AgentWorld: An Interactive Simulation Platform for Scene Construction and Mobile Robotic Manipulation

Yizheng Zhang[1*], Zhenjun Yu[2*], Jiaxin Lai[1], Cewu Lu[2], Lei Han[1 §]
[1]Tencent Robotics X, [2]Shanghai Jiao Tong University
[1]zyz.robotics@gmail.com, jesselai@tencent.com, leihan.cs@gmail.com
[2]{jeffson-yu, lucewu}@sjtu.edu.cn

Figure 1: **Overview of AgentWorld.** AgentWorld simulation platform features several core abilities for embodied AI: (1) Procedural scene construction supporting various layout generation. (2) Abundant semantic 3D assets repository with realistic visual material and physical properties. (3) Mobile-based Teleoperation system for robotic manipulation.

Abstract: We introduce *AgentWorld*, an interactive simulation platform for developing household mobile manipulation capabilities. Our platform combines automated scene construction that encompasses layout generation, semantic asset placement, visual material configuration, and physics simulation, with a dual-mode teleoperation system supporting both wheeled bases and humanoid locomotion policies for data collection. The resulting *AgentWorld Dataset* captures diverse tasks ranging from primitive actions (pick-and-place, push-pull, etc.) to multistage activities (serve drinks, heat up food, etc.) across living rooms, bedrooms, and kitchens. Through extensive benchmarking of imitation learning methods including behavior cloning, action chunking transformers, diffusion policies, and vision-language-action models, we demonstrate the dataset's effectiveness for sim-to-real transfer. The integrated system provides a comprehensive solution for scalable robotic skill acquisition in complex home environments, bridging the gap between simulation-based training and real-world deployment. The code, datasets will be available at https://yizhengzhang1.github.io/agent_world/

Keywords: Simulation Platform, scene construction, teleoperation

---

* indicates equal contributions. § Lei Han is the corresponding author.

1 Introduction

Recent advancements in embodied AI and robotic manipulation have highlighted the need for scalable, interactive simulation environments that support both scene construction and data collection for training autonomous agents. While existing platforms [1, 2, 3, 4] offer partial solutions, such as scene generation[5, 3, 4] or task-specific manipulation datasets [6, 7, 8], few provide a unified framework that integrates high-fidelity scene construction with flexible mobile robotic data collection system. To bridge this gap, we present **AgentWorld**, an interactive simulation platform designed for **procedural scene construction** and **mobile-based teleoperation**, enabling efficient data collection for imitation learning in complex household environments.

AgentWorld addresses two critical challenges in embodied AI research: (1) **stable and diverse scene generation**, ensuring that the simulated environments are visually realistic and physically plausible, and (2) **a comprehensive data collection system** in simulation, which allows seamless control of mobile bases and robotic arms for data collection. AgentWorld is built upon NVIDIA's Omniverse Isaac Sim [9] and Unreal Engine [10], allowing it to inherit both strengths including the physics engine for robot parallel training and realistic rendering effects.

For procedural construction of diverse household scenes, our platform supports four key components: (i) **Layout Generation**, which procedurally generates different room layouts with walls, stairs and floors; (ii) **Semantic Asset Selection and Placement**, leveraging a rich 3D asset repository and arranging objects in semantically meaningful configurations; (iii) **Visual Material Configuration**, enabling fine-grained semantic object material appearances; and (iv) **Interactive Physics Simulation**, ensuring dynamic interactions of robots and assets remain stable and accurate.

Beyond scene construction, AgentWorld introduces a **Mobile-based Data Collection system** that combines keyboard-controlled mobile base navigation with arm and hand manipulation. For facilitating the collection of diverse robotic interaction data for both wheel-based and legged robots, we implement a reinforcement learning method for lower-limb control of humanoid robots. A VR-based teleoperation system is employed for arm control and hand keypoint detection, and we use a retargeting approach for operating both grippers and dexterous hands.

We validate our platform by constructing the AgentWorld Dataset, in which we generate 150 household scenes with more than 9000 assets and 4 different embodiments with either gripper or dexterous hand as end effectors. Utilizing our data collection system, we collect more than 1000 robot manipulation trajectories with tasks from simple object interactions to long-horizon, multistage challenges. Using imitation learning algorithms (BC, ACT[11], Diffusion Policy[12], and π_0 [13]), we demonstrate successful policy training and sim-to-real transfer with few-shot real data finetuning, highlighting the utility of our dataset for real-world robotic applications.

Our primary contributions include:

(1) A procedural scene construction framework supporting diverse layout generation and semantic object placement, features by an extensive 3D asset library with realistic visual and physical properties, enhancing simulation fidelity.

(2) A mobile teleoperation system for data collection, enabling precise control of both mobile bases and manipulators with grippers and dexterous hands.

(3) The AgentWorld Dataset, a large-scale benchmark for robotic manipulation, was validated through imitation learning and sim-to-real experiments.

