# ManiLadder: Benchmarking Manipulation Intelligence Frontier via a Categorized and Multi-Level Task Ladder


Fig. 1: The ManiLadder benchmark. It comprises 4 difficulty levels, each with 8 task-category units combining diverse object types (rigid, articulated, deformable), end-effectors (grippers, dexterous hands), and embodiments (single-, dual-, and mobile-arm). Each unit contains 3∼4 tasks, totaling 114 simulated manipulation tasks.

***Abstract*— We introduce ManiLadder, a large-scale simulation benchmark designed to quantitatively assess progress in robotic manipulation intelligence and capacity. It turns *How difficult a task can my algorithm solve?* into a measurable ladder to climb. ManiLadder consists of 114 simulation tasks spanning four difficulty levels, covering diverse object types (rigid, articulated, and deformable) and robot embodiments (single-arm, dual-arm, grippers, and dexterous hands). Each task is paired with 50 high-quality human demonstrations. To construct ManiLadder, we propose a Metric-Anchored Iterative Task Ladder DEsign (MILE) pipeline: tasks are tuned until their objective composite scores fall into predefined difficulty intervals, as measured by 2D- and 3D-based imitation learning policies. Our experiments show that commonly used imitation learning algorithms achieve performance corresponding roughly to Level 2, revealing a significant gap to higher-level manipulation competence and setting clear targets for future research. We further provide preliminary results on vision-language-action (VLA) models and transfer learning. Additional experiments and videos are available on our website.**

## I. Introduction

Recent years have witnessed remarkable advances in the field of robotic manipulation, encompassing more affordable and user-friendly hardware [1, 2], powerful foundation models [3, 4], effective training algorithms [5, 6], as well as large-scale datasets [7, 8], strengthening the belief that generalizable and broadly capable manipulation agents are increasingly within our reach. These advances have, in turn, amplified the need for a new benchmark capable of fairly quantifying progress in robotic manipulation intelligence and **evaluating how far we are from a robot with AGI-level manipulation capacity**. However, current benchmarks in robotic manipulation lag considerably behind the huge progress in hardware, algorithms, models, and datasets. Most of benchmarks focus on evaluating specific dimensions of manipulation tasks and policies such as task horizon [9], generalization ability [10], sim2real transfer ability [11], and continual learning [8], which makes researchers frequently choose different benchmarks or even self-designed tasks when evaluating their novel algorithms and models [12, 13]. This makes it challenging to precisely evaluate the level of task difficulty a model can successfully handle, and thus to determine whether it represents a genuine step toward general-purpose intelligent robots relative to prior work.

To build such a benchmark for systematically evaluating the progress of manipulation capacity, one approach is to assemble a suite of simulation-based manipulation tasks organized by difficulty. Intuitively, there are two ways to

Manuscript 2486 submitted to 2026 IEEE International Conference on Robotics and Automation (ICRA). Received September 15, 2025.

| Benchmark | Simulator | No. of tasks | Human demonstrations | Task Diversity | W/ deformable objects | Real-world reproducibility |
|---|---|---|---|---|---|---|
| MetaWorld [14] | Mujoco [15] | 50 | ✗ | Objects | ✗ | ✗ |
| FactoredWorld [16] | Mujoco [15] | 19 | ✗ | Different goal shapes of cubes | ✗ | ✓ |
| COLOSSEUM [10] | V-Rep [17] | 20 | ✗ | Objects, camera, and visual perturbations | ✗ | ✓ |
| LIBERO [8] | Mujoco [15] | 120 | ✓ | Objects | ✗ | ✗ |
| ManiLadder (ours) | Genesis [18] | 112 | ✓ | Objects, robots, difficulty levels | ✓ | ✓ |

TABLE I: Comparison between ManiLadder and other manipulation benchmarks focusing on task taxonomy and difficulty levels. ManiLadder is a large-scale benchmark and is the only one that involves different robots, difficulty levels, and deformable objects. Building upon Genesis [18], we also enabled large-scale GPU-accelerated parallel simulation. Similar to previous works, we also reproduce part of the tasks in the real-world.

divide task difficulty levels when constructing this task suite: 1) *Direct Method*: directly defining task difficulty based on the intrinsic properties of the task itself, such as the objects involved and the types of skills required; 2) *Indirect Method*: indirectly defining task difficulty by using objective metrics derived from policies executed on the task, such as success rate. For the first way, prior works have tried to model task complexity purely from a theoretical standpoint [19, 20], from intuition [21], from compositions and transferring subskills [22, 23], or manually defined predicate verbs [24, 25] and contact patterns [26–28]. However, the inherent complexity of manipulation tasks makes these methods overly simplistic, lacking practical algorithms for assigning difficulty levels to arbitrary tasks.

For the second way, when defining task difficulty through policy performance, two key challenges arise: (i) selecting a policy applicable across a broad spectrum of tasks, and (ii) specifying performance metrics that are both fair and comparable across diverse tasks. Thanks to recent advances in robot imitation learning algorithms [4, 5, 29] that can perform reasonably well on a wide range of tasks, the first issue is largely addressed. For the second question, while success rate is a popular metric in manipulation, it is coarse and one-dimensional, failing to capture finer-grained aspects of performance [30], so we need to devise composite evaluation metrics that are broadly applicable and better reflect nuanced differences across large-scale task suites.

