ManipBench: Benchmarking Vision-Language Models for Low-Level Robot Manipulation

Enyu Zhao* **Vedant Raval*** **Hejia Zhang*** **Jiageng Mao**
Zeyu Shangguan **Stefanos Nikolaidis** **Yue Wang** **Daniel Seita**
Department of Computer Science, University of Southern California

Abstract: Vision-Language Models (VLMs) have revolutionized artificial intelligence and robotics due to their commonsense reasoning capabilities. In robotic manipulation, VLMs are used primarily as high-level planners, but recent work has also studied their lower-level reasoning ability, which refers to making decisions about precise robot movements. However, the community currently lacks a clear and common benchmark that can evaluate how well VLMs can aid low-level reasoning in robotics. Consequently, we propose a novel benchmark, ManipBench, to evaluate the low-level robot manipulation reasoning capabilities of VLMs across various dimensions, including how well they understand object-object interactions and deformable object manipulation. We extensively test 33 representative VLMs across 10 model families on our benchmark, including variants to test different model sizes. Our evaluation shows that the performance of VLMs significantly varies across tasks, and there is a strong correlation between this performance and trends in our real-world manipulation tasks. It also shows that there remains a significant gap between these models and human-level understanding. See our website at: https://manipbench.github.io.

Keywords: Vision-Language Models, Robotics Benchmark, Robot Manipulation

1 Introduction

One long-standing goal in robotics is to train a "generalist" robot capable of performing diverse tasks, particularly robot manipulation. A promising paradigm for this is to leverage the broad knowledge in Vision-Language Models (VLMs) such as GPT-4 [1] and Gemini [2]. While the community has used VLMs to achieve great generalization in domains like computer vision and natural language processing, robotics faces unique challenges with requiring either difficult-to-scale physical real-world interaction data or simulation data with sim-to-real gaps, making it challenging for VLMs to act as low-level planners. However, recent work has extensively explored incorporating these "foundation" models [3] such that they can generate low-level trajectories executable by an embodiment [4, 5, 6, 7]. This direction is especially important because it offers a path to bypass large-scale, task-specific data collection by leveraging general-purpose pre-trained models. Beyond improving scalability, this enables faster deployment in open-world settings where generalization to unseen tasks and objects is critical. It remains unclear, however, which is the optimal foundation model for a "VLM agent" in tasks like fabric or articulated object manipulation, and how VLMs perform in low-level reasoning tasks required for manipulation.

Motivated from these questions, we propose ManipBench: a novel open-source benchmark to evaluate how well VLMs understand the low-level effect of a robot's action on its environment (see Fig. 1). While there exist benchmarks to evaluate VLMs for robotics [8, 9, 10, 11, 12, 13, 14, 15, 16], our approach and benchmark differ significantly along axes such as task diversity, model diversity, and particularly our novel multiple-choice question (MCQ) based evaluation design, which efficiently assesses the low-level reasoning capabilities of VLMs without requiring trajectory rollouts, as detailed in Table 1. We evaluate 33 VLMs across 10 families (2 closed-source and 8 open-source),

*Denotes a co-lead author.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

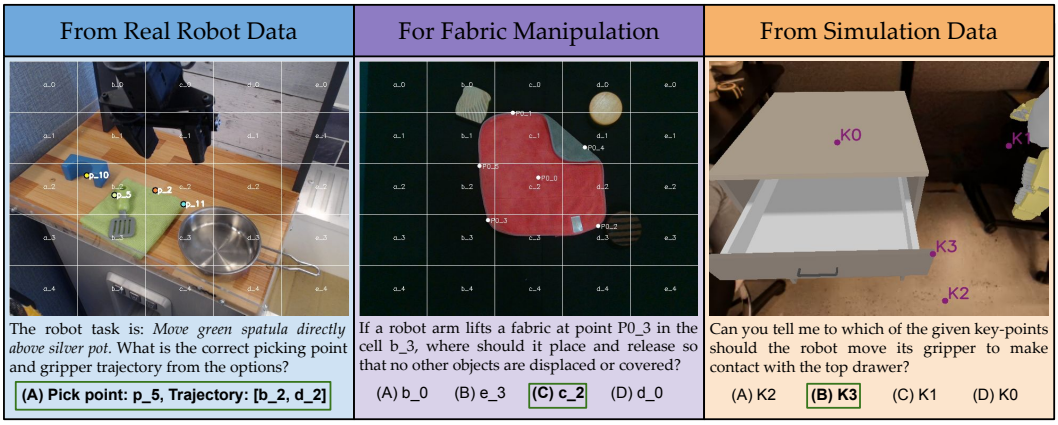


Figure 1: ManipBench is a novel benchmark with over 12,000 multiple-choice questions across three different categories to evaluate low-level physical reasoning capabilities of VLMs in the context of robotic manipulation. For the first question (from real robot data), we do not display the text of all options due to space limitations.

including size-based variants, to assess their strengths and weaknesses. The best models, such as Gemini-2.5-pro, significantly outperform random chance and other models on multiple choice questions, but still show substantial room for improvement, highlighting the need for further innovation in VLM development. Furthermore, we include experiments indicating a significant connection between performance on our benchmark versus performance in the real world when using VLMs to select robot actions, validating the consistency of ManipBench. In summary, this paper contributes:

1. ManipBench, an MCQ-based benchmark for evaluating VLMs' reasoning for low-level robotic manipulation, consisting of **12617** questions across tasks ranging from pick-and-place, articulated object manipulation, deformable object manipulation, and dynamic manipulation.
2. Extensive evaluation of various VLM families to assess different dimensions of reasoning.
3. Real world experiments on separate manipulation tasks, demonstrating a significant correlation between a model's ManipBench performance and its effectiveness at selecting robot actions.

