on RoboVerse by integrating MatterPort 3D scenes [8] (90 scenes) and off-the-shelf instructions from R2R [44] (10k episodes) and RxR [46] (20k episodes). We provide two types of mobile embodiments, including the Unitree Dog (a legged robot) and the JetBot (a wheeled robot), which support different control policies. A detailed elaboration on the navigation dataset is provided in the supplementary materials.

c) Humanoid Dataset: We migrate HumanoidBench [84] tasks for reinforcement learning benchmarks and integrate tasks, policies, and data samples from Humanoid-X [62] and SkillBlender [47]. Additionally, we re-implement the UH-1 inference pipeline within our framework. The pretrained policy successfully enables humanoid robots to follow demonstrated poses while maintaining stable locomotion across multiple simulators based on RoboVerse.

V. RoboVerse Benchmark

A. Benchmark Overview

With the collected tasks, assets, and trajectories, RoboVerse establishes standardized benchmarks for robot learning, including both imitation learning and reinforcement learning. We define a unified training and evaluation protocol within the RoboVerse platform and implement standardized baselines and learning frameworks for benchmarking. Specifically, for imitation learning, we introduce different levels of generalization benchmarks to assess the generalization capability of models.



(a) Level 0 (b) Level 1 (c) Level 2 (d) Level 3

Fig. 9: **Benchmark Protocol:** We define a four-level generalization benchmarking protocol, allocating 90% of the data for training and 10% for generalization evaluation. From left to right, Levels 0 to 3 corresponds to task space generalization, environment radomization, camera randomization, lighting and reflection randomization, respectively.

B. Imitation Learning Benchmark

For each imitation learning benchmark, we establish a standardized evaluation framework with a fixed set of demonstrations and a controlled evaluation environment. Policies must be trained exclusively on the provided training data and assessed within this environment to ensure fair comparison. To rigorously test generalization capability, we curate training data from specific domains and evaluate policies on unseen samples, challenging their adaptability to novel scenarios. We systematically categorize visual generalization factors into multiple levels, including task space generalization, environment setup generalization, camera setting generalization, and lighting and reflection generalization. Each level introduces controlled variations to assess a policy's adaptability and robustness in increasingly diverse and challenging conditions.

a) Level 0: Task Space Generalization: We establish a controlled evaluation by standardizing the environment with consistent camera, materials, lighting, and other parameters. The task space, including object initialization and instructions, is split into 90% training and 10% validation to assess generalization within a fixed setting, as shown in Fig. 9 (a).

b) Level 1: Environment Randomization: Building on the standardized setup, we introduce scene randomization while keeping the camera, materials, and lighting fixed [63]. By varying house, table, and ground configurations, we create diverse visual inputs to test robustness against environmental changes [38]. A fixed set of predefined randomized scenes ensures structured evaluation, as shown in Fig. 9 (b).

c) Level 2: Camera Randomization: To assess generalization across camera variations, we introduce different viewing heights and angles using carefully annotated, realistic camera poses. Following the 90/10 training/testing split, we ensure consistent and rigorous evaluation, as illustrated in Fig. 9 (c).

d) Level 3: Lighting and Reflection Randomization: Real-world environments involve diverse materials and lighting conditions [92]. To simulate these challenges, we randomize lighting and reflections, curating realistic object materials and illumination setups [17]. This enhances robustness testing under varying conditions, as shown in Fig. 9 (d).

C. Reinforcement Learning Benchmark

In addition to imitation learning, RoboVerse offers a comprehensive reinforcement learning (RL) benchmark designed to accommodate a diverse range of tasks, robot embodiments, and simulation backends. Specifically, we integrate the PPO [83] algorithm from both Stable-Baselines3 [76] and rsl_rl [80] into our MetaSim interface, enabling straightforward task definition, seamless environment switching, and standardized performance logging.

Building upon this infrastructure, we have successfully ported multiple humanoid control tasks from the HumanoidBench [84] benchmark into RoboVerse. Through our adapted interface for rsl_rl [80], we have efficiently extended framework compatibility to support the TD-MPC2 [31, 32] algorithm from the original benchmark while preserving implementation fidelity.

VI. Experimental Results

A. Overview

We conduct extensive experiments to validate the effectiveness and practicality of RoboVerse. First, we evaluate baselines on representative tasks from various benchmark sources to ensure the reliability of the collected datasets and established benchmarks. This includes assessments of both imitation learning baselines Sec. VI-B and reinforcement learning baselines Sec. VI-C.