In this work, we present ManiLadder, a comprehensive and systematic simulation benchmark for robotic manipulation that organizes tasks into well-defined categories and multiple difficulty levels. By framing task difficulty as a ladder to climb, ManiLadder provides a measurable path toward human-level manipulation competence and offers the community a clear policy-based metric for assessing and advancing algorithmic capability. It comprises four difficulty levels. Within each level, we construct standardized mini-suites spanning two object families (rigid and articulated, and deformable) and four robot embodiments (single-arm with gripper, single-arm with dexterous hand, dual-arm with grippers, and dual-arm with dexterous hands). Each mini-suite contains three to four tasks, yielding a total of 112 carefully designed manipulation tasks, as shown in Figure 1. ManiLadder covers a diverse spectrum of object categories, supports multi-view stereo observations, and enables large-scale parallel simulation based on the Genesis [18] simulator. Every task is paired with a well-specified reward function, 50 high-quality human demonstrations, a visual-based teleoperation interface, and reference algorithms for benchmarking. To stratify task difficulty, we adopt the indirect method and introduce Metric-Anchored Iterative Task Ladder DEsign (MILE) pipeline: tasks are tuned until their objective composite scores fall into predefined difficulty intervals, as measured by 2D- and 3D-based imitation learning policies (2D diffusion policy [5] and 3D diffusor actor [29]). The composite metric jointly accounts for task success rate and stage-wise progress, providing a more fine-grained and robust assessment of difficulty.

In our experiments, we observed that current common imitation learning algorithms typically perform at approximately Level 2 on ManiLadder, thereby motivating further research toward more capable robotic manipulation learning algorithms. We also perform initial experiments on ManiLadder for Vision-Language-Action models on our website. We envision ManiLadder as a stepping stone for measuring, tracking, and accelerating advances in robotic manipulation intelligence, while also fostering progress in broader areas such as transfer learning and continual learning.

## II. Related Works

### A. Robot Manipulation Task Taxonomy

Classification and taxonomy for a research problem can provide a structured framework that enables targeted methodological approaches and systematic knowledge organization. However, for robot manipulation tasks, it is challenging to divide robotic manipulation tasks into concise and well-defined categories, due to the extreme complexity arising from the tasks themselves, the diversity of the objects involved, and the variety of end effectors employed [28]. Some works use grasp types and contact modes [28, 31] or task primitives and skills [23, 32, 33] for task taxonomy. Others provide taxonomies for specific tasks, such as deformable object manipulation tasks [34] or bi-manual manipulation tasks [35]. However, the criteria underlying these task classifications are overly detailed and restrictive, making it difficult to accommodate complex, multi-stage tasks. In this work, we categorize robot manipulation tasks using two axes: the horizontal axis represents different tasks of the same difficulty level, distinguished by the type of manipulated object and the end effector employed; the vertical axis represents tasks of varying difficulty levels designed and measured by our MILE method, in which the object type and end effector remain the same.

Manuscript 2486 submitted to 2026 IEEE International Conference on Robotics and Automation (ICRA). Received September 15, 2025.

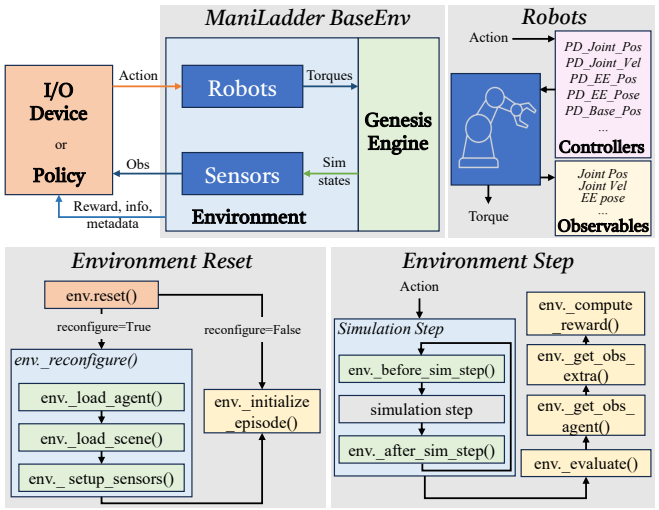


Fig. 2: The system design of ManiLadder. Top: the control and interaction logic and robots diagram. We implement modular classes, including Environment, Robot, Controllers, and Sensor, to standardize user workflows. Bottom: the `reset()` and `step()` logic in ManiLadder.

### B. Robot Manipulation Benchmarks and Evaluations

There are numerous simulation manipulation benchmarks [8, 10, 14, 21, 36–39] over the years. These studies target different aspects of manipulation research, such as task horizon, generalization, task transfer, and task reasoning. Simulated benchmarks offer superior reproducibility and low-cost evaluation, but face sim-to-real gaps that fail to accurately reflect real-world policy performance. On the other hand, there are also many real-world benchmarks [30, 40–44] or in-person challenges [45, 46] for robot manipulation tasks. These works try to ensure reproducibility with manuals for environment setups or enabling remote access to a centrally hosted evaluation platform. However, reproducibility and high participation barriers remain the greatest challenges faced by them. Despite the sim2real gap, we argue that using the objective, policy-based indirect method to define task difficulty yields a benchmark design principle that can transfer from simulation to the real world: conclusions drawn from simulation-based difficulty-stratified benchmarks are expected to remain consistent with those from real-world counterparts, even if the specific tasks differ. In this work, we choose to build ManiLadder in simulation.