2 Related Work

Simulation Platform for Embodied AI   Recent progress in embodied AI has been driven by simulation platforms that balance realism, scalability, and task diversity. [14, 15] establish robotic simulation environments with Unity and extend the Python-Unity communication interface, while still lacking programmatic scene construction and accurate physics simulation. [2] employs proce-

dural generation to create large-scale interactive environments, while [16, 17] focus on manipulation tasks with physically realistic interactions. Similarly, [18] provides a benchmark for robotic learning with a suite of predefined tasks, and [19] extends scene complexity by simulating human-like activities. [20] introduces a social interaction layer, enabling NPC-driven task allocation, and [21] emphasizes household robotics with large-scale object interactions. While these platforms excel in specific domains, AgentWorld distinguishes itself by integrating procedural scene construction with mobile teleoperation, offering a more holistic approach to embodied AI research.

Scene Construction Effective scene construction requires both geometric diversity and physical realism. Prior works [5, 3, 1, 4] have explored procedural generation with primitive methods or large language models. However, many existing systems rely on static assets with limited material customization. SceneCAD [22] introduces a data-driven approach for indoor scene synthesis, while 3DTopia-XL [23] leverages generative models for diverse 3D asset creation. In contrast, Agent-World introduces a dynamic asset repository with PBR materials, ensuring realistic visual and physical behaviors. Our platform also supports automated layout generation, allowing users to define semantic rules for object arrangement.

Data Collection for robotic manipulation Data collection methodologies in robotics have evolved from scripted demonstrations [24] to human teleoperation [25]. While [11] synthesizes manipulation data via motion retargeting, it lacks mobile base integration. AgentWorld advances this domain by combining keyboard-controlled mobility with arm and hand teleoperation, enabling the collection of long-horizon, multistage tasks. Our approach aligns with recent trends in imitation learning [12, 26] while addressing the sim-to-real gap through physics-aware simulation.

3   AgentWorld Simulation Platform

Name	Num of Assets	Asset		Robotic Platforms			Data Collection		
		Material Selection	Physics Config	Fixed-B	Mobile-B	Legged	Tele-Operation	Dexterous Hand	Num of Trajectories
Maniskill2[17]	2144	✓	✓	✓	✓	✗	✗	✗	30k
ProcTHOR[2]	3578	✓	✗	✗	✓	✗	✗	✗	–
RLBench[18]	28	✗	✓	✓	✗	✗	✗	✗	–
BiGym[7]	<200	✗	✓	✓	✓	✓	J + FB	✗	>2000
Behavior-1K[19]	5215	✓	✓	✗	✓	✗	✗	✗	–
MimicGen[6]	40	✗	✓	✓	✓	✓	✗	✗	50k
RoboCasa[21]	2509	✗	✓	✓	✓	✓	J + FB	✗	100k
InfiniteWorld[5]	>10000	✗	✓	✓	✓	✓	✗	✗	–
GRUtopia[20]	≈25000	✗	✓	✓	✓	✓	FB	✗	–
AgentWorld	>9000	✓	✓	✓	✓	✓	J + FB + L	✓	>1000

Table 1: Comparison of robotic simulation platforms in terms of asset properties, robotic platform support, and data collection capabilities. Fixed-B and Mobile-B stands for fixed and mobile base robots. The teleoperation colomn demonstrates support for joint action control (J), floating base control (FB), and locomotion control (L) for humanoid robots. **AgentWorld** represents our proposed platform integrating all key capabilities.

We introduce the AgentWorld simulation platform, which mainly consists of two parts: **Procedural Scene Generation**, and **Data Collection with Teleoperation**. Our simulation system can programmatically construct various types of household scene with abundant 3D assets, and provides a VR-based data collection system for recording human-demonstrate manipulation data. We compare the AgentWorld simulation platform with popular platforms in Tab. 1.

3.1   Procedural Scene Generation

To construct large-scale, realistic simulation environments for household scenarios and tasks, we employ a programmatic approach with an interactive interface for scene generation. As illustrated
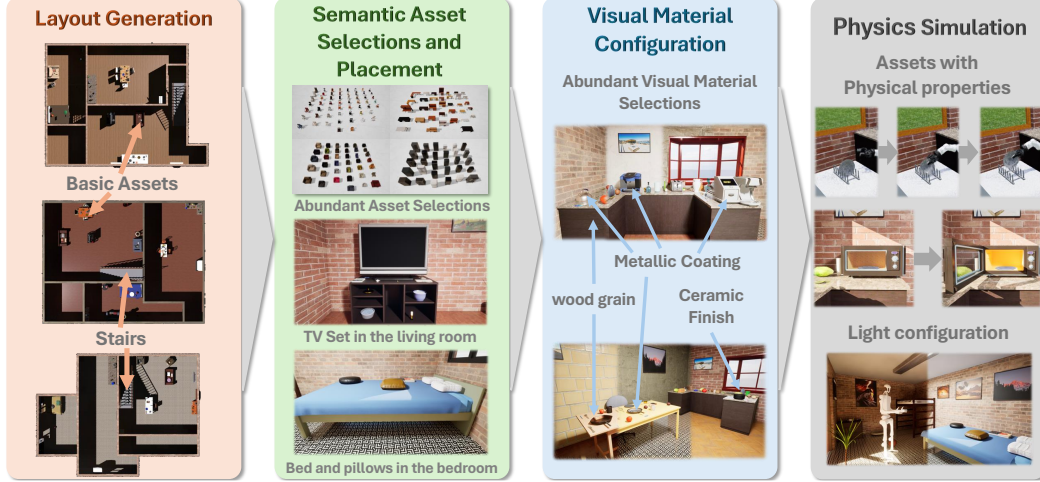
Figure 2: Pipeline of the scene construction module in AgentWorld.

in Fig. 2, our pipeline mainly comprises four key stages: **Layout Generation**, **Semantic Asset Selections and Placement**, **Visual Material Configuration**, and **Interactive Physics Simulation**. The first three stages are implemented in Unreal Engine [10], leveraging its superior rendering capabilities and intuitive blueprint system for efficient scene design. For the Interactive Physics Simulation stage, we utilize NVIDIA's Omniverse Isaac Sim [9], which provides enhanced physics engine performance crucial for sim-to-real transfer and enables efficient parallel training of robotic agents.