2 Related Work

Vision-Language Models in Robotics. VLMs have increasingly become effective as high-level planners [17, 18, 19] which can produce executable robot code [20, 21, 22]. However, VLMs can have difficulty reasoning about spatial and physical properties [9], which are essential for manipulation [23]. This has led to work on improving their zero-shot spatial reasoning via methods such as iterative prompting [24] and annotating images [25]. Another line of work focuses on fine-tuning VLMs to improve spatial reasoning [26, 12, 27, 28]. In contrast, our objective is to systematically evaluate VLMs for predicting robot actions and to identify which VLMs are best as "agents" for a robot. We inspect the affordance reasoning capabilities of VLMs using the MOKA framework [6] which leverages VLMs to predict keypoints [29] (i.e., affordances) to define robot actions. VLMs are also closely related to robotic foundation models [30, 31], also referred to as Vision-Language-Action (VLA) models. Some examples of these include RT-2 [32], Octo [33], OpenVLA [34], and π_0 [35, 36]. Our work is complementary to these, as we focus on benchmarking VLMs which are not robotics-specific. In addition, evaluating general VLMs can help understand their impact on robotic foundation models if they are part of them (e.g., OpenVLA uses Llama 2 7B [37]).

Benchmarking LLMs and VLMs. Alongside the rapid advances in LLMs and VLMs, significant work has focused on benchmarking these models. Popular benchmarks evaluate mathematical reasoning [38, 39], trust and safety [40], and visual tasks like interpreting charts and maps [41]. Others, such as AgentBench [42], assess LLMs as agents in code, game, and web environments. In contrast, our benchmark centers on robotics, complementing existing efforts in VLM-based mapping and navigation [43], compositional reasoning [44], and task planning for embodied AI [45, 46, 47].

	Low-level Manip. Reasoning	Manip. Task Factorization	Perf. Trans. Eval.	Real World Data	Fabric Manipulation	Dynamic Manipulation	Model Diversity
PhysBench [9]	✗	✗	✓	✓	✗	✗	High
NEWTON [8]	✗	✗	✗	✗	✗	✗	Medium
VLABench [11]	✓	✓	✓	✗	✗	✗	Low
PhysObjects* [12]	✗	✗	N/A	✓	✗	✗	N/A
MultiNet [13]	✓	✗	✗	✓	✓	✓	Low
Octopi* [10]	✗	✗	N/A	✓	✗	✗	N/A
Open6DOR [14]	✓	✓	✗	✓	✗	✗	Medium
ETPBench [15]	✗	✓	✗	✗	✗	✗	Medium
GemBench* [16]	✓	✗	N/A	✗	✗	✗	N/A
ManipBench	✓	✓	✓	✓	✓	✓	High

Table 1: Comparison of ManipBench and other vision-language benchmarks or datasets (denoted with *). ManipBench is the only vision-language benchmark that evaluates low-level manipulation reasoning in VLMs using multiple-choice questions, enabling efficient and effective assessment. See Appendix A for more details.

More closely-related benchmarks, such as NEWTON [8] and Octopi [10], evaluate physical reasoning capabilities. Here, VLMs answer multiple choice questions about object properties (such as whether an object is "brittle" or "soft") from language and, for Octopi, high-resolution tactile data. Recently, the PhysBench [9] benchmark evaluates VLMs using multiple choice questions to test spatial reasoning capabilities. In contrast, ManipBench does not involve explicitly predicting object properties, and it evaluates how well VLMs can directly predict keypoints which define a low-level action for a robot. Other benchmarks for VLMs and robot manipulation include MultiNet [13] and VLABench [11]. MultiNet uses data from Open-X [48] and assesses how well VLMs predict trajectories using mean square error (MSE). However, using MSE for evaluation may not adequately measure performance in the case of multimodality. VLABench evaluates VLMs on manipulation tasks and assumes the presence of a skill library, whereas we do not use a skill library. In contrast to prior benchmarks, ManipBench also has a much greater focus on the important topic of deformable object manipulation [49, 50]. See Table 1 for an overview comparison.

Benchmarks and Datasets in Robot Manipulation. Robotics benchmarks and simulation environments are critical to evaluate algorithms and to measure progress in robot manipulation. Benchmarks for high-level planning with mobile manipulators include BEHAVIOR-1K [51], AI2-THOR [52], Habitat 2.0 [53], and RoboCasa [54]. Our main objective, however, is to study how well VLMs understand lower-level and more precise manipulation, though we could still leverage such simulators for data collection. Other manipulation benchmarks focus on low-level control using high-DOF hands [55] or humanoids [56, 57]; these are complementary and out of scope. Benchmarks closer in scope to ManipBench test deformable object manipulation [58, 59, 60]. ManipBench can leverage these to create questions for deformable manipulation reasoning capabilities.

More general manipulation task suites include MetaWorld [61], Ravens [62], CALVIN [63], RoboSuite [64], ManiSkill [65, 66], and RLBench [67]. These study aspects of robot manipulation and propose new simulation tasks and environments. Our benchmark is complementary; ManipBench contains questions to evaluate VLMs. Thus, as the community creates more task-related robotics benchmarks, these provide an expanding source of data for ManipBench. Furthermore, our benchmark uses large-scale data collected from the robotics community, including DROID [68] and Bridge [69] from Open-X [48]. Thus, much of our benchmark's data is already used in practice.

3 ManipBench: Overview

ManipBench comprises **12617** multiple-choice questions (MCQs) spanning diverse domains. These questions are categorized based on their origin (see Sec. 4): those derived from existing real-world datasets, those manually curated by us for fabric manipulation, and those sourced from simulation data. Our preliminary experiments reveal that questions centered on robot action trajectories and the keypoints guiding those trajectories provide the most valuable insights when evaluating VLMs. Thus, we use mark-based visual prompting to curate the MCQs, which primarily focus on select-

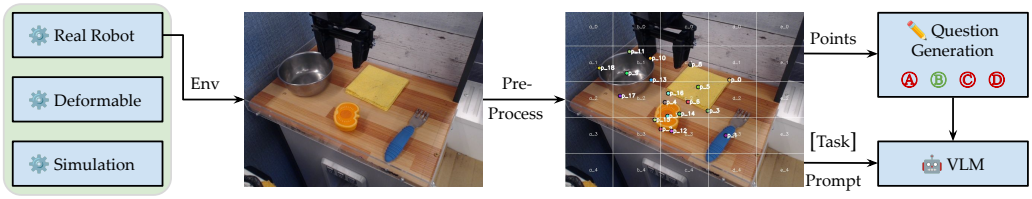


Figure 2: ManipBench uses real and simulated environments, typically pre-processed with a MOKA-style [6] pipeline to extract key-points and grid annotations for generating VLM-evaluable multiple choice questions.

ing appropriate *interaction keypoints*: contact-initiation (e.g., picking points), contact-release (e.g., placing points), and post-contact motion (e.g., pushing direction points). To achieve this, we process the collected data from each origin. The data, mainly consisting of the image observations (with some exceptions), is used to formulate the multiple-choice questions. The necessary information, such as the action trajectories, is used to guide the option generation, along with acting as the ground-truth for evaluating the VLMs. We incorporate a MOKA-style [6] pipeline for most of the image processing involved in ManipBench. See Fig. 2 for an overview.