## III. The ManiLadder Benchmark

ManiLadder is a benchmark designed to evaluate robotic manipulation capabilities across different object and robot types (the horizontal axis) with varying difficulty levels (the vertical axis). In this section, we begin by presenting the ManiLadder system diagram and the setup of the base environment, followed by its horizontal axis, and the accompanying teleoperation system for data collection.

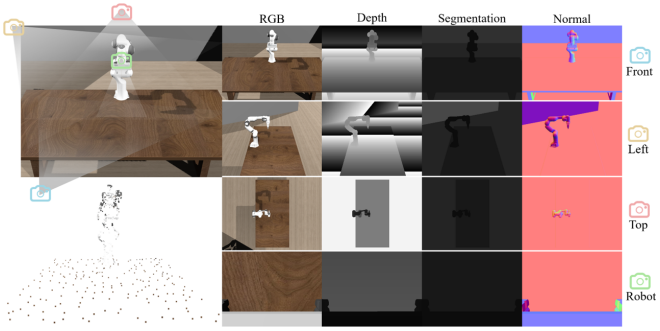


Fig. 3: The base environment and camera setups of ManiLadder. By default, we have 4 camera views for single-arm tasks (front, left, top, robot), and 5 camera views for dual-arm tasks. We process the fused point clouds to 1024 points, with only 200 points on the table. The camera's observation has RGB, Depth, Segmentation, and Normals simultaneously.

### A. Base Environment and System Design

ManiLadder is built upon Genesis [18], which is a cross-platform simulator that supports both rigid body and deformable objects, as well as GPU-based parallel simulation. To standardize the interaction between policy and environment, we establish a modular system upon the simulator as a set of APIs for users, as shown in Figure 2. These modular APIs provide useful high-level abstractions tailored for manipulation research, such as robot controllers, rewards, observation acquisition, and object location randomization. All these operations support batched actions, i.e., we use the same logic for single environment simulation and parallel simulation.

Then, we build a `ManiLadderBaseEnv` class, an extensible base environment that can be inherited by all other tasks, as shown in Figure 3. To mimic realistic indoor tabletop manipulation scenarios, we introduce a wooden-textured floor, a trapezoidal grey back wall, and a large table, with the table center aligned to the world origin for spatial consistency. We provide multiple default camera views for perception: front, left, and top views fixed in the scene, along with a wrist-mounted camera attached to the end-effector. All cameras support RGB, depth, segmentation, and point cloud modalities. To support 3D perception, we offer multi-view point cloud fusion for the scene, which merges multi-view point clouds, crops them to the agent's workspace, and downsamples to a fixed number of points. We also adjust the lighting to make the scene under better illumination.

### B. Task Organization and Taxonomy

To support broad applicability and reproducibility, tasks in ManiLadder are organized by two factors: object physical properties and robot embodiments. For the object factor, we follow the established distinction between *rigid and articulated objects* versus *deformable objects*. The former includes solid items with fixed geometry or articulated parts, such as blocks, mugs, doors, and drawers. The latter encompasses objects whose shape can change continuously under external


Fig. 4: The objects and robots gallery in ManiLadder.

force. To reflect the diversity of such materials, we further subdivide them into three geometric types: 1D (e.g., ropes, cables), 2D (e.g., cloths, towels), and 3D (e.g., sponges, plasticine, liquids), making tasks cover diverse object types.

For the robot embodiment factor, we divide tasks into four configurations: 1) single-arm with a parallel gripper, 2) single-arm with a dexterous hand, 3) dual arms with grippers, and 4) dual arms with dexterous hands. We also include mobile manipulators in some tasks, which is built upon the model in TidyBot++ [47]. Currently, we support Franka Panda and XArm as our robot arms, and LeapHand [2] as our dexterous hand. We will incorporate more robot embodiments in our later version.

For each combination of object category and embodiment type, we design 3 tasks for rigid and articulated objects and 3 tasks for deformable objects (one each for 1D, 2D, and 3D deformable objects). This results in a total of 112 manually designed tasks across four difficulty levels, forming a diverse and scalable testbed for manipulation intelligence. The object and robot gallery are shown in Figure 4.

### C. ManiLadder Objects and Task Design

We curated object assets from multiple open-source repositories to ensure both diversity and physical realism. For rigid body and articulated tasks, we primarily utilize the YCB Object and Model Set [41] for everyday household objects like packaged food (e.g., chips can, sugar box), fruits (e.g., banana, strawberry), and stationery items (e.g., small marker), and the PartNet-Mobility dataset [48] for articulated objects such as doors, drawers, and scissors. We refine the scales and perform convex decomposition to ensure these models can be positioned more appropriately in the environment. Other rigid bodies and articulated objects are created by ourselves.

For deformable objects, most assets are created in Blender and then processed using Trimesh to generate simulation-ready assets. For simulation, different deformable objects are under different simulation techniques. For 1D objects like ropes, we simulate flexible chains of capsule-shaped rigid bodies connected via hinge joints using MuJoCo's articulation system [15]. The number of segments controls rope resolution: higher segment counts yield more realistic deformations at the cost of higher computational costs. 2D deformables such as cloths are modeled using a position-based dynamics (PBD) solver, which balances numerical stability with real-time efficiency. For 3D deformables, we employed different solvers tailored to the physical properties of each material type. Clay-like objects were simulated using the Material Point Method (MPM) solver, which robustly handles large deformations and plastic flow while preserving material cohesion, enabling realistic modeling of shaping and compression tasks. Liquid objects were simulated using the Smoothed Particle Hydrodynamics (SPH) solver, which captures high-fidelity free-surface fluid motion and is particularly effective for tasks involving liquid–container interactions such as pouring, mixing, and splashing.