3.1.1 Layout Generation

The first stage of our pipeline generates complete room layouts based on user-specified room types. Our system currently supports three fundamental room categories: *Living room*, *Kitchen*, and *Bedroom*. The procedural generation module automatically constructs architectural elements including walls, ceilings, and floors, with support for both single-room configurations and multi-room combinations. Additionally, as demonstrated in Fig. 2, our framework can generate multi-floor environments interconnected by staircases.

Following the architectural layout, the system populates the scene with category-appropriate functional assets such as furniture and decorative elements. The asset selection mechanism and its underlying methodology are detailed in the following section.

3.1.2 Semantic Asset Selections and Placement

To enable diverse scene construction, our simulation platform integrates a comprehensive collection of open-source 3D assets sourced from repositories including ProcTHOR [2] and Behavior-1K [19]. Each asset has been manually annotated with semantic metadata describing its functional properties. We classify these assets into two primary categories: **Basic Assets** and **Interactable Assets**.

Basic Assets constitute the fundamental elements of room layouts and are automatically generated during the initial scene construction phase with contextually appropriate placements. This category encompasses room-specific furnishings, such as sofas and TV sets in the living room, beds and chairs in the bedroom, and tables in the kitchen, etc.

Interactable Assets represent objects that serve functional purposes and can be manipulated by robotic agents. Our scene generation module supports dynamic addition and adjustment of these assets during the second construction phase. We further distinguish several object types: articulated assets like microwaves, refrigerators and closets that have doors, and rigid objects such as household tools (knives, broomsticks, etc.) and common items like fruits, bottles and toys, etc. Leveraging this

extensive interactable asset repository enables the simulation of numerous household manipulation tasks, which we detail in Section 4.1.

Our system incorporates intelligent placement algorithms that position assets according to their semantic functions (e.g., placing food items on dining surfaces, pillows on bedding). The interactive interface provides fine-grained control over asset positioning and orientation adjustments.

3.1.3 Visual Material Configuration

To better facilitate the sim-to-real paradigm, our platform incorporates a diverse selection of visual materials to enhance scene generalization through effective data augmentation. We implement an extensive library of high-fidelity Physically Based Rendering (PBR) materials covering various categories of common household surfaces. As shown in Fig. 2, our system supports material customization at multiple levels: for architectural components like walls and floors, we provide realistic material options such as marble and brick textures; for 3D assets, we enable property-appropriate material adjustments including wood grains, ceramic finishes, fabric textures, and metallic coatings. This comprehensive material system allows for precise visual customization while maintaining physical accuracy in rendering, significantly expanding the variety of achievable visual appearances for robust robotic training.

3.1.4 Interactive Physics Simulation

The Interactive Physics Simulation module implements AgentWorld's robotic interaction layer using NVIDIA Isaac Sim with its GPU-accelerated PhysX 5.0 engine. Our system automatically configures physical properties for all 3D assets based on semantic annotations, generating optimized collision primitives (convex hulls, convex decomposition or SDF meshes, etc.) and assigning material properties including friction coefficients, restitution values, and mass distributions.

For articulated mechanisms such as doors and drawers, the system precisely configure joint parameters including: joint types (revolute for cabinets/doors, prismatic for drawers), movement ranges, material-specific friction coefficients (wood: 0.4 ± 0.1, metal: 0.2 ± 0.05), and actuation models. The simulation environment incorporates Isaac Sim's advanced lighting system, providing photometrically accurate illumination (50 - $20,000$ lux) with programmable color temperature (2700K - 6500K) and light exposure (-5.0 - 5.0) through various light sources.

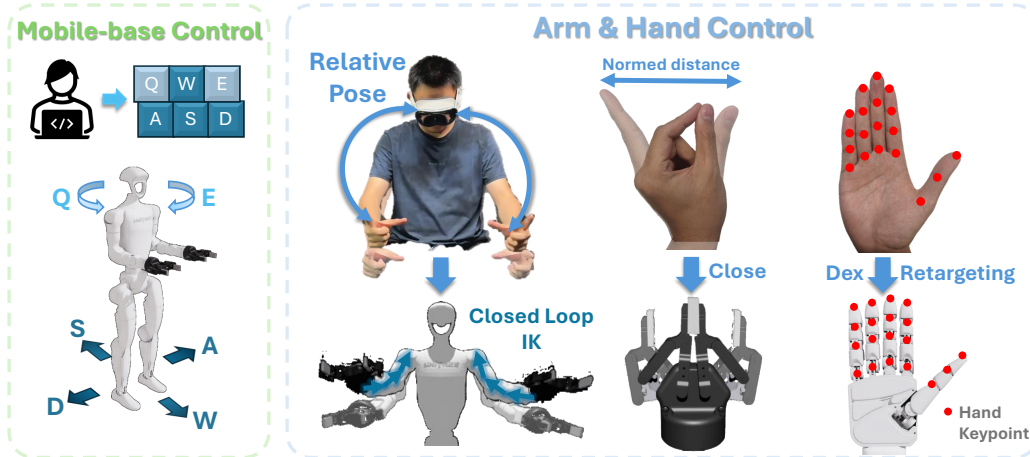### 3.2 Mobile-based Teleoperation for data collection