4 ManipBench: Data Preparation

From public robotic manipulation datasets. We use demonstrations from the open-source robot manipulation datasets of DROID [68] and Bridge [69] for curating ManipBench questions. We study these datasets because of their importance in large-scale robotic imitation learning [48]. We separate the DROID data into two subsets based on the manipulation task: *DROID articulated (art.)* and *DROID pick-and-place (p&p)*. To prepare the data and generate questions, we extract the ground-truth gripper trajectories to obtain the picking and placing points by finetuning GroundingDINO [70] and leveraging the data from [71], which contains human-annotated gripper positions for affordance prediction across multiple datasets. See Appendix B.1 for details.

From in-house fabric manipulation setup. Given its practical importance [35, 72, 73], we manually curate data specifically designed to evaluate VLMs' understanding of fabric manipulation, such as folding or smoothing tasks. We break down common aspects of fabric manipulation into ten distinct dimensions, such as understanding of fabric-object interactions and understanding of inverse dynamics. Each dimension represents a fundamental aspect that an agent must implicitly grasp to successfully perform fabric manipulation. We get data from our real-world workstation. This comprises of a 3cm thick 98cm $\times$ 78cm foam, an Intel Realsense d415i RGBD camera mounted at a height of 87.2cm, rectangular fabrics, and solid objects of varying dimensions. This setup is used to capture top-down image observations of numerous scenes in different task settings for formulating questions (Sec. 5.3). The data collection procedure varies slightly across different dimensions. See Appendix B.2 for the description, choice, and importance of the dimensions.

From simulation. To assess manipulation tasks where real-world deployment may be cumbersome, such as tool use and dynamic manipulation, we compile a suite of simulated tasks that spans all task categories in ManipBench: (i) pick-and-place, (ii) articulated object manipulation, (iii) deformable object manipulation, (iv) tool manipulation, and (v) dynamic manipulation. We primarily adapt simulation assets, and pre-trained policies from popular existing benchmarks: SimplerEnv [74], RLBench [67], and SoftGym [59] to generate data for our evaluation questions (Sec. 5.4). We use environments from SimplerEnv for tasks (i) and (ii), while using SoftGym and RLBench, respectively, for tasks (iii) and (iv). For task (v), we construct a new ball-shooting environment in IsaacSim [75]. Additional details on data collection for each task are presented in Appendix B.3.

5 ManipBench: Question Generation

Using the data sources described in Sec. 4, we generate **12617** multiple-choice questions. We present preprocessing steps (Sec. 5.1) and how we generate the three types of MCQs (Sec. 5.2, 5.3, and 5.4).

5.1 Preprocessing Steps for Generating MCQs on Real World Data

For MCQs based on real world data, we begin by pre-processing image observations following MOKA [6]. Given an observation image and a natural language task description, we prompt GPT-4o to identify the key object O_K for the task. We use Grounded SAM [76] to segment all objects (O_K and any others) and obtain their image masks. From each object mask, we sample one point from the center and the rest from its contour using farthest-point sampling [77]. We then annotate the original observation image with the sampled points, and a grid overlay (usually 5×5). This annotated image encodes the necessary information of objects and environments, and will serve as the image prompt for VLMs (e.g., see Fig. 1, first two examples).

We use the prepared ground-truth keypoint p_g (e.g., pick and place point) to generate the correct choice for the multiple-choice questions. To ensure that the correct picking point lies on the key object O_K, we replace the ground-truth picking point p_{pick} with the closest point among those sampled from O_K, if the distance between those two points is lower than a threshold. The process to generate the natural language prompts for the VLMs, along with the incorrect options, is different for the three question types. To assist VLMs, we include CoT prompting [78] in the questions.

5.2 MCQs from Public Robotic Manipulation Datasets

For each episode, we identify the frames f_s and f_e where the manipulation "starts" and "ends," respectively. For each frame, we follow the pre-processing technique as described in Sec. 5.1 to construct vision-language question prompts for VLMs. We design two types of questions:

Type 1 (Q1): Given an image and language description of a task, the VLM selects the best matching trajectory from four candidates. Each trajectory includes a picking keypoint and two image tiles: the start tile (containing p_{pick}) and the end tile (containing p_{place}).

Type 2 (Q2): We derive type 2 questions from type 1 by having the VLM first choose a picking point from four candidates. Then, it selects the appropriate ending tile using the ground-truth picking point. Both are sourced from the type 1 candidate trajectories.

Overall, we have 9180 questions with 6120 type 1 questions and 3060 type 2 questions. Pure random guessing on type 1 questions will lead to a success rate of 25% since there are always 4 choices, while random guessing for type 2 questions will be worse than 25% performance since it combines predictions from picking and placing. We perform "question augmentation" to build several questions from the same episode. See Appendix C.1 for more details.

5.3 MCQs from In-house Fabric Manipulation Setup

We pre-process all the image observations following Sec. 5.1. We discard data where the preprocessing pipeline failed (e.g., Grounded SAM could not detect keypoints). With the desired keypoints and grids, we manually create choices for the multiple-choice questions. We generate 2662 questions across all the ten dimensions. We do not perform question augmentation, as done for the other question categories to encourage diverse scenarios being considered, given the simplicity of many dimensions. Thus, in general, we only form one question for each recorded scene. The questions vary across the dimensions, based on what they are trying to evaluate. Pure random guessing will result in 25% success given 4 choices. Additional details and sample questions are in Appendix C.2.

5.4 MCQs from Simulation

For the questions generated from existing simulation environments, we do not need to perform preprocessing steps (from Sec. 5.1) given the access to the ground-truth keypoint information. In this case, we plot the ground-truth point, along with sampling some incorrect points, on the image (Fig. 1, the third example). We do not overlay a grid pattern for these questions. We consider candidate object contact points and after-contact movements on the frames to generate evaluation questions. VLMs are required to select one of four candidate keypoints to complete the given manipulation tasks. Pure random guessing will result in 25% success for all tasks. See Appendix C.3 for details.