We design multi-stage dense reward functions to decompose each task into sequential phases. For instance, in a water-pouring task, rewards are structured to guide (1) reaching the beaker, (2) grasping the beaker and aligning it, and (3) completing the pour. In experiments, we also use the task stage-progress as an additional metric for deciding task difficulty levels.

### D. Dataset and Teleoperation System

We designed two teleoperation systems for collecting human demonstrations for tasks in ManiLadder, as shown in Figure 5. Both systems support single- and dual-arm robots as well as grippers and dexterous hands. In this work, we used the second system for data collection. We collect 50 human demonstrations for each task.

The first system uses monocular hand-pose detection with retargeting. A webcam placed in front of the operator streams images in real time. Then, we apply WiLoR [49], an end-to-end neural-network–based 3D hand reconstruction method, to recover 3D positions of hand keypoints. To improve teleoperation stability, we apply an exponential moving average for temporal smoothing of the keypoint trajectories. Next, we perform dexterous-hand retargeting [50] to map the human hand keypoints to a robotic hand (e.g., LeapHand [2]),

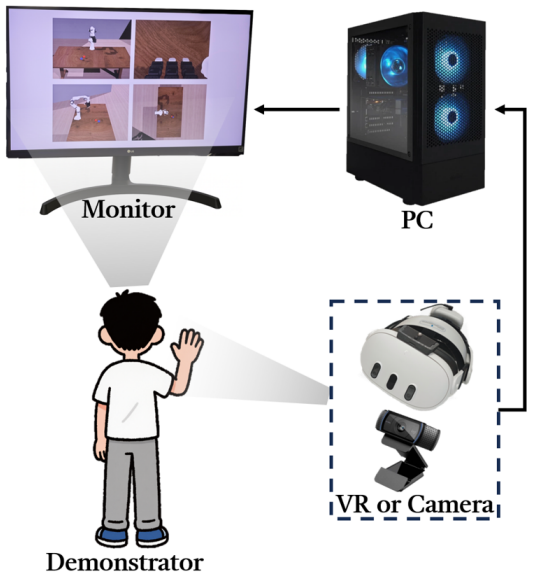


Fig. 5: The teleoperation system in ManiLadder. We use either VR- or Camera-based system.

yielding a 6-DoF end-effector pose together with finger joint angles. We subsequently align the human and robot coordinate systems so that motions of the real hand can control the simulated robot end effector. For grippers, the distance between the index finger and the thumb serves as the open/close signal. Because WiLoR natively distinguishes left and right hands, the system directly supports bimanual teleoperation. This pipeline is simple and cheap (it only requires a webcam of about $20) and does not require external network connectivity, making it accessible to a broad user base. Its main drawback is latency, dominated by NN-based monocular hand-pose and joint-angle estimation; on an RTX 4090, the system runs at approximately 10 Hz. The system is intended to facilitate future demonstration collection on ManiLadder by a wide range of researchers across the globe.

The second system uses a Meta Quest 3 VR pipeline for hand-pose estimation and retargeting. We use the Meta Quest 3's built-in hand-tracking algorithm to estimate the human hand 6-DoF pose in the headset (HMD) coordinate frame. An NGrok reverse proxy is launched on the local PC, and the 6-DoF hand pose is streamed to the PC in real time by enabling immersive mode on the headset and visiting our target webpage in the headset browser. We then apply the same retargeting procedure [50] to recover the robot arm's end-effector 6-DoF pose and the dexterous hand's joint angles. This setup offers lower latency and higher accuracy, at the cost of a higher price (Meta Quest 3 is roughly $700) and a requirement for external network connectivity. On an RTX 4090, it operates at about 25 Hz.

## IV. Metric-Anchored Iterative Task Ladder Design

In this section, we introduce the vertical axis design of ManiLadder: different task difficulty levels. We first introduce the Metric-Anchored Iterative Task Ladder DEsign (MILE) scheme, and then provide the theoretical analysis.

### A. The MILE Scheme

As discussed in Section I, we aim to train demonstration-based policies on each task and use objective metrics of the trained policies to stratify the corresponding task difficulty level. The workflow is naturally cyclic: begin with an initial task, gather data and train a policy; refine the task's difficulty according to the observed performance; and continue this collect–train–adjust loop until the metrics meet the desired thresholds, as shown in the left part of Figure 6. The key ingredients of this loop are the design of initial tasks, the policy algorithm, and the evaluation metrics. For the design of initial tasks, although the MILE method ultimately partitions task difficulty using objective criteria, the design of the initial task set still depends on human intuition and experience. In this work, we construct initial tasks of varying difficulty guided by two principles and along five dimensions.

**Two Principles**: 1) Avoid too complex object geometries. 2) Avoid tasks that can only be completed with unusual motion skills. These choices reflect our goal of assessing manipulation intelligence for commonplace, broadly useful activities, rather than abilities tied to particular objects or niche skills, for example, sleight-of-hand card shuffling or spinning a basketball on one finger.

**Five Dimensions**: 1) the task horizon (steps required to complete the task); 2) the spatial error tolerance; 3) the complexity of objects; 4) the degree of spatial–temporal coordination required between robots and between robots and objects; and 5) the level of physical and functional understanding needed (e.g., correct tool use). Figure 6 illustrates representative tasks designed under these guidelines.