Figure 3: Data collection system of AgentWorld. For the mobile-base control, we allow the users to use the keyboard to to control robots, both wheel-based and legged. For arm & hand control, we use the VR head set to get the hand pose and compute IK for obtaining the arm action, and utilize re-targeting methods to drive robotic hands.

Teleoperation serves as an effective method for collecting robotic manipulation data, though it presents particular challenges for long-horizon tasks in expansive household environments. These challenges primarily stem from the need for coordinated mobile navigation between multiple locations. Our teleoperation system addresses these difficulties through the integrated pipeline shown in Fig. 3, which comprises two key components: **Mobile-base Control** and **Arm & Hand Control**.

Mobile-base Control. In order to realize mobile manipulation tasks data collection, we aim to make the mobile-based robot move around in the room. Basically, for every time step t, we tend to control the robot with an action $a_t = (v_x, v_y, v_\theta) \in \mathcal{A}$ with $\mathcal{A} = \mathbb{R}^3$ a 3 DOFs action space , where v_x and v_y is the robot velocity of the movement on the plane, and v_θ is the velocity of the yaw angle of the robot floating base.

For wheeled or direct controllable floating base robots, movement control is straightforward. For legged humanoids, we employ IsaacLab's reward-driven locomotion policy [27], adapted for dual-arm stability. The model enables robot movement control via keyboard as demonstrated in Fig. 3. We can switch the robot between locomotion and manipulation by fixing or freeing its base link. The training details will be discussed in the supplementary material.

Arm & Hand Control. We implement the OpenTeleVision system [28] in IsaacLab to control robotic arms with grippers or dexterous hands. The VR headset captures hand keypoints in its coordinate frame. With the headset camera fixed on the head of a humanoid robot (or a certain reasonable location for other embodiments), we calibrate the relative pose between hand wrists and headset in real world to match the camera-end effector relationship in simulation. Arm joint poses are computed using a Closed-loop Inverse Kinematics (CLIK) algorithm based on Pinocchio [29].

For different types of end effectors, as shown in Fig. 3, the gripper action is controlled by normalizing the thumb-to-index fingertip distance, and we employ dex-retargeting [30] to convert hand keypoints to joint actions of the dexterous hands.

4 AgentWorld Dataset

Based on our AgentWorld simulation platform, we construct a comprehensive dataset containing diverse household activities performed by different embodiments. This section details our task settings across different scene layouts and the collected dataset specifications.

4.1 Scene & Task Description

We focus on daily household activities involving both bimanual and single-arm manipulation tasks. Our environment features three primary room layouts: **Living room**, **Bedroom**, and **Kitchen**. For each room type, we generate 10 distinct base layouts using our scene generation module. After establishing the basic structure, we randomly populate each room with inter-actable 3D assets with reasonable placements, ultimately creating 150 unique room configurations. To cover most of the household activities, we separate our task settings in two kinds: **Basic Tasks** and **Multistage Tasks**.

Basic Tasks. We collect sequences of data on basic manipulation tasks in order to enable robots to learn some basic human operations, including but not limited to:

- **Pick & Place**: Simple pick & place tasks are basic ability for robots to master. In our dataset, we control the gripper jaw or dexterous hand of the robot to pick cups, fruits, plates and toys, etc, and place the objects in boxes, sinks, trash bins or racks.
- **Open & Close**: We teleoperate the robot to conduct articulated manipulation tasks, including opening doors of refrigerators, cabinets, and the ovens, followed by closing actions to bring articulation knowledge to manipulation models.
- **Push & Pull**: The dataset includes manipulation tasks where the robot interacts with drawers (pulling them open or pushing them shut), slides objects (such as boxes or books) across tables, and operates buttons or switches, to teach precise position controls to the robot.

Multistage Tasks. After introducing basic tasks of the robots, we believe it is crucial for the robots to learn how to perform household activities that are more meaningful. Therefore, we design and record multistage tasks in three typical home environments:

- **Living Room**: Tasks include *Organizing books* (put the books on shelves), *serving drinks* (pick up the pitcher and the cup and pour the drink), and *cleaning tables* (put trash in trash cans).

- **Bedroom**: We simulate activities like *making a bed* (Put the objects on the bed away, placing pillows), *organizing a wardrobe* (open/close closets, hang the clothes rack), and *setting an alarm clock* (pressing buttons or sliding switches).

- **Kitchen**: Complex interactions such as *Store the food* (pick up food on the table and store it into the fridge), *heat up the food* (pick up food and put it into the microwave, and then press the button), and *Clean the dishes* (use a sponge to clean the dishes and place them into a rack).

These tasks require sequential decision-making and combine primitive skills (pick & place, Open & Close and push & pull) to achieve higher-level goals, mimicking real-world household demands.