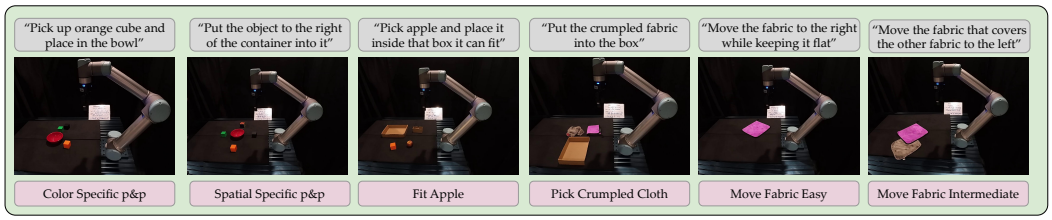


Figure 3: The different tasks, with their descriptions and initial states, present in our real-world experiments.

6 ManipBench: Evaluation Plan

We evaluate 10 VLM families on ManipBench (see Table 7 for details). This includes GPT and Gemini, two popular industry-backed closed-source model families widely regarded as among the most effective and versatile models in the Generative AI community. The other models are open-source and are known for their effective performance in multi-modal tasks (e.g., InternVL [79] and Qwen-VL [80]). These evaluations were conducted mainly on 2 RTX 4090 GPUs. Another server with 5 RTX 6000 Ada GPUs was used for larger models. Our evaluation metric is the percentage accuracy of correctly answering the questions. For each model and question type, we average over all the questions to report the numbers. As an upper bound baseline to compare with human intuition, we have 36 human volunteers evaluate these questions, through a custom web portal. The volunteers answer a subset of 50-70 questions for each question type, and we use their accuracies to obtain the final human score. We provide more details on the human evaluation in Appendix D.

Real-World Robot Manipulation. We also conduct physical robot manipulation experiments to verify whether performance on ManipBench translates to unseen robot tasks in real world. We design 7 manipulation tasks whose specific combinations of task goals, language instructions, camera angles, and objects never appear together in ManipBench: *Color Specific p&p, Spatial Specific p&p, Fit Apple, Pick Crumpled Cloth, Move Fabric Easy, Move Fabric Intermediate*, and *Move Fabric Hard* (Fig. 3)[2]. In these experiments, VLMs select keypoints to define robot actions. The pipeline for generating candidate keypoints for real world tasks is similar to the one for generating MCQs in ManipBench (Sec. 5.1). We use a UR5 robot arm with a Robotiq 2F-85 parallel-jaw gripper and a top-down RGBD camera. See Appendix E for more details about the real world experiments.

7 ManipBench: Results, Analysis, and Key Insights

Results on MCQs from public robotic manipulation datasets. Table 2 reports the performance of VLMs, random guessing, and human evaluation on the questions from public robotic manipulation datasets (Sec. 5.2). We use red to represent the highest accuracy, orange for the second-highest, and yellow for the third-highest, and we repeat this color code for subsequent tables. The Gemini-2.5-pro model attains the best performance across all three type 1 question categories, while o1 performs the best at type 2 questions for DROID (p&p). Among open-source VLMs, InternVL2.5-38B achieves the overall best performance across most questions. While the closed-source VLMs demonstrate a higher-than-random performance, all open-source VLMs with a size smaller than 2B perform similarly to random guessing, if not worse. See Appendix F for additional details.

Results on MCQs from in-house fabric manipulation setup. See Fig. 4 for a comparison of various VLMs' performance across multiple dimensions. Overall, VLMs outperform random guessing, suggesting a notable presence of physical reasoning capabilities in the context of fabric manipulation. Certain dimensions, like *Task Planning Understanding*, are easier for all models, while others, like *Fabric-Fabric Interaction Understanding*, are more challenging. However, consistent human accuracies across most dimensions suggest that the difficulty is not inherent, revealing gaps in VLMs' physical reasoning abilities. Among models, Gemini-2.5-pro consistently outperforms others, with o1 being a close contender. We also observe that InternVL2.5-78B usually outperforms

[2]The starting states for *Move Fabric Easy* and *Move Fabric Hard* are the same, so we only show one.

Model	Bridge		DROID (art.)		DROID (p&p)	
	Q1	Q2	Q1	Q2	Q1	Q2
Closed-Source						
o1	0.866	0.458	0.805	0.384	0.822	0.542
GPT-4.1	0.656	0.401	0.683	0.301	0.724	0.433
GPT-4o	0.503	0.309	0.687	0.396	0.676	0.401
GPT-4o-mini	0.487	0.183	0.524	0.156	0.572	0.210
Gemini-2.5-pro	0.916	0.403	0.909	0.362	0.869	0.453
Gemini-2.0-flash	0.594	0.287	0.645	0.387	0.560	0.235
Gemini-1.5-pro	0.398	0.309	0.414	0.126	0.384	0.158
Gemini-1.5-flash	0.458	0.347	0.378	0.123	0.324	0.193
Open-Source						
GLM-4V-9B	0.593	0.249	0.481	0.381	0.463	0.292
InternVL2-1B	0.015	0.028	0.000	0.105	0.000	0.066
InternVL2-2B	0.330	0.052	0.139	0.120	0.197	0.052
InternVL2-4B	0.341	0.134	0.261	0.283	0.317	0.242
InternVL2-8B	0.445	0.258	0.282	0.228	0.406	0.258
InternVL2-26B	0.634	0.326	0.545	0.293	0.593	0.296
InternVL2-40B	0.694	0.338	0.656	0.272	0.654	0.313
InternVL2-76B	0.748	0.509	0.601	0.235	0.632	0.411
InternVL2.5-1B	0.243	0.016	0.213	0.099	0.221	0.057
InternVL2.5-2B	0.404	0.026	0.182	0.096	0.244	0.027
InternVL2.5-4B	0.639	0.199	0.326	0.181	0.559	0.283
InternVL2.5-8B	0.600	0.330	0.348	0.229	0.511	0.272
InternVL2.5-26B	0.780	0.376	0.677	0.433	0.736	0.389
InternVL2.5-38B	0.904	0.528	0.829	0.316	0.839	0.425
InternVL2.5-78B	0.851	0.541	0.745	0.348	0.722	0.431
QwenVL-Chat	0.224	0.067	0.220	0.141	0.292	0.077
Qwen2VL-2B	0.237	0.045	0.223	0.156	0.204	0.089
Qwen2VL-7B	0.479	0.191	0.375	0.366	0.536	0.329
Qwen2VL-72B	0.670	0.460	0.645	0.395	0.742	0.460
Qwen2.5-VL-3B	0.428	0.190	0.335	0.144	0.512	0.197
Qwen2.5-VL-7B	0.548	0.344	0.430	0.372	0.602	0.408
Qwen2.5-VL-32B	0.649	0.428	0.632	0.399	0.574	0.459
Qwen2.5-VL-72B	0.809	0.470	0.809	0.390	0.796	0.481
LLaVA-NeXT-7B	0.228	0.071	0.111	0.094	0.206	0.100
Llama3.2-11B-VI	0.264	0.101	0.292	0.115	0.242	0.111
Random	0.250	0.061	0.250	0.063	0.250	0.084
Human	0.880	0.825	0.990	0.940	0.980	0.635