Then we further demonstrate the strength of the high-quality synthetic dataset. We find that synthetic data could significantly boost world model learning.

B. Results on the Imitation Learning Benchmark

1) Baseline and Task Selection: To genuinely reflect the data quality of the RoboVerse dataset and provide a standard benchmark for all kinds of imitation learning policy models,

[1]Due to resource and time constraints, we uniformly sample 20 testing scenarios for the OpenVLA baseline.

Representative Task Benchmark Source		PickCube ManiSkill	StackCube ManiSkill	CloseBox RLBench	MoveSliderLeft CALVIN	PickChocolatePudding LIBERO	NutAssembly RoboSuite	Average -
Diffusion Policy [12]	78M	52.7	53.8	51.5	76.5	50.0	7.1	48.6
ACT [107]	84M	31.7	36.7	68.3	85.0	78.3	0.0	50.0

TABLE II: **Baseline Results on RoboVerse Imitation Learning Benchmark.** We report baseline results on representative tasks from various benchmark sources to validate the effectiveness and reliability of the RoboVerse benchmark.

Task and Generalization Level	MoveSliderLeft				CloseBox				PickCube			
	Level 0	Level 1	Level 2	Level 3	Level 0	Level 1	Level 2	Level 3	Level 0	Level 1	Level 2	Level 3
Diffusion Policy [12]	76.5	81.3	72.0	60.0	51.5	42.8	20.0	10.4	52.7	11.1	0.0	0.0
ACT [107]	85.0	83.3	43.3	16.6	68.3	73.3	0.0	20.0	31.7	30.0	6.7	3.3
OpenVLA[1] [42]	45.0	40.0	35.0	30.0	0.0	0.0	0.0	0.0	40.0	15.0	0.0	0.0

TABLE III: **Generalization Performance on Imitation Learning Benchmark.** This table presents the experimental results for each generalization level in our benchmark across different tasks and methodologies. The tasks are divided into distinct levels (Level 0, Level 1, Level 2, and Level 3) to evaluate performance under progressively challenging scenarios.

Method	Simple		Language-conditioned Grasping		
	PickCube	MoveSliderLeft	Object Set 1	Object Set 2	Object Set 3
OpenVLA [42]	40.0	45.0	46.0	33.3	14.4
Octo [70]	50.0	30.0	42.0	14.4	2.2

TABLE IV: **Vision-Language-Action (VLA) Model Results on RoboVerse Imitation Learning Benchmark.** Constrained with time and resources, we report VLA models' results on two simple tasks from RoboVerse and grasping tasks with diverse and challenging language instructions. We split 58 objects in GraspNet into three sets, each containing progressively more challenging objects based on their geometry.

we select both prevailing specialist and generalist models as baselines of our RoboVerse benchmark. Specifically, for specialist models, we integrate ACT [107] and Diffusion Policy [12]. For generalist models, We benchmark our approach on OpenVLA [42] and Octo [70], both of which we fine-tuned using our synthetic dataset. ACT is one of the most widely used methods in bi-manual manipulation. Diffusion Policy [12] is the first work that applies the conditional denoising diffusion process as a robot visuomotor policy and achieves great generalization capabilities.

Leveraging the RoboVerse format and infrastructure design, we are able to evaluate models on different tasks within a unified platform. To fully test policy models' performance under versatile settings, we select one representative task from each of the source benchmarks integrated by the RoboVerse dataset as shown in Tab. II. The experiment subset includes `PickCube` and `StackCube` from ManiSkill [66], `CloseBox` from RLBench [36], `MoveSliderLeft` from CALVIN [64], `PickChocolatePudding` from LIBERO [54], and `NutAssembly` from `robosuite` [109]. These tasks not only demand precise pick-and-place skills but also require contact-rich physical interactions with articulated objects. Through these tasks, the benchmark results can provide a comprehensive reflection of each model's performance under different scenarios.

2) Implementation Details: Due to time and resource constraints, we implement specialist and generalist models using different strategies, and all the results are obtained under the single-task setting. The training and evaluation settings follow the 90/10 RoboVerse benchmark protocol as specified in Sec. V-B. During evaluations, we randomly select ten task settings from training sets and another ten from the validation sets. The reported success rates are computed as the averages over three random seeds.