4.2 Dataset Details

We employ four different embodiments: Unitree G1 [31], H1 [32], Franka Emika panda with a wheel base, and a DOBOT X-Trainer [33] with a fixed platform. The end effectors of the two humanoid robots are set to be either the 2F-85 Gripper [34] or the TRX-Hand5 [35]. For the basic tasks, we have provided 10 different assets of every task category, and recorded 10 sequences of each robot manipulating a certain asset with different poses, with the length of the sequences from 80 to 150 steps. As for the multistage tasks, we recorded 30 to 50 sequences for every task, with the length from 300 to 800 steps. The control frequency is fixed to 30 fps. In our settings, a **Wrist Camera** is attached on each end effector on the robot arms, and a **Head Camera** is set immobile on the humanoid robots. For the X-Trainer, two **Wrist Cameras** and a fixed **Third-person Camera** are set to align with the real-world setting. For each frame, we record the current robot joint poses, the action to take, and RGB-D output of the cameras. More dataset details will be discussed in the supplementary material.

5 Experiments

5.1 Imitation Learning for AgentWorld Dataset

To comprehensively evaluate our AgentWorld dataset, we benchmark multiple imitation learning paradigms: (1) conventional *Behavior Cloning (BC)* as our baseline; (2) *Action Chunking Transformers (ACT)* [11] for sequence-aware action prediction; (3) *Diffusion Policies (DP)* [12] for stochastic action modeling; and (4) π_0 [13] that integrates language instructions with visual inputs. All models process synchronized multi-view RGB images, with π_0 additionally using language task descriptions. The unified output space comprises: (1) dual-arm joint and end effector actions, (2) 3-DOF velocity commands for mobile base navigation, and (3) a binary locomotion-manipulation mode selector that determines whether to prioritize mobile navigation or stationary manipulation. The velocity commands will be forwarded to the mobile-base controller in our data collection system, either a wheel-base driver or a pretrained legged locomotion policy.

For Basic Tasks, we train per-category models and evaluate with unseen assets/surroundings. For Multistage Tasks, we train on single tasks and evaluate with varied asset poses and materials. We evaluate 50 episodes of each task with every model, and use success rate to assess the models' performances. The average success rates of every task category are demonstrated in Tab. 2. For basic manipulation tasks, ACT [11] achieves the most consistent performance (62-76% success rate) due to its action chunking mechanism effectively handling short-horizon action sequences. In multistage tasks, π_0 [13] shows superior results (20-30% v.s. others' 4-16%), benefiting from its pretrained representation that captures long-horizon task structures. However, all methods struggle with complex mobile manipulation, where both precise control and long-term planning are simultaneously re-

Task	Basic Tasks			Multistage Tasks		
Algo.	Pick & Place	Open & Close	Push & Pull	Living Room	Kitchen	Bedroom
BC	52%	62%	58%	10%	4%	6%
ACT[11]	**66%**	**84%**	72%	16%	12%	14%
DP[12]	64%	78%	**76%**	28%	**20%**	**24%**
π_0[13]	64%	82%	70%	**30%**	**20%**	18%

Table 2: Quantitative results for different imitation learning algorithms in AgentWorld Dataset.

quired. Several demonstrations for the inference results of imitation learning algorithms are shown in Fig. 4. The training details will be elaborated in our supplementary material.
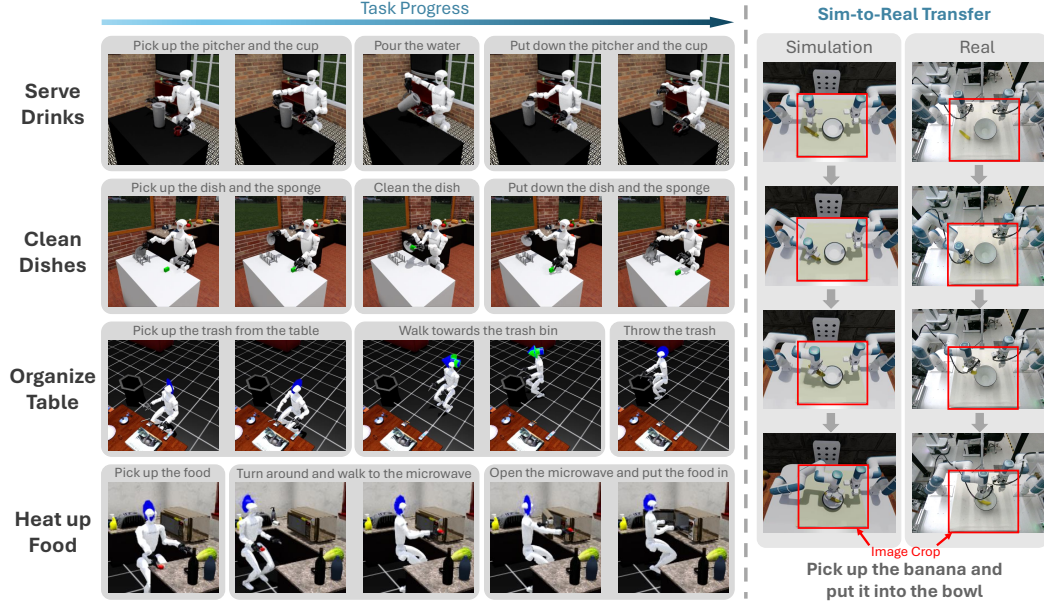

Figure 4: Qualitative results for different imitation learning algorithms in AgentWorld Dataset, and a Sim-to-real transfer example to validate the availability and generalizability of our data.