### B. Imitation Algorithm Choice

We choose two imitation learning policies for MILE to evaluate the task: the 2D diffusion policy [5] and the 3D Diffuser Actor [29]. We didn't use other demonstration-based policies, such as offline RL methods, since they perform significantly worse than diffusion policies with visual inputs. We also did not adopt reinforcement learning approaches, as it is challenging to design reward functions that are fair and consistent across different tasks.

For the 2D Diffusion Policy, we generally follow the CNN- and UNet-based original Diffusion Policy [5] implementation, where we modify the inputs to 4 camera-view images. The resolution of images is $224 \times 224$. For the 3D Diffuser Actor, we employ the standard network structure. We also perform experiments with Vision-Language-Action Models (VLA) on ManiLadder. In this work, we choose $\pi_0$ [4], which is a 3B VLA model pretrained on large-scale manipulation datasets. More details are in the Appendix.

### C. Evaluation Metrics

MILE integrates both outcome-oriented and progression-oriented measures to capture task difficulty at multiple levels of granularity:

Manuscript 2486 submitted to 2026 IEEE International Conference on Robotics and Automation (ICRA). Received September 15, 2025.

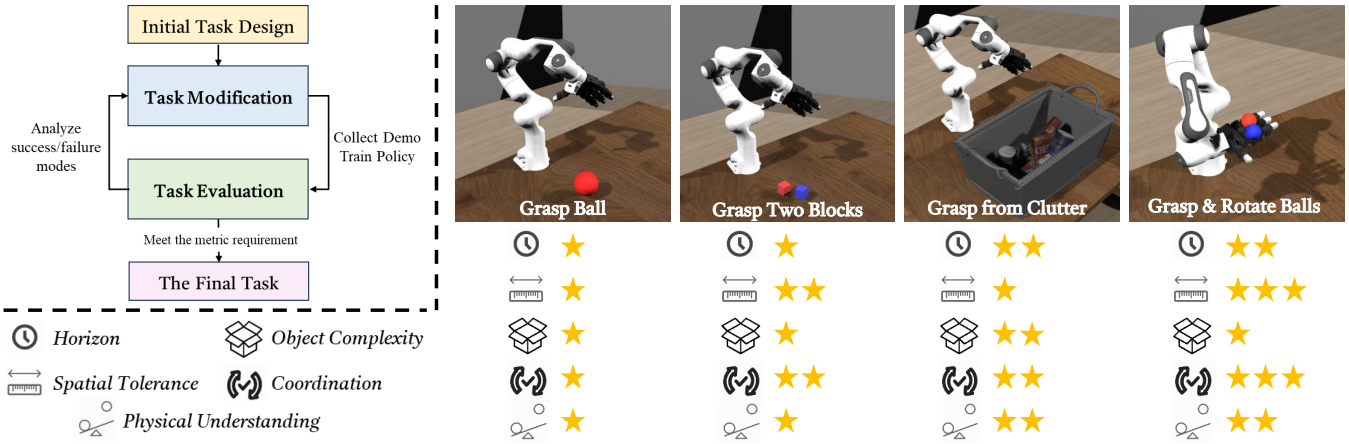


Fig. 6: Left: the MILE task design iteration. Right: a sequence of tasks of the same type (single-arm dexterous hand manipulating rigid bodies) at multiple difficulty levels. The two principles (simple objects and normal skills) and five design dimensions are shown in these tasks. Note, we try to keep the difficulty increment between successive levels roughly uniform, with each step increasing by approximately two "stars".

*a) Outcome Metric:* We use *task success rate* as the outcome metric:

$$\text{Succ}(T,\pi) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\pi \text{ completes task } T\}, \quad (1)$$

where $\pi$ denotes the policy, $T$ denotes the task, and $N$ the number of evaluation rollouts. In addition,

*b) Progression Metrics:* To capture fine-grained progress within a task, we define a set of binary *stage completion flags* for key sub-goals (e.g., grasp, lift, place) for each task, providing an interpretable measure of intermediate progress. These signals are normalized to the range $[0,1]$ and averaged across rollouts:

$$\text{Prog}(T,\pi) = \frac{1}{N}\sum_{i=1}^{N} \frac{\sum_{j=1}^{M} \pi \text{ completes } m_j}{M}, \quad (2)$$

where $M$ is the total sub-stages of the task and $m_j$ is the $j$-th sub-stage.

*c) Composite MILE Score:* Finally, we define the MILE score as a weighted combination:

$$\text{MILE}(T) = 0.5 \cdot \text{Succ}(T,\pi) + 0.5 \cdot \text{Prog}(T,\pi). \quad (3)$$

This score ensures that both final completion and intermediate progress are consistently reflected in the difficulty measure. In ManiLadder, we define four difficulty levels as follows: tasks with scores in $[0.5,1]$ are Level 1; $[0.2,0.5)$ are Level 2; $[0.1,0.2)$ are Level 3; and $[0,0.1)$ are Level 4.

### D. Theoretical Analysis

In this section, we formally prove that using the success rate of policies trained on tasks as the criterion for stratifying task difficulty is consistent with the definition of task difficulty defined by the computational complexity reduction theory [20]. Here, we present the necessary notations and lemmas, followed by the two theorems we prove. The complete proof process can be found in the appendix.