For each step, the inputs are $256 \times 256 \times 3$ RGB images and a short language description depending on the task settings. For specialist models, we train from scratch with action in 9-dim robot joint state space. For generalist models, the action is pre-processed into delta end-effector position space from absolute end-effector position space, and The gripper action is discretized into binary values $\{0, +1\}$. Owing to the lack of time and resources, we are only able to fine-tune the generalist models in the single-task setting. During evaluations, we employ cuRobo [86] as the inverse-kinematics solver to transform the action to robot joint state space. Specific model implementation details and hyperparameters are provided in supplementary materials.

3) Experiment Results: We present the imitation learning benchmark results in Tab. II and the generalization evaluation in Tab. III. We further fine-tune large vision-language-action models on both simple and complex language-conditioned tasks, as shown in Tab. IV.

C. Results on the Reinforcement Learning Benchmark

Using Stable-Baselines3 [76] and rsl_rl [80] implementations of PPO, we train policies on tasks from IsaacLab [65] under consistent hyperparameters.

For additional tasks (humanoid, dexterous hand), the same PPO-based workflow applies. We successfully migrate the HumanoidBench [84] from MuJoCo to RoboVerse, enabling training across multiple simulators (Isaac Sim and MuJoCo) with consistent interfaces. Experiment results demonstrate

stable policy convergence across simulators, achieving comparable performance to native MuJoCo baselines. Leveraging the generalizability of rsl_rl [80], we further extend the benchmark to support TD-MPC2 [31, 32] algorithm , which exhibits robust training dynamics in all environments. For implementation details, reward curve, and extended experimental results, please refer to the supplementary materials.

D. Augmentation Experiments

To verify the effectiveness of our trajectory augmentation API, on four representative tasks, we compare the success rates of trained Diffusion Policy on 50 source demonstrations and 200, 1000, and 3000 generated augmentation demonstrations under the imitation learning setting. The results presented in Fig. 10 demonstrate a consistent improvement in model performance as the number of generated data increases, highlighting both the effectiveness and scalability of the trajectory augmentation API.

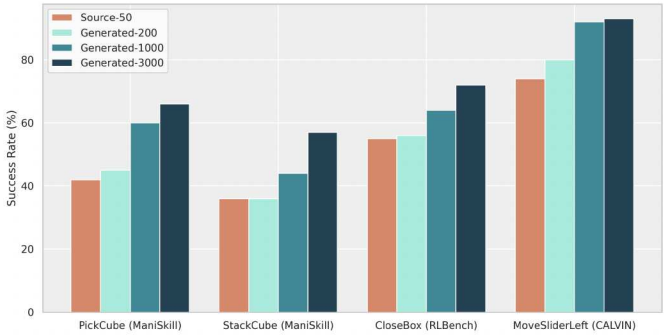


Fig. 10: **Effectiveness of Trajectory Augmentation.** Success rates of policy trained with augmented dataset and source dataset.

E. World Model Learning

Recent advances in general-purpose video generation and interactive world models [89, 6] have shown promising progress. Yet, the scarcity of gigantic-scale robotic datasets still impedes the development of robust world models for a wide range of robotic applications. In this session, we demonstrate how synthetic data from the RoboVerse simulation can augment real-world datasets to train more capable robotics world models.

When a model is trained exclusively on 50,000 episodes from the DROID dataset [41], it generally respects action conditions but struggles to accurately capture physical interactions between the gripper and target objects. Notably, the objects appear "warped" during contact with the gripper, as shown in Fig. 11. By incorporating an additional 50,000 synthetic episodes from RoboVerse to create a combined dataset of 100,000 episodes, the model predictions improve with regard to preserving object geometry. However, merely "watching videos" remains insufficient for learning the intricate physical interactions in DROID.

In contrast, training solely on the RoboVerse-50K or on the DROID-RoboVerse-100K dataset and then validating on

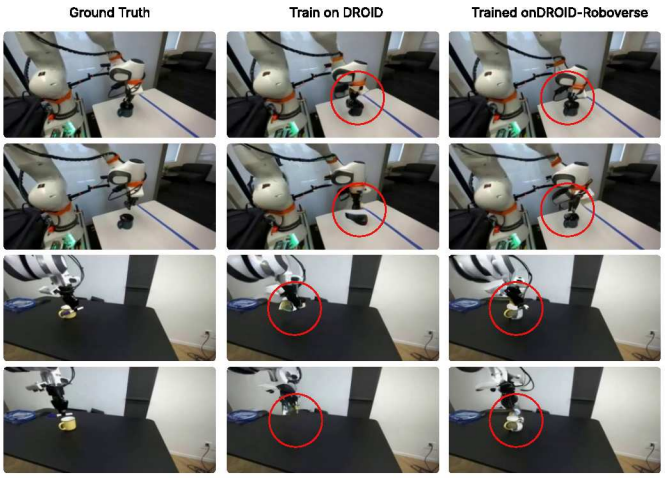


Fig. 11: **Ablation Study of Action-conditioned World Model Learning.** We compare the qualitative results of an action-conditioned world model trained on pure DROID and DROID-RoboVerse datasets, with evaluations sampled from the DROID dataset.