Table 2: Performance comparison of various VLMs on our MCQs for the Bridge and DROID datasets. Each dataset includes two question types detailed in Sec. 5.2.

Model	Place Carrot	Close Drawer	Straight. Rope	Sweep Object	Ball Shoot.
Closed-Source					
o1	0.776	0.627	0.721	0.608	0.593
GPT-4.1	0.798	0.687	0.800	0.644	0.506
GPT-4o	0.798	0.578	0.743	0.546	0.457
GPT-4o-mini	0.722	0.554	0.507	0.608	0.321
Gemini-2.5-pro	0.729	0.819	0.614	0.789	0.716
Gemini-2.0-flash	0.830	0.638	0.643	0.696	0.543
Gemini-1.5-pro	0.711	0.289	0.586	0.644	0.531
Gemini-1.5-flash	0.671	0.494	0.779	0.536	0.444
Open-Source					
GLM-4V-9B	0.404	0.422	0.450	0.732	0.086
InternVL2-1B	0.141	0.169	0.171	0.263	0.296
InternVL2-2B	0.451	0.084	0.357	0.340	0.173
InternVL2-4B	0.466	0.410	0.429	0.227	0.235
InternVL2-8B	0.534	0.289	0.414	0.278	0.309
InternVL2-26B	0.592	0.108	0.543	0.510	0.037
InternVL2-40B	0.567	0.494	0.429	0.309	0.222
InternVL2-76B	0.549	0.386	0.257	0.407	0.309
InternVL2.5-1B	0.329	0.253	0.379	0.381	0.309
InternVL2.5-2B	0.473	0.241	0.386	0.479	0.370
InternVL2.5-4B	0.505	0.277	0.429	0.469	0.246
InternVL2.5-8B	0.527	0.229	0.414	0.247	0.259
InternVL2.5-26B	0.635	0.277	0.571	0.418	0.346
InternVL2.5-38B	0.704	0.482	0.636	0.459	0.370
InternVL2.5-78B	0.507	0.578	0.507	0.474	0.407
QwenVL-Chat	0.458	0.277	0.214	0.356	0.309
Qwen2VL-2B	0.343	0.277	0.214	0.562	0.506
Qwen2VL-7B	0.596	0.518	0.514	0.521	0.148
Qwen2VL-72B	0.668	0.470	0.721	0.423	0.444
Qwen2.5-VL-3B	0.567	0.325	0.329	0.521	0.222
Qwen2.5-VL-7B	0.574	0.506	0.679	0.577	0.407
Qwen2.5-VL-32B	0.635	0.506	0.457	0.500	0.494
Qwen2.5-VL-72B	0.661	0.482	0.621	0.495	0.704
LLaVA-NeXT-7B	0.466	0.108	0.271	0.505	0.247
Llama3.2-11B-VI	0.585	0.410	0.486	0.665	0.210
Random	0.250	0.250	0.250	0.250	0.250

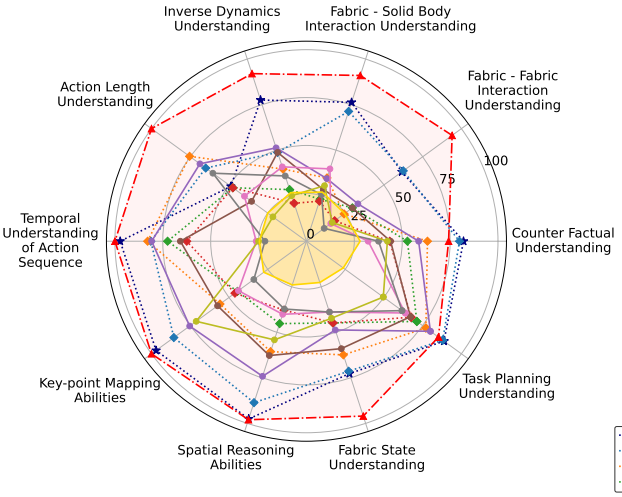
Table 3: Performance comparison of various VLMs on our MCQs for the simulation tasks. The tasks include *Place Carrot*, *Close Drawer*, *Straighten Rope*, *Sweep Object*, and *Ball Shooting*.

the smaller open-source models, though it trails behind Gemini-2.5-pro and o1. Furthermore, the high standard deviations of the model accuracies across different dimensions indicate their effectiveness in distinguishing the models' low-level inference abilities, particularly in dimensions like *Temporal Understanding of Action Sequence* and *Spatial Reasoning Abilities*, justifying our choice of dimensions. Performance of all models on these questions is detailed in Appendix G.

Results on MCQs from simulation. Table 3 reports the performance of VLMs on MCQs from simulation. While Gemini-2.5-pro still has the overall best performance among closed-source models, we observe that certain models excel in specific tasks. For instance, Gemini-2.0-flash achieves the best performance on the *Place Carrot* task, suggesting that some models may be better optimized for certain tasks. Among open-source models, larger variants frequently outperform smaller ones in the same family, though the rate of improvement varies based on the task.