5.2 Real-world Experiment

To validate the sim-to-real transfer capability of our platform, we conducted real-world basic task experiments involving placing various objects into a bowl. We trained the π_0 policy using a sim-real-combined paradigm: initial training on simulation data (10 assets with 10 trajectories each) followed by fine-tuning with limited real-world demonstrations (3 objects - banana, apple, mug - with 3 trajectories each with varied object positions). This combined approach achieved a **29.3%** success rate, and as illustrated in Fig. 4, the policy successfully handles basic manipulation tasks despite the reality gap. These results suggest that AgentWorld's realistic material rendering and physical simulation properties provide an effective foundation for sim-to-real transfer, while highlighting the importance of targeted real-world fine-tuning for practical deployment.

6 Conclusion

We present AgentWorld, a versatile simulation platform that bridges the gap between procedural scene construction and mobile robotic data collection for embodied AI research, which addresses key challenges in robotic learning, demonstrated through successful imitation learning experiments and sim-to-real validation. Our experiments with the AgentWorld Dataset show that the platform's flexible scene configuration and comprehensive teleoperation system enable effective training of manipulation policies. Future work will expand the platform's capabilities to include more complex physical interactions and multi-agent scenarios, further enhancing its utility for robotics research.

7   Limitations

Despite the comprehensive capabilities of AgentWorld, we identify two key technical limitations in the current implementation:

- **Limited Deformable Object Support**: Due to constraints in the underlying Isaac Sim physics engine, our platform currently has limited support for simulating deformable objects (e.g., cloth, ropes, or soft bodies). This restricts the range of manipulation tasks that can be accurately modeled and transferred to real-world scenarios involving such objects.

- **Pure Sim-to-Real Realization**: The significant domain gap between simulation and reality requires extensive manual tuning of rendering parameters to achieve visual alignment. While we have demonstrated successful sim-to-real transfer for simple tasks through simulation pretraining followed by few-shot real-world fine-tuning, it's still difficult for real-world deployment with policies trained on pure simulation data. Previous work[36] has shown an improvement in real-world performance when combined synthetic data with real data, and Isaac Sim's validated physics and rendering assure the effectiveness of our platform. Besides, multistage manipulation tasks still demand substantially more real-world demonstrations to bridge the reality gap effectively. Future works include verifying the effectiveness of the introduction of the simulation data, realizing sim-to-real transfer for long-horizon tasks based on pure synthetic data, etc.

These limitations point to important future research directions, including integration of more advanced physics engines and development of robust domain adaptation techniques for challenging manipulation scenarios.

References

[1] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[2] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.

[3] X. Li, D. Guo, H. Liu, and F. Sun. Embodied semantic scene graph generation. In *Conference on robot learning*, pages 1585–1594. PMLR, 2022.

[4] S. Chen, R. Ye, S. Dingwani, P. Fazli, H. Seifi, and T. Bhattacharjee. Rcaregen: An interface for scene and task generation in rcareworld. In *Proceedings of the 2025 ACM/IEEE International Conference on Human-Robot Interaction*, pages 1274–1278, 2025.

[5] P. Ren, M. Li, Z. Luo, X. Song, Z. Chen, W. Liufu, Y. Yang, H. Zheng, R. Xu, Z. Huang, et al. Infiniteworld: A unified scalable simulation framework for general visual-language robot interaction. *arXiv preprint arXiv:2412.05789*, 2024.

[6] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

[7] N. Chernyadev, N. Backshall, X. Ma, Y. Lu, Y. Seo, and S. James. Bigym: A demo-driven mobile bi-manual manipulation benchmark. In *8th Annual Conference on Robot Learning*.

[8] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

[9] NVIDIA. Isaac sim 4.5 - robotics simulation and synthetic data generation. https://developer.nvidia.com/isaacsim, 2025.

[10] U. Engine. Unreal engine. *Retrieved from Unreal Engine: https://www. unrealengine. com/en-US/what-is-unreal-engine-4*, 2018.

[11] T. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *Robotics: Science and Systems XIX*, 2023.

[12] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

[13] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[14] H. Fu, W. Xu, R. Ye, H. Xue, Z. Yu, T. Tang, Y. Li, W. Du, J. Zhang, and C. Lu. Demonstrating rfuniverse: A multiphysics simulation platform for embodied ai. In *Robotics: Science and Systems*, 2023.

[15] R. Ye, W. Xu, H. Fu, R. K. Jenamani, V. Nguyen, C. Lu, K. Dimitropoulou, and T. Bhattacharjee. Rcare world: A human-centric simulation world for caregiving robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 33–40. IEEE, 2022.

[16] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su. Maniskill: Learning-from-demonstrations benchmark for generalizable manipulation skills. *CoRR, abs/2107.14483, 2021b. URL https://arxiv. org/abs/2107*, 14483, 2021.

[17] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.

[18] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[19] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023.

[20] H. Wang, J. Chen, W. Huang, Q. Ben, T. Wang, B. Mi, T. Huang, S. Zhao, Y. Chen, S. Yang, et al. Grutopia: Dream general robots in a city at scale. *arXiv preprint arXiv:2407.10943*, 2024.