RoboVerse samples, we observe that the generated frames are physically more realistic in most scenes, with details in the supplementary materials. This improvement can be attributed to the extensive randomization and augmentation available in RoboVerse. Conversely, a model trained solely on DROID data fails to transfer effectively to the RoboVerse scene. We hypothesize that this shortcoming stems from limited samples per scene coverage in DROID and incomplete gripper visibility in the camera view.

F. Imitating the RoboVerse Dataset Enables Direct Sim-to-Real Transfer

The RoboVerse system seamlessly integrates a powerful physics engine with a high-quality renderer, ensuring the generation of realistic, high-fidelity data. To demonstrate its potential, we conduct experiments validating its effectiveness in direct sim-to-real transfer. As shown in Fig. 12, we fine-tune OpenVLA [42] on the RoboVerse dataset and transfer the learned policy to real-world scenarios without additional fine-tuning. The model successfully manipulates unseen objects in previously unseen real-world environments, showcasing the robustness and generalization capabilities of our system. The quantitative results on more challenging language-guided tasks, as shown in Tab. V, further demonstrate the high success rate of models trained on the RoboVerse dataset. Additional details are provided in the supplementary materials.

G. Reinforcement Learning in RoboVerse Enables Sim-to-Sim-to-Real Transfer

Large-scale parallel environments offer significant potential for large-scale exploration and are highly effective for reinforcement learning (RL) tasks. However, while they provide excellent efficiency, their accuracy may be limited in certain scenarios [21]. To address this problem, Sim-to-sim evaluation