Manipulation task category breakdown. ManipBench covers five task categories: pick-and-place, articulated object, deformable object, tool, and dynamic manipulation. Aggregating over all model accuracies in each task category (w/o Q2 MCQs from public datasets) reveals that existing VLMs handle pick-and-place comparatively well, achieving a mean accuracy of 0.525. In contrast, articulated object and dynamic manipulation are more challenging, with accuracies of 0.396 and 0.357.

Model robustness across task categories. We compute each model's coefficient of variation (CV), across all manipulation task categories to analyze its robustness, an ability to perform accurately in different contexts (lower CV values are better). We find that Gemini-2.5-pro is the most robust closed-source model with a CV of 0.089. Qwen2.5-VL-32B is the most robust open-source model (0.085); while it has slightly worse average accuracy than its larger variant (Qwen2.5-VL-72B with a CV of 0.133), it has smaller standard deviation.



Dimension	Std. Deviation
Task Planning Understanding	11.59
Fabric State Understanding	12.28
Spatial Reasoning Abilities	20.23
Key-point Mapping Abilities	19.40
Temporal Understanding of Action Sequence	26.08
Action Length Understanding	15.65
Inverse Dynamics Understanding	15.51
Fabric - Solid Body Interaction Understanding	18.38
Fabric - Fabric Interaction Understanding	17.22
Counter Factual Understanding	16.28

Figure 4: The percentage accuracies of the VLMs for evaluating the dimensions of Fabric Manipulation, depicted as a Radar Chart, along with the standard deviation of their performances across these dimensions.

Model	Color Specific p&p	Spatial Specific p&p	Fit Apple	Pick Crumpled Cloth	Move Fabric Easy	Move Fabric Intermediate	Move Fabric Hard	Total Success Rate
o1	1/3	3/3	2/3	3/3	3/3	3/3	0/3	15/21
GPT-4o	2/3	2/3	3/3	3/3	3/3	0/3	1/3	14/21
Gemini-2.5-pro	3/3	3/3	3/3	3/3	3/3	3/3	0/3	18/21
Gemini-1.5-pro	2/3	0/3	0/3	1/3	2/3	0/3	0/3	5/21
InternVL2.5-78B	2/3	2/3	1/3	2/3	3/3	2/3	0/3	12/21
InternVL2.5-26B	1/3	2/3	2/3	1/3	2/3	3/3	0/3	8/21
InternVL2.5-8B	2/3	2/3	0/3	0/3	1/3	0/3	0/3	5/21
GLM-4V-9B	1/3	0/3	0/3	1/3	0/3	0/3	0/3	2/21

Table 4: Performance comparisons for the real-world experiments with different VLMs via the success rate.

7.1 Performance Transfer to Unseen Real World Robot Experiments

To demonstrate the utility of ManipBench, we evaluate a set of selected VLMs on 7 unseen real world manipulation tasks (Sec. 6). We perform statistical analysis to assess the correlation between VLM performance on our MCQ benchmark and their effectiveness as real-world robotic agents (Table. 4). On aggregating the accuracies over the different question-types, we observe a Pearson's coefficient of 0.889 ($p = 0.003$), a Spearman's coefficient of 0.850 ($p = 0.007$), and a Kendall's Tau of 0.691 ($p = 0.018$). These results indicate a strong positive correlation between MCQ benchmark performance and real-world effectiveness. The statistically significant ($p < 0.05$) Pearson's coefficient highlights strong linear alignment, while the Spearman and Kendall coefficients confirm robust monotonic agreement with high confidence in rank-order consistency. Collectively, these trends support the utility of ManipBench as a reliable proxy for evaluating VLMs in embodied robotic settings. See Appendix H for details on correlations for each question type with the real-world experiments.

8 Conclusion

In this work, we propose a novel benchmark, ManipBench. This is a robotics-focused benchmark which critically analyzes modern VLMs and their ability to reason about object properties and precise movements for robotic manipulation. Despite the presence of better than random reasoning capabilities, our results indicate that robot manipulation understanding across various models is relatively poor, leaving much room for improvement. We plan to maintain this benchmark and to further improve it by addressing some of its current limitations. We hope that this work helps to facilitate a better understanding of VLMs as they continue to play a bigger role in robotic manipulation.

9 Limitations

While we believe ManipBench is a valuable benchmark for VLMs and robotics, there are some limitations that suggest opportunities for future work. First, due to limited computational and financial resources, we did not conduct experiments on all possible closed-source and open-source model variants, which may draw an incomplete conclusion. Second, MCQ evaluations depend on access to potential correct/incorrect options to choose from. In real-world scenarios where VLMs are employed as agents, they might not always be able to choose among pre-selected options. Third, the benchmark is not exhaustive across different tasks and excludes crucial deformable manipulation tasks that can require low-level reasoning, such as robotic bag manipulation [81, 82, 83]. Fourth, although we demonstrate a strong correlation between benchmark performance and real-world experiment performance, our current setup leverages VLMs in the same way across both settings where the VLMs select an answer or action from a predefined list of options. A natural next step would have been to explore alternative approaches for real-world experiments, such as requiring VLMs to generate actions directly rather than choosing from fixed options. Finally, ManipBench lacks emphasis on higher-level aspects (e.g., task planning) since it is focused on low-level reasoning.

Acknowledgments

We thank our colleagues who gave us helpful feedback, including Minjune Hwang and Rajas Chitale. We used LLMs to help with proofreading our writing and manually verified all such content.

References

[1] OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] G. T. Google. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805*, 2023.

[3] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. S. Chatterji, A. S. Chen, K. A. Creel, J. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. E. Gillespie, K. Goel, N. D. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. F. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. S. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. P. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. F. Nyarko, G. Ogut, L. Orr, I. Papadilitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. H. Roohani, C. Ruiz, J. Ryan, C. R'e, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. P. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. A. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang. On the opportunities and risks of foundation models. *ArXiv*, 2021. URL https://crfm.stanford.edu/assets/report.pdf.

[4] T. Kwon, N. D. Palo, and E. Johns. Language Models as Zero-Shot Trajectory Generators. In *IEEE Robotics and Automation Letters (RA-L)*, 2024.