A task is defined as a partially observable Markov decision process (POMDP) with the tuple $\tau = (\mathcal{S}, \mathcal{A}, \mathcal{O}, p, \sigma, r, p_0)$which describes the state space, action space, observation space, dynamics sensor, reward function, and the initial state distribution. Let $\mathcal{T} = \{\tau_\epsilon\}$ be all tasks where $\tau_\epsilon$ denotes a specific task. Let $\pi : \mathcal{O} \to \mathcal{A}$ denote a policy and $\Pi_\epsilon$ denotes all policies for task $\tau_\epsilon$. Originally, a policy $\pi_\epsilon$ is *admissible* on task $\tau_\epsilon$ if the reward achieved by the policy $R_\epsilon(\pi_\epsilon) \geq R^*_\epsilon$, where $R^*_\epsilon$ is the success threshold of $\tau_\epsilon$. In this work, we extend this definition to accommodate using policy success rate as the criterion for task success:

**Definition 1** *A policy $\pi_\epsilon$ is $q_\delta$ admissible if its success rate on task $\tau_\epsilon$ is within the range of $[q-\delta, q+\delta)$, and we denote $\pi_\epsilon$ to be $\pi^*_\epsilon$ and all $q_\delta$ admissible policies to be $\Pi^*_\epsilon$.*

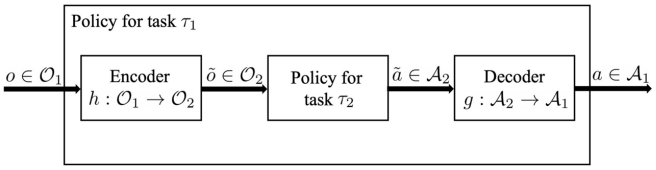


Fig. 7: The transformation between two tasks $\tau_1$ and $\tau_2$ using encoder $h$ and decoder $g$.

Now we can discuss how to compare the complexity between two tasks. The central idea is, task $\tau_1$ reduces to task $\tau_2$ if we can use any $q_\delta$-admissible policy for $\tau_2$ to solve $\tau_1$ with $q_\delta$ success rate, and according to the complicity theory, if task $\tau_1$ reduces to task $\tau_2$, then task $\tau_2$ is at least as complex as task $\tau_1$. To formalize this, we need to introduce *encoders* and *decoders*, as shown in Figure 7:

**Definition 2** *Let $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{A}_1$, $\mathcal{A}_2$ be the observation and action spaces of task $\tau_1$, $\tau_2$. Let $H_{1,2}$ denote a space of functions from $\mathcal{O}_1$ to $\mathcal{O}_2$, and let $G_{2,1}$ denote a space of functions from $\mathcal{A}_2$ to $\mathcal{A}_1$. We will refer to a function $h \in H_{1,2}$ as an encoder and a function $g \in G_{2,1}$ as a decoder.*

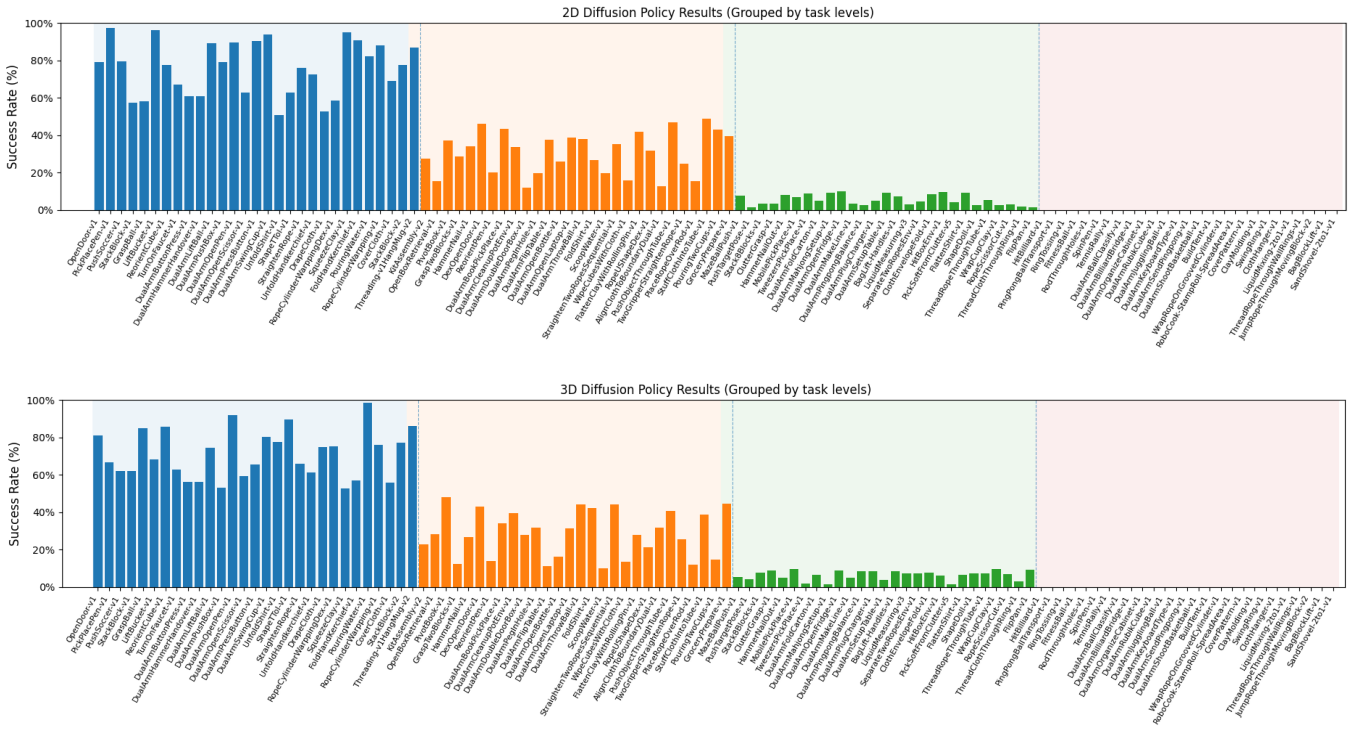


Fig. 8: All task success rates of two policies in ManiLadder. Results are calculated from 20 evaluation trajectories.