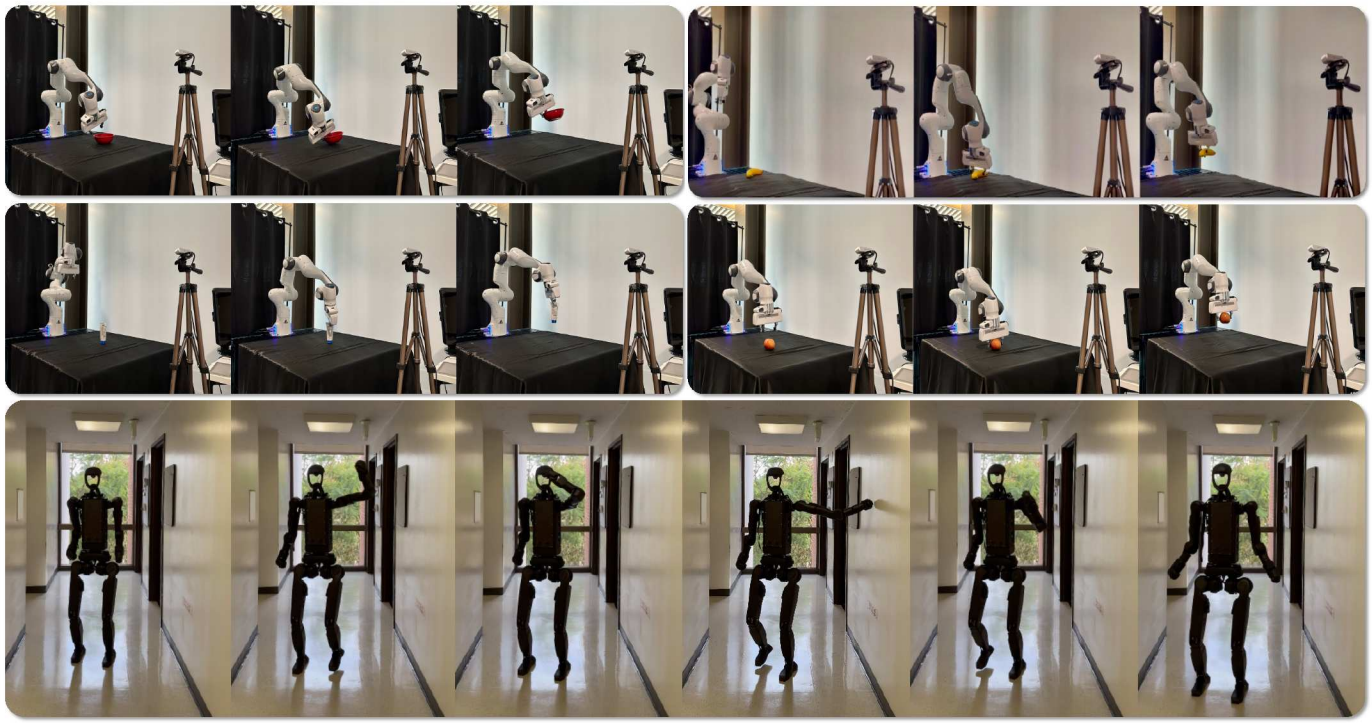


Fig. 12: **Sim-to-Real and Sim-to-Sim-to-Real Experiment Results.** We demonstrate that learning within the RoboVerse framework enables seamless direct Sim-to-Real transfer for manipulating unseen objects in new environments (imitation learning) and Sim-to-Sim-to-Real transfer for whole-body humanoid control (reinforcement learning).

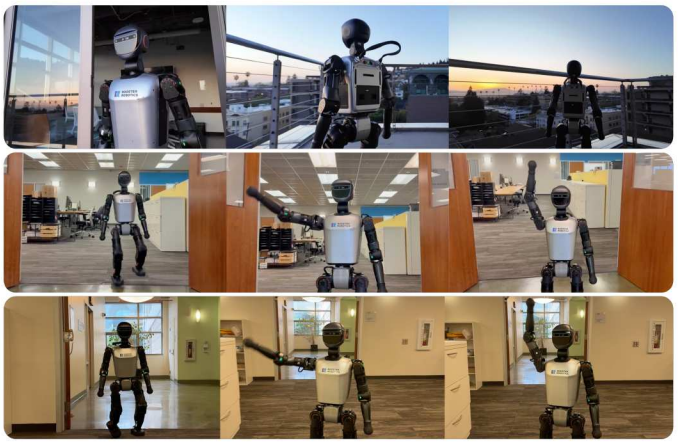


Fig. 13: **Generalization of Sim-to-Sim-to-Real.** This figure shows the in-the-wild generalization ability of our lower-body RL policy with upper-body PD control by the sim-to-sim-to-real approach.

and fine-tuning present promising solutions [52]. As shown in Fig. 13, RoboVerse platform seamlessly supports such functionalities, enabling robust sim-to-sim and sim-to-real transitions. We further demonstrate the effectiveness of sim-to-sim-to-real generalization through comprehensive experiments, highlighting the platform's ability to bridge simulation and real-world performance.

GraspNet Objects	Pick up Wash Soap	Lift Mouth Rinse	Grasp Green Dish
Octo [70]	5.0/10.0	3.0/10.0	6.0/10.0
OpenVLA [42]	7.0/10.0	8.0/10.0	5.0/10.0

TABLE V: **Direct Sim-to-Real.** We fine-tune two baseline models using demonstrations adapted from GraspNet [23] to validate the effectiveness of the RoboVerse dataset. The final performance score for each task is reported, where a baseline receives 1 point for successfully grasping the target. Additionally, we adopt the partial reward scheme from OpenVLA [42], awarding 0.5 points when the gripper makes contact with the target.

VII. Limitations

While RoboVerse provides a comprehensive and scalable platform, several limitations remain. First, the integration of a unified format for non-rigid objects is not yet fully supported, which we leave for future work to develop. Additionally, while our large-scale dataset presents significant potential for pretraining a foundation model, this exploration falls beyond the scope of this paper due to resource constraints. Furthermore, despite our extensive efforts to fully reimplement and optimize all baseline methods within the RoboVerse baselines, some implementations may still be suboptimal. Our primary goal is not to directly compare policy performance but to demonstrate

that the system is comprehensive, supports diverse policies, and ensures strong alignment between simulation and real-world performance. While we have made every effort to build a robust platform, it is inevitable that some oversights or errors may remain. We encourage the broader research community to contribute to maintaining and refining the baselines, fostering collaboration to further enhance the platform's capabilities.