[5] V. Raval, E. Zhao, H. Zhang, S. Nikolaidis, and D. Seita. GPT-Fabric: Smoothing and Folding Fabric by Leveraging Pre-Trained Foundation Models. In *International Symposium on Robotics Research (ISRR)*, 2024.

[6] K. Fang, F. Liu, P. Abbeel, and S. Levine. MOKA: Open-World Robotic Manipulation through Mark-Based Visual Prompting. In *Robotics: Science and Systems (RSS)*, 2024.

[7] Z. Wang, R. Shen, and B. C. Stadie. Wonderful Team: Zero-Shot Physical Task Planning with Visual LLMs. In *Transactions on Machine Learning Research*, 2025.

[8] Y. R. Wang, J. Duan, D. Fox, and S. Srinivasa. NEWTON: Are Large Language Models Capable of Physical Reasoning? In *Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[9] W. Chow, J. Mao, B. Li, D. Seita, V. Guizilini, and Y. Wang. PhysBench: Benchmarking and Enhancing Vision-Language Models for Physical World Understanding. In *International Conference on Learning Representations (ICLR)*, 2025.

[10] S. Yu, K. Lin, A. Xiao, J. Duan, and H. Soh. Octopi: Object Property Reasoning with Large Tactile-Language Models. In *Robotics: Science and Systems (RSS)*, 2024.

[11] S. Zhang, Z. Xu, P. Liu, X. Yu, Y. Li, Q. Gao, Z. Fei, Z. Yin, Z. Wu, Y.-G. Jiang, and X. Qiu. VLABench: A Large-Scale Benchmark for Language-Conditioned Robotics Manipulation with Long-Horizon Reasoning Tasks. *arXiv preprint arXiv:2412.18194*, 2024.

[12] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh. Physically Grounded Vision-Language Models for Robotic Manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[13] P. Guruprasad, H. Sikka, J. Song, Y. Wang, and P. P. Liang. Benchmarking Vision, Language, and Action Models on Robotic Learning Tasks. *arXiv preprint arXiv:2411.05821*, 2024.

[14] Y. Ding, H. Geng, C. Xu, X. Fang, J. Zhang, S. Wei, Q. Dai, Z. Zhang, and H. Wang. Open6dor: Benchmarking open-instruction 6-dof object rearrangement and a vlm-based approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[15] X. Fu, M. Zhang, P. Han, H. Zhang, L. Shi, H. Tang, et al. What can vlms do for zero-shot embodied task planning? In *ICML 2024 Workshop on LLMs and Cognition*, 2024.

[16] R. Garcia, S. Chen, and C. Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. In *International Conference on Robotics and Automation (ICRA)*, 2025.

[17] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Conference on Robot Learning (CoRL)*, 2022.

[18] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence. PaLM-E: An Embodied Multimodal Language Model. In *International Conference on Machine Learning (ICML)*, 2023.

[19] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents. In *International Conference on Machine Learning (ICML)*, 2022.

[20] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as Policies: Language Model Programs for Embodied Control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[21] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg. ProgPrompt: Program generation for situated robot task planning using large language models. *Autonomous Robots (AURO)*, 2023.

[22] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei. VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models. In *Conference on Robot Learning (CoRL)*, 2023.

[23] R. Tedrake. *Robotic Manipulation*. 2024. URL http://manipulation.mit.edu.

[24] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, Q. Vuong, T. Zhang, T.-W. E. Lee, K.-H. Lee, P. Xu, S. Kirmani, Y. Zhu, A. Zeng, K. Hausman, N. Heess, C. Finn, S. Levine, and B. Ichter. PIVOT: Iterative Visual Prompting Elicits Actionable Knowledge for VLMs. In *International Conference on Machine Learning (ICML)*, 2024.

[25] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao. Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V. *arXiv preprint arXiv:2310.11441*, 2023.

[26] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia. SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[27] G. Tang, S. Rajkumar, Y. Zhou, H. R. Walke, S. Levine, and K. Fang. KALIE: Fine-Tuning Vision-Language Models for Open-World Manipulation without Robot Data. *arXiv preprint arXiv:2409.14066*, 2024.

[28] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox. RoboPoint: A Vision-Language Model for Spatial Affordance Prediction for Robotics. In *Conference on Robot Learning (CoRL)*, 2024.

[29] L. Manuelli, W. Gao, P. R. Florence, and R. Tedrake. kPAM: KeyPoint Affordances for Category Level Manipulation. In *International Symposium on Robotics Research (ISRR)*, 2019.

[30] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman, B. Ichter, D. Driess, J. Wu, C. Lu, and M. Schwager. Foundation Models in Robotics: Applications, Challenges, and the Future. *arXiv preprint arXiv:2312.07843*, 2023.

[31] Y. Hu, Q. Xie, V. Jain, J. Francis, J. Patrikar, N. Keetha, S. Kim, Y. Xie, T. Zhang, S. Zhao, Y. Q. Chong, C. Wang, K. Sycara, M. Johnson-Roberson, D. Batra, X. Wang, S. Scherer, Z. Kira, F. Xia, and Y. Bisk. Toward General-Purpose Robots via Foundation Models: A Survey and Meta-Analysis. *arXiv preprint arXiv:2312.08782*, 2023.

[32] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Conference on Robot Learning (CoRL)*, 2023.

[33] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An Open-Source Generalist Robot Policy. In *Robotics: Science and Systems (RSS)*, 2024.

[34] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. OpenVLA: An Open-Source Vision-Language-Action Model. In *Conference on Robot Learning (CoRL)*, 2024.

[35] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

[36] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. *arXiv preprint arXiv:2501.09747*, 2025.

[37] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023.

[38] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[39] B. Gao, F. Song, Z. Yang, Z. Cai, Y. Miao, Q. Dong, L. Li, C. Ma, L. Chen, R. Xu, Z. Tang, B. Wang, D. Zan, S. Quan, G. Zhang, L. Sha, Y. Zhang, X. Ren, T. Liu, and B. Chang. Omni-MATH: A Universal Olympiad Level Mathematic Benchmark For Large Language Models. *arXiv preprint arXiv:2410.07985*, 2024.

[40] Y. Huang, L. Sun, H. Wang, S. Wu, Q. Zhang, Y. Li, C. Gao, Y. Huang, W. Lyu, Y. Zhang, X. Li, H. Sun, Z. Liu, Y. Liu, Y. Wang, Z. Zhang, B. Vidgen, B. Kailkhura, C. Xiong, C. Xiao, C. Li, E. P. Xing, F. Huang, H. Liu, H. Ji, H. Wang, H. Zhang, H. Yao, M. Kellis, M. Zitnik, M. Jiang, M. Bansal, J. Zou, J. Pei, J. Liu, J. Gao, J. Han, J. Zhao, J. Tang, J. Wang, J. Vanschoren, J. Mitchell, K. Shu, K. Xu, K.-W. Chang, L. He, L. Huang, M. Backes, N. Z. Gong, P. S. Yu, P.-Y. Chen, Q. Gu, R. Xu, R. Ying, S. Ji, S. Jana, T. Chen, T. Liu, T. Zhou, W. Y. Wang, X. Li, X. Zhang, X. Wang, X. Xie, X. Chen, X. Wang, Y. Liu, Y. Ye, Y. Cao, Y. Chen, and Y. Zhao. TrustLLM: Trustworthiness in Large Language Models. In *International Conference on Machine Learning (ICML)*, 2024.

[41] B. Li, Y. Ge, Y. Chen, Y. Ge, R. Zhang, and Y. Shan. SEED-Bench-2-Plus: Benchmarking Multimodal Large Language Models with Text-Rich Visual Comprehension. *arXiv preprint arXiv:2404.16790*, 2024.

[42] X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang, S. Zhang, X. Deng, A. Zeng, Z. Du, C. Zhang, S. Shen, T. Zhang, Y. Su, H. Sun, M. Huang, Y. Dong, and J. Tang. AgentBench: Evaluating LLMs as Agents. In *International Conference on Learning Representations (ICLR)*, 2024.

[43] P. Ding, J. Fang, P. Li, K. Wang, X. Zhou, M. Yu, J. Li, M. R. Walter, and H. Mei. MANGO: A Benchmark for Evaluating Mapping and Navigation Abilities of Large Language Models. *arXiv preprint arXiv:2403.19913*, 2024.

[44] K. Zheng, X. Chen, O. Jenkins, and X. E. Wang. VLMbench: A Compositional Benchmark for Vision-and-Language Manipulation. In *Neural Information Processing Systems (NeurIPS)*, 2022.

[45] J.-W. Choi, Y. Yoon, H. Ong, J. Kim, and M. Jang. LoTa-Bench: Benchmarking Language-oriented Task Planners for Embodied Agents. In *International Conference on Learning Representations (ICLR)*, 2024.

[46] M. Li, S. Zhao, Q. Wang, K. Wang, Y. Zhou, S. Srivastava, C. Gokmen, T. Lee, L. E. Li, R. Zhang, W. Liu, P. Liang, L. Fei-Fei, J. Mao, and J. Wu. Embodied Agent Interface: Benchmarking LLMs for Embodied Decision Making. In *Neural Information Processing Systems (NeurIPS)*, 2024.

[47] L. Zhang, Y. Wang, H. Gu, A. Hamidizadeh, Z. Zhang, Y. Liu, Y. Wang, D. G. A. Bravo, J. Dong, S. Zhou, T. Cao, Y. Zhuang, Y. Zhang, and J. Hao. ET-Plan-Bench: Embodied Task-level Planning Benchmark Towards Spatial-Temporal Cognition with Foundation Models. *arXiv preprint arXiv:2410.14682*, 2024.

[48] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[49] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar. Robotic Manipulation and Sensing of Deformable Objects in Domestic and Industrial Applications: a Survey. In *International Journal of Robotics Research (IJRR)*, 2018.

[50] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, et al. Challenges and Outlook in Robotic Manipulation of Deformable Objects. *IEEE Robotics and Automation Magazine*, 2021.

[51] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, W. Ai, B. Martinez, H. Yin, M. Lingelbach, M. Hwang, A. Hiranaka, S. Garlanka, A. Aydin, S. Lee, J. Sun, M. Anvari, M. Sharma, D. Bansal, S. Hunter, K.-Y. Kim, A. Lou, C. R. Matthews, I. Villa-Renteria, J. H. Tang, C. Tang, F. Xia, Y. Li, S. Savarese, H. Gweon, C. K. Liu, J. Wu, and L. Fei-Fei. BEHAVIOR-1K: A Human-Centered, Embodied AI Benchmark with 1,000 Everyday Activities and Realistic Simulation. In *Conference on Robot Learning (CoRL)*, 2022.

[52] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

[53] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra. Habitat 2.0: Training Home Assistants to Rearrange their Habitat. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[54] 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. In *Robotics: Science and Systems (RSS)*, 2024.

[55] K. Zakka, P. Wu, L. Smith, N. Gileadi, T. Howell, X. B. Peng, S. Singh, Y. Tassa, P. Florence, A. Zeng, and P. Abbeel. RoboPianist: Dexterous Piano Playing with Deep Reinforcement Learning. In *Conference on Robot Learning (CoRL)*, 2023.

[56] N. Chernyadev, N. Backshall, X. Ma, Y. Lu, Y. Seo, and S. James. BiGym: A Demo-Driven Mobile Bi-Manual Manipulation Benchmark. In *Conference on Robot Learning (CoRL)*, 2024.

[57] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel. HumanoidBench: Simulated Humanoid Benchmark for Whole-Body Locomotion and Manipulation. In *Robotics: Science and Systems (RSS)*, 2024.

[58] C. Lin, J. Fan, Y. Wang, Z. Yang, Z. Chen, L. Fang, T.-H. Wang, Z. Xian, and C. Gan. UBSoft: A Simulation Platform for Robotic Skill Learning in Unbounded Soft Environments. In *Conference on Robot Learning (CoRL)*, 2024.

[59] X. Lin, Y. Wang, J. Olkin, and D. Held. SoftGym: Benchmarking Deep Reinforcement Learning for Deformable Object Manipulation. In *Conference on Robot Learning (CoRL)*, 2020.

[60] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng. Learning to Rearrange Deformable Cables, Fabrics, and Bags with Goal-Conditioned Transporter Networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[61] 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 *Conference on Robot Learning (CoRL)*, 2019.

[62] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee. Transporter Networks: Rearranging the Visual World for Robotic Manipulation. In *Conference on Robot Learning (