[21] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

[22] A. Avetisyan, T. Khanova, C. Choy, D. Dash, A. Dai, and M. Nießner. Scenecad: Predicting object alignments and layouts in rgb-d scans. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 596–612. Springer, 2020.

[23] Z. Chen, J. Tang, Y. Dong, Z. Cao, F. Hong, Y. Lan, T. Wang, H. Xie, T. Wu, S. Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. *arXiv preprint arXiv:2409.12957*, 2024.

[24] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

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

[26] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

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

[28] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang. Open-television: Teleoperation with immersive active visual feedback. In *8th Annual Conference on Robot Learning*.

[29] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard. The pinocchio c++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *2019 IEEE/SICE International Symposium on System Integration (SII)*, pages 614–619. IEEE, 2019.

[30] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Robotics: Science and Systems*, 2023.

[31] Unitree Robotics. G1. www.unitree.com/g1, 2024.

[32] Unitree Robotics. H1. www.unitree.com/h1, 2024.

[33] Dobot. Dobot x-trainer. https://www.dobot-robots.com/products/education/x-trainer.html.

[34] Robotiq. 2f-85 adaptive robot gripper. https://robotiq.com/products/2f85-140-adaptive-robot-gripper.

[35] S. Yang, W. W. Lee, Z. Zhang, Y. Xiong, J. Liang, P. Lu, Y. Zhu, T. Liu, J. Li, R. Wang, et al. Trx-hand5: An anthropomorphic hand with integrated tactile feedback for grasping and manipulation in human environments. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5289–5296. IEEE, 2024.

[36] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev, S. Reed, K. Goldberg, A. Mandlekar, L. Fan, and Y. Zhu. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, Los Angeles, CA, USA, 2025.

[37] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[38] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[39] R. Cadene, S. Alibert, A. Soare, Q. Gallouedec, A. Zouitine, and T. Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

A Supplementary Video

The supplementary video presented in our website provides dynamic visualizations of AgentWorld's core capabilities, organized as follows:

- **Scene Construction Demonstration (0:00-0:45)**:

    - **0:00-0:13**: Layout generation showcasing different room organization with stairs.
    - **0:13-0:20**: Semantic asset selection and placement demonstration.
    - **0:20-0:29**: Visual material configuration with walls, floors and assets with various materials.
    - **0:29-0:45**: Interactive physics simulation demonstrating inter-actable assets with physics properties like collisions and joints, etc.

- **Teleoperation System (0:45-2:04)**:

    - **0:45-1:05**: An example of VR-based arm and gripper teleoperation.
    - **1:05-2:04**: An example of dexterous hand control with manipulation of objects and articulated assets.

- **Learning Results (2:04-2:56)**:

    - **2:04-2:42**: Imitation learning performance in simulation for multistage tasks.
    - **2:42-2:56**: A Sim-to-real transfer demonstration of simple pick & place task.

The video complements our paper by providing real-time demonstrations of AgentWorld's interactive features and experimental outcomes. Each segment highlights key functionalities of our simulation platform.

B Details on Procedural Scene Generation

Our procedural scene generation framework is **fully automatic** for scenes constructed from existing assets, requiring no manual intervention for object placement or physics configuration. For new assets, only **minimal one-time human input** is needed during initialization: assigning categorical labels, room types, and optional co-occurrence constraints (following ProcTHOR[2] design principles). Subsequent bounding box calculations and physics property assignments (e.g., mass, friction) are handled automatically. With this approach, unlimited scenes can be generated without additional human effort once assets are annotated.

The system extends ProcTHOR's rule-based algorithms by integrating Unreal Engine's **Procedural Content Generation (PCG)** framework, enabling real-time scene generation (2–3 seconds per scene) with visual feedback. Key technical improvements include:

- **Geometric realism**: Wall thickness simulation and optimized room layouts using metrics like aspect ratio, floor-to-bounding-box area ratio, and L-shape expansion to produce natural, rectangular spaces.
- **Layout pragmatism**: Enhanced inter-room connectivity checks to avoid impractical designs (e.g., isolated rooms) and support for multi-floor structures.
- **Physics automation**: Rule-based material properties (e.g., "wood" $\rightarrow$ predefined mass/friction) triggered by semantic asset tags.

While scene generation and physics are fully automatic after asset setup, the initial **semantic tagging of new assets** (e.g., labeling a chair as "furniture") remains the sole manual step. These enhancements collectively advance environmental realism beyond ProcTHOR's original implementation while maintaining computational efficiency.

C Training Details

C.1 Locomotion Policy for Humanoid Robots

As shown in Table C.1, we define a series of rewards to encourage humanoid robots to achieve walking with different terrain adaptation such as flat planes and stairs of varying steepness, etc. Meanwhile, we define relevant termination conditions to end a certain current episode, as shown in Table C.1. In addition, we use the foot trajectory generator[37] to generate a reference trajectory for robot feet. The output of the locomotion policy network will be added as an increment to this reference trajectory. In this way, we can narrow the exploration range of the action and accelerate the training process. During the training process, we randomly set different joystick control commands within a certain period, enabling the robot to encounter as many different control commands as possible. The goal of the training is to make the robot follow these commands to train the robot's locomotion policy.