ACKNOWLEDGEMENT

We thank Hanyang Zhou and Sicheng He for providing valuable suggestions for setting up robotics hardware. We thank Yufeng Chi and Sophia Shao for providing humanoid robots for testing. We thank Jie Yang and Muzhi Han for valuable discussion. We thank Koushil Sreenath for insightful feedback. We thank Jiawei Yang, Sumeet Batra, and Gaurav Sukhatme for their generous help. Pieter Abbeel holds concurrent appointments as a professor at UC Berkeley and as an Amazon Scholar. This paper describes work performed at UC Berkeley and is not associated with Amazon.

REFERENCES

[1] Unai Antero, Francisco Blanco, Jon Oñativia, Damien Sallé, and Basilio Sierra. Harnessing the power of large language models for automated code generation and verification. *Robotics*, 2024.

[2] Genesis Authors. Genesis: A universal and generative physics engine for robotics and beyond, 2024. URL https://github.com/Genesis-Embodied-AI/Genesis.

[3] Tamir Blum, Gabin Paillet, Mickael Laine, and Kazuya Yoshida. Rl star platform: Reinforcement learning for simulation based training of robots. *arXiv preprint arXiv:2009.09595*, 2020.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[6] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments, 2024.

[7] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*, 2015.

[8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *International Conference on 3D Vision (3DV)*, 2017.

[9] Yuanpei Chen, Chen Wang, Yaodong Yang, and Karen Liu. Object-centric dexterous manipulation from human motion data. In *Conference on Robot Learning (CoRL)*, 2024.

[10] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024.

[11] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 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 (RSS)*, 2023.

[13] Alberto Silvio Chiappa, Alessandro Marin Vargas, Ann Huang, and Alexander Mathis. Latent exploration for reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[14] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models, 2023.

[15] Erwin Coumans and Yunfei Bai. PyBullet, a python module for physics simulation for games, robotics and machine learning, 2016–2021.

[16] Wenbo Cui, Chengyang Zhao, Songlin Wei, Jiazhao Zhang, Haoran Geng, Yaran Chen, and He Wang. GAPartManip: A large-scale part-centric dataset for material-agnostic articulated object manipulation. *arXiv preprint arXiv:2411.18276*, 2024.

[17] Qiyu Dai, Jiyao Zhang, Qiwei Li, Tianhao Wu, Hao Dong, Ziyuan Liu, Ping Tan, and He Wang. Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects. In *European Conference on Computer Vision (ECCV)*, 2022.

[18] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[20] Yufei Ding, Haoran Geng, Chaoyi Xu, Xiaomeng Fang, Jiazhao Zhang, Songlin Wei, Qiyu Dai, Zhizheng Zhang, and He Wang. Open6DOR: Benchmarking open-instruction 6-dof object rearrangement and a vlm-based approach. In *International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[21] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning, 2019.

[22] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *International Conference on Robotics and Automation (ICRA)*, 2015.

[23] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. GraspNet-1Billion: A large-scale benchmark for general object grasping. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[24] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[25] Caelan Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. *arXiv preprint arXiv:2410.18907*, 2024.

[26] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[27] Haoran Geng, Songlin Wei, Congyue Deng, Bokui Shen, He Wang, and Leonidas Guibas. SAGE: Bridging semantic and actionable parts for generalizable articulated-object manipulation under language instructions, 2024.

[28] Yiran Geng, Boshi An, Haoran Geng, Yuanpei Chen, Yaodong Yang, and Hao Dong. RLAfford: End-to-end affordance learning for robotic manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2023.

[29] Ran Gong, Jiangyong Huang, Yizhou Zhao, Haoran Geng, Xiaofeng Gao, Qingyang Wu, Wensi Ai, Ziheng Zhou, Demetri Terzopoulos, Song-Chun Zhu, et al. ARNOLD: A benchmark for language-grounded task learning with continuous states in realistic 3d scenes. In *International Conference on Computer Vision (ICCV)*, 2023.

[30] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. ManiSkill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.

[31] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, 2022.

[32] Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

[33] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[34] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. 2024.

[35] Marcel Hussing, Jorge A Mendez, Anisha Singrodia, Cassandra Kent, and Eric Eaton. Robotic manipulation datasets for offline compositional reinforcement learning. *arXiv preprint arXiv:2307.07091*, 2023.

[36] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *Robotics and Automation Letters (RA-L)*, 2020.

[37] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu.

DexMimicGen: Automated data generation for bimanual dexterous manipulation via imitation learning. *arXiv preprint arXiv:2410.24185*, 2024.