Then we have the following lemmas from [20]:

**Lemma 1** *Task $\tau_1$ reduces to task $\tau_2$ (written $\tau_1 \preceq \tau_2$) if for all $q_\delta$ admissible policies $\pi_2^*$, there exists an encoder $h \in H_{1,2}$ and a decoder $g \in G_{2,1}$ such that:*

$$g \circ \pi_2^\star \circ h \in \Pi_1^\star. \tag{4}$$

**Lemma 2** *Task $\tau_1$ and task $\tau_2$ are equivalent (written $\tau_1 \equiv \tau_2$) if $\tau_1 \preceq \tau_2$ and $\tau_2 \preceq \tau_1$. If $\tau_1 \equiv \tau_2$ holds with probability at least $\delta$, we say $\tau_1 \overset{\delta}{\equiv} \tau_2$.*

Then we can discuss the task difficulty levels in ManiLadder. Assuming all policies are trained with the same network structure, same training algorithms, same number of data, and same data quality (which is exactly the case in our work), we have the following two theorems:

**Theorem 1** *Let $\Pi_1$ and $\Pi_2$ be the policies trained in $\tau_1$ and $\tau_2$ respectively. If $\forall \pi_1 \in \Pi_1$, $\pi_1$ is $p_\delta^1$ admissible and $\forall \pi_2 \in \Pi_2$, $\pi_2$ is $p_\delta^2$ admissible, and $p^1 \in [p^2 - \delta, p^2 + \delta)$ and $p^2 \in [p^1 - \delta, p^1 + \delta)$, then $\tau_1 \overset{\delta/3}{\equiv} \tau_2$.*

**Theorem 2** *Let $\Pi_1$ and $\Pi_2$ be the policies trained in $\tau_1$ and $\tau_2$ respectively. If $\forall \pi_1 \in \Pi_1$, $\pi_1$ is $p_{\delta^1}^1$ admissible and $\forall \pi_2 \in \Pi_2$, $\pi_2$ is $p_{\delta^2}^2$ admissible, and $p^1 - \delta^1 \geq p^2 + \delta_2$, then $\tau_1 \preceq \tau_2$.*

With Theorem 1, we know that all tasks with the same range of success rates are in the same difficulty level, which are the tasks in ManiLadder of the same level. With Theorem 2, we know that tasks with higher success rate ranges can be reduced to tasks with lower success rate ranges, i.e., tasks with lower success rate ranges are in higher difficulty levels than tasks with higher success rate ranges.

## V. Experiments

In this section, we present the MILE success rates across all tasks in ManiLadder in Figure 8. We show the progress metric in the supplementary materials. We can see that current diffusion policies are generally lie in the level 2 tasks (more than 20% success rates with 50 demonstrations).

For more experiments like VLA and real-world transfer ability experiments, please check our website.

## VI. Conclusion and Limitations

In this work, we present ManiLadder, a simulation-based robot manipulation benchmark with a categorized and different difficulty-level task ladder for benchmarking the manipulation capacity of current learning-based policies. ManiLadder encompasses 114 diverse manipulation tasks across different object types, robots, and difficulty levels. We use MILE to assign the difficult levels for each task with objective metrics of policies trained on them. Our results show that current mainstream imitation learning algorithms generally stay at the level 2 tasks, which shows huge improvement space for future algorithms. We envision ManiLadder as a robust experimental platform that will support and catalyze future advances in robotic manipulation research.

Our work has several limitations: 1) We did not design benchmark tasks explicitly targeting task generalization. Although generalization is a crucial property of generally intelligent robots, constructing a single benchmark that comprehensively evaluates all aspects of robotic policy performance remains a formidable challenge. This work focuses primarily on assessing an algorithm's ability to solve individual

tasks; 2) Our domain randomization is currently insufficient for robust sim-to-real transfer. While object positions are randomized, camera poses, object physical parameters, and visual attributes such as color receive limited randomization.

## REFERENCES


[1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[2] K. Shaw, A. Agarwal, and D. Pathak, "Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning," *arXiv preprint arXiv:2309.06440*, 2023.

[3] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[4] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "pi0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[5] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *IJRR*, 2023.

[6] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao, "D (r, o) grasp: A unified representation for cross-embodiment dexterous grasping," *arXiv preprint arXiv:2410.01702*, 2024.

[7] A. O'Neill, A. Rehman *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," in *ICRA*, 2024.

[8] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *NeurIPS*, 2023.

[9] S. Zhang, Z. Xu, P. Liu *et al.*, "Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks," *arXiv preprint arXiv:2412.18194*, 2024.

[10] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox, "The colosseum: A benchmark for evaluating generalization for robotic manipulation," *arXiv preprint arXiv:2402.08191*, 2024.

[11] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees *et al.*, "Evaluating real-world robot manipulation policies in simulation," *arXiv preprint arXiv:2405.05941*, 2024.

[12] J. Barreiros, A. Beaulieu, A. Bhat, R. Cory, E. Cousineau, H. Dai, C.-H. Fang, K. Hashimoto, M. Z. Irshad, M. Itkina *et al.*, "A careful examination of large behavior models for multitask dexterous manipulation," *arXiv preprint arXiv:2507.05331*, 2025.