Reward	Description
track_angle_vel_z_exp	Encourages the robot to follow the rotation commands in the yaw direction.
track_linear_vel_xy_exp	Encourages the robot to follow the translation commands in the x and y directions.
linear_vel_z_l2	Penalizes the robot for fluctuations in the z - direction.
feet_air_time	Encourages the robot to lift its feet rather than drag them on the ground.
feet_slide	Penalizes the robot for foot sliding.
dof_acc_l2	Penalizes the robot's joint accelerations to ensure smooth joint movements.
dof_torques_l2	Penalizes the robot's joint torques to keep the joint torques as stable as possible.
dof_pos_limits	Penalizes the robot when its joints exceed the defined limits.
joint_deviation_arms	Penalizes the deviation of the robot's arm joints from their default positions, guiding the policy to explore around the default positions.
joint_deviation_hip	Penalizes the deviation of the robot's hip joints from their default positions, guiding the policy to explore around the default positions.
joint_deviation_torso	Penalizes the deviation of the robot's torso joints from their default positions, guiding the policy to explore around the default positions.

Table 3: Rewards

In practical applications, the robot's upper limbs will grasp different objects, which means that the positions of the upper limbs can vary widely. Therefore, to ensure that our locomotion policy is robust enough, we have incorporated randomization of the arm positions during the training process. Based on the given default arm position, for every 150 frames, we superimpose the result of a uniform sampling as the arm target joint position. This approach enables the training to cover a wide range of different positions of the upper limbs.

C.2 Implementation of Imitation Learning Algorithms

Observation & Action Space. The input RGB observations maintain a consistent resolution of 480×640. The robot's proprioceptive state is defined as $s = [s_{qpos}, s_e, s_f]$, where s_{qpos} represents joint positions of the arms, s_e denotes end effector poses, and $s_f = (x, y, \theta)$ is the float-

Condition	Description
base_contact	If the robot's body collides with the ground, it indicates that the robot has fallen.
time_out	Limits the maximum time for each training episode.

Table 4: Termination Conditions

ing base's plane location and yaw angle. As previously described, the algorithm outputs the action $a = [a_{qpos}, a_e, a_f, 0/1]$, where a_{qpos} and a_e specify arm joint and end effector actions, $a_f = (v_x, v_y, v_\theta)$ represents mobile base velocity commands, and $0/1$ the binary value selects between locomotion and manipulation modes. The Unitree G1, H1, and Franka Emika Panda arm each possess 7 degrees of freedom (DOFs), while the dual-arm X-Trainer has 6 DOFs per arm. For end effectors, grippers are controlled via a single normalized value for opening distance, and the TRX-Hand5 features 18 actuated DOFs.

Algorithm Architecture. Our behavior cloning algorithm utilizes a ResNet-18 [38] image encoder for visual feature extraction. The flattened image features are concatenated with proprioception states and processed through a 5-layer MLP for action prediction. Layer normalization and tanh activation are applied to the output. For ACT, Diffusion Policy, and π_0, we adopt the LeRobot [39] implementation with action chunk sizes of 20 for basic tasks and 40 for multistage tasks. All approaches incorporate *temporal ensemble* to enhance action smoothness.

Implementation Details. All algorithms are trained with batch size 16, except π_0 which uses 8. Training spans 50K steps for basic tasks and 150K for multistage tasks. We observe convergence across all methods with 80K-100K steps, with no subsequent performance increases. On NVIDIA L20 GPU, training durations are 6 hours for behavior cloning, 12 hours for ACT and Diffusion Policy, and 18 hours for π_0.

D   Dataset Details

Table. D demonstrates detailed information of our datasets including concrete descriptions and number of trajectories of each task in every category. Our dataset currently contains 1,150 trajectories (">1000" indicates active expansion). We're actively expanding the collection to improve generalization and sim-to-real transfer, and our platform supports community contributions to scale diversity.

Category	Task Description	Num of Trajectories
Pick & Place	Pick some objects and place them in the bowl.	10×10
	Pick some objects and place them into the opened microwave / fridge / stove.	30×10
	Pick some objects out from the bowl.	10×10
	Pick some objects out from the opened microwave / fridge / stove.	30×10
Open & Close	Open & Close doors of rooms.	10×20
	Open & Close doors of the furniture: fridges, microwaves, stoves.	30×20
Push & Pull	Pull out & Push in the drawers.	10×20
	Push the buttons on microwaves / stoves.	10×20
	Push some objects on the tables away from the robot.	10×10
Living Room	Pick up the books on the table and walk to the shelf and organize them.	10×50
	Pick up the pitcher and the cup and pour the drink.	5×30
	Pick up the trash on the table and walk to the trash bin to throw them away	10×50
Bedroom	Slide the objects on the bed away and put the pillow on the top of the bed	10×50
	Pick up the clothes rack and open the closets and hang them in.	10×50
	Pick up the alarm clock and push the button.	5×30
Kitchen	Pick up dishes on the table and store it into the fridge.	5×30
	Pick up the food and turn around to open the microwave and put the food in.	5×50
	Pick up the dishes on the rack and the sponge and clean the dishes.	5×50

Table 5: Dataset details for task descriptions and number of trajectories. The number of trajectories is demonstrated as number of assets $\times$ number of recorded sequences.