[38] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *Robotics and Automation Letters (RA-L)*, 2020.

[39] Pushkal Katara, Zhou Xian, and Katerina Fragkiadaki. Gen2sim: Scaling up robot learning in simulation with generative models. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[40] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 2023.

[41] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *Robotics: Science and Systems (RSS)*, 2024.

[42] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[43] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *International Conference on Computer Vision (ICCV)*, 2023.

[44] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, 2020.

[45] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[46] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[47] Yuxuan Kuang, Amine Elhafsi, Haoran Geng, Marco Pavone, and Yue Wang. SkillBlender: Towards versatile humanoid whole-body control via skill blending. In *CoRL 2024 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*.

[48] Yuxuan Kuang, Junjie Ye, Haoran Geng, Jiageng Mao, Congyue Deng, Leonidas Guibas, He Wang, and Yue Wang. RAM: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation, 2024.

[49] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. BEHAVIOR-1K: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning (CoRL)*, 2022.

[50] Puhao Li, Tengyu Liu, Yuyang Li, Muzhi Han, Haoran Geng, Shu Wang, Yixin Zhu, Song-Chun Zhu, and Siyuan Huang. Ag2Manip: Learning novel manipulation skills with agent-agnostic visual and action representations. *arXiv preprint arXiv:2404.17521*, 2024.

[51] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. ManipLLM: Embodied multimodal large language model for object-centric robotic manipulation, 2024.

[52] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[53] Yu Li, Xiaojie Zhang, Ruihai Wu, Zilong Zhang, Yiran Geng, Hao Dong, and Zhaofeng He. UniDoorManip: Learning universal door manipulation policy over large-scale and diverse door manipulation environments. *arXiv preprint arXiv:2403.02604*, 2024.

[54] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[55] Ruoshi Liu, Alper Canberk, Shuran Song, and Carl Vondrick. Differentiable robot rendering, 2024. URL https://arxiv.org/abs/2410.13851.

[56] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[57] Haozhe Lou, Yurong Liu, Yike Pan, Yiran Geng, Jianteng Chen, Wenlong Ma, Chenglong Li, Lin Wang, Hengzhen Feng, Lu Shi, Liyi Luo, and Yongliang Shi. Robo-GS: A physics consistent spatial-temporal model for robotic arm with hybrid representation, 2024.

[58] Haoran Lu, Ruihai Wu, Yitong Li, Sijie Li, Ziyu Zhu, Chuanruo Ning, Yan Shen, Longzan Luo, Yuanpei Chen, and Hao Dong. GarmentLab: A unified simulation and benchmark for garment manipulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[59] Jiangran Lyu, Yuxing Chen, Tao Du, Feng Zhu, Huiquan Liu, Yizhou Wang, and He Wang. Scissorbot: Learning generalizable scissor skill for paper cutting

via simulation, imitation, and sim2real. *arXiv preprint arXiv:2409.13966*, 2024.

[60] 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, 2021.

[61] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning (CoRL)*, 2023.

[62] Jiageng Mao, Siheng Zhao, Siqi Song, Tianheng Shi, Junjie Ye, Mingtong Zhang, Haoran Geng, Jitendra Malik, Vitor Guizilini, and Yue Wang. Learning from massive human videos for universal humanoid pose control. *arXiv preprint arXiv:2412.14172*, 2024.

[63] Pablo Martinez-Gonzalez, Sergiu Oprea, Alberto Garcia-Garcia, Alvaro Jover-Alvarez, Sergio Orts-Escolano, and Jose Garcia-Rodriguez. Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. *Virtual Reality*, 2020.

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

[65] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *Robotics and Automation Letters (RA-L)*, 2023.

[66] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. ManiSkill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021.

[67] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

[68] NVIDIA. vMaterials, 2024. URL https://developer.nvidia.com/vmaterials.

[69] NVIDIA. Isaac sim simulator, 2025. URL https://developer.nvidia.com/isaac/sim.

[70] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems (RSS)*, 2024.

[71] Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P Schoellig. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. In *International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[72] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[73] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. ShapeLLM: Universal 3D object understanding for embodied interaction. *European Conference on Computer Vision (ECCV)*, 2024.

[74] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dietor Fox. AnyTeleop: A general vision-based dexterous robot arm-hand teleoperation system. 2023.

[75] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[76] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 2021.