[13] C. Gao, H. Zhang, Z. Xu, Z. Cai, and L. Shao, "Flip: Flow-centric generative planning as general-purpose manipulation world model," *arXiv preprint arXiv:2412.08261*, 2024.

[14] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *CoRL*, 2020.

[15] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[16] O. Biza, T. Kipf, D. Klee, R. Platt, J.-W. van de Meent, and L. L. Wong, "Factored world models for zero-shot generalization in robotic manipulation," *arXiv preprint arXiv:2202.05333*, 2022.

[17] S. James, M. Freese, and A. J. Davison, "Pyrep: Bringing v-rep to deep robot learning," *arXiv preprint arXiv:1906.11176*, 2019.

[18] G. Authors, "Genesis: A universal and generative physics engine for robotics and beyond," December 2024.

[19] B. R. Donald, "On information invariants in robotics," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 217–304, 1995.

[20] M. Ho, A. Farid, and A. Majumdar, "Towards a framework for comparing the complexity of robotic tasks," in *WAFR*, 2022.

[21] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *RAL*, 2020.

[22] C. Gao, Y. Jiang, and F. Chen, "Transferring hierarchical structures with dual meta imitation learning," in *CoRL*. PMLR, 2023.

[23] S. Haresh, D. Dijkman, A. Bhattacharyya, and R. Memisevic, "Clevrskills: Compositional language and visual reasoning in robotics," *NeurIPS*, 2024.

[24] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *ANNU REV CONTR ROBOT*, vol. 4, no. 1, pp. 265–293, 2021.

[25] J. D. Morrow and P. K. Khosla, "Manipulation task primitives for composing robot skills," in *Proceedings of International Conference on Robotics and Automation*, vol. 4. IEEE, 1997, pp. 3354–3359.

[26] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, "The grasp taxonomy of human grasp types," *IEEE Transactions on human-machine systems*, vol. 46, no. 1, pp. 66–77, 2015.

[27] D. Blanco-Mulero, Y. Dong, J. Borras, F. T. Pokorny, and C. Torras, "T-dom: A taxonomy for robotic manipulation of deformable objects," *arXiv preprint arXiv:2412.20998*, 2024.

[28] I. M. Bullock and A. M. Dollar, "Classifying human manipulation behavior," in *ICORR*. IEEE, 2011.

[29] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *arXiv*, 2024.

[30] Y. R. Wang, C. Ung, G. Tannert, J. Duan, J. Li, A. Le, R. Oswal, M. Grotz, W. Pumacay, Y. Deng *et al.*, "Roboeval: Where robotic manipulation meets structured and scalable evaluation," *arXiv preprint arXiv:2507.00435*, 2025.

[31] M. R. Cutkosky *et al.*, "On grasp choice, grasp models, and the design of hands for manufacturing tasks." *T-RO*, 1989.

[32] J. D. Morrow and P. K. Khosla, "Manipulation task primitives for composing robot skills," in *Proceedings of International Conference on Robotics and Automation*, vol. 4. IEEE, 1997, pp. 3354–3359.

[33] E. Huang, "Robotic manipulation primitives," Ph.D. dissertation, Carnegie Mellon University, 2021.

[34] D. Blanco-Mulero, Y. Dong, J. Borras, F. T. Pokorny, and C. Torras, "T-dom: A taxonomy for robotic manipulation of deformable objects," *arXiv preprint arXiv:2412.20998*, 2024.

[35] F. Krebs and T. Asfour, "A bimanual manipulation taxonomy," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, 2022.

[36] C. Bao, H. Xu, Y. Qin, and X. Wang, "Dexart: Benchmarking generalizable dexterous manipulation with articulated objects," in *ICCV*, 2023.

[37] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv:2009.12293*, 2020.

[38] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *RA-L*, 2022.

[39] R. Yang, H. Chen, J. Zhang *et al.*, "Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents," *arXiv preprint arXiv:2502.09560*, 2025.

[40] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," *IJRR*, p. 02783649241304789, 2023.

[41] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *ICAR*. IEEE, 2015.

[42] P. Atreya, K. Pertsch, T. Lee, M. J. Kim, A. Jain, A. Kuramshin, C. Eppner, C. Neary, E. Hu, F. Ramos *et al.*, "Roboarena: Distributed real-world evaluation of generalist robot policies," *arXiv*, 2025.

[43] Z. Zhou, P. Atreya, Y. L. Tan, K. Pertsch, and S. Levine, "Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world," *arXiv preprint arXiv:2503.24278*, 2025.

[44] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine, "Fmb: a functional manipulation benchmark for generalizable robotic learning," *IJRR*, 2025.

[45] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA urban challenge: autonomous vehicles in city traffic*. Springer Science & Business Media, 2009, vol. 56.

[46] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first amazon picking challenge," *TASE*, 2016.

[47] J. Wu, W. Chong, R. Holmberg, A. Prasad, Y. Gao, O. Khatib, S. Song, S. Rusinkiewicz, and J. Bohg, "Tidybot++: An open-source holonomic mobile manipulator for robot learning," *arXiv*, 2024.

[48] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," in *CVPR*, 2019.

[49] R. A. Potamias, J. Zhang, J. Deng, and S. Zafeiriou, "Wilor: End-to-end 3d hand localization and reconstruction in-the-wild," in *CVPR*, 2025.

[50] 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.