[77] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

[78] Pengzhen Ren, Min Li, Zhen Luo, Xinshuai Song, Ziwei Chen, Weijia Liufu, Yixuan Yang, Hao Zheng, Rongtao Xu, Zitong Huang, et al. InfiniteWorld: A unified scalable simulation framework for general visual-language robot interaction. *arXiv preprint arXiv:2412.05789*, 2024.

[79] E. Rohmer, S. P. N. Singh, and M. Freese. CoppeliaSim (formerly V-REP): a versatile and scalable robot simulation framework. In *International Conference on Intelligent Robots and Systems (IROS)*, 2013. URL www.coppeliarobotics.com.

[80] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning, 2022. URL https://arxiv.org/abs/2109.11978.

[81] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[82] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Confer-*

ence on Computer Vision (ECCV), 2016.

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

[84] Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation, 2024.

[85] Arth Shukla, Stone Tao, and Hao Su. ManiSkill-HAB: A benchmark for low-level manipulation in home rearrangement tasks, 2024. URL https://arxiv.org/abs/2412.13211.

[86] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. cuRobo: Parallelized collision-free minimum-jerk robot motion generation, 2023.

[87] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[88] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. ManiSkill3: GPU parallelized robotics simulation and rendering for generalizable embodied AI. *arXiv preprint arXiv:2410.00425*, 2024.

[89] Movie Gen team. Movie gen: A cast of media foundation models, 2024.

[90] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[91] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

[92] Christine M Vaccaro, Catrina C Crisp, Angela N Fellner, Christopher Jackson, Steven D Kleeman, and James Pavelka. Robotic virtual reality simulation plus standard robotic orientation versus standard robotic orientation alone: a randomized controlled trial. *Urogynecology*, 2013.

[93] Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović. Practical motion capture in everyday surroundings. *ACM Transactions on Graphics (TOG)*, 2007.

[94] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. UniDexGrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. *arXiv preprint arXiv:2304.00464*, 2023.

[95] Hanqing Wang, Jiahe Chen, Wensi Huang, Qingwei Ben, Tai Wang, Boyu Mi, Tao Huang, Siheng Zhao, Yilun Chen, Sizhe Yang, et al. Grutopia: Dream general robots in a city at scale. *arXiv preprint arXiv:2407.10943*, 2024.

[96] Jun Wang, Yuzhe Qin, Kaiming Kuang, Yigit Korkmaz, Akhilan Gurumoorthy, Hao Su, and Xiaolong Wang. CyberDemo: Augmenting simulated human demonstration for real-world dexterous manipulation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[97] Songlin Wei, Haoran Geng, Jiayi Chen, Congyue Deng, Cui Wenbo, Chengyang Zhao, Xiaomeng Fang, Leonidas Guibas, and He Wang. D3RoMa: Disparity diffusion-based depth sensing for material-agnostic robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

[98] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[99] Xintong Yang, Ze Ji, Jing Wu, and Yu-Kun Lai. An open-source multi-goal reinforcement learning environment for robotic manipulation with pybullet. In *Annual Conference Towards Autonomous Robotic Systems*, 2021.

[100] Chongjie Ye, Yinyu Nie, Jiahao Chang, Yuantao Chen, Yihao Zhi, and Xiaoguang Han. GauStudio: A modular framework for 3D gaussian splatting and beyond. *arXiv preprint arXiv:2403.19632*, 2024.

[101] Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. StableNormal: Reducing diffusion variance for stable and sharp normal. *ACM Transactions on Graphics (TOG)*, 2024.

[102] 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 (CoRL)*, 2019.

[103] Kevin Zakka, Baruch Tabanpour, Qiayuan Liao, Mustafa Haiderbhai, Samuel Holt, Jing Yuan Luo, Arthur Allshire, Erik Frey, Koushil Sreenath, Lueder A. Kahrs, Carlo Sferrazza, Yuval Tassa, and Pieter Abbeel. MuJoCo playground: An open-source framework for GPU-accelerated robot learning and sim-to-real transfer., 2025. URL https://github.com/google-deepmind/mujoco_playground.

[104] Jialiang Zhang, Haoran Liu, Danshi Li, XinQiang Yu, Haoran Geng, Yufei Ding, Jiayi Chen, and He Wang. DexGraspNet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *Confer-*

ence on Robot Learning (CoRL), 2024.

[105] Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024.

[106] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *Robotics: Science and Systems (RSS)*, 2024.

[107] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[108] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022.

[109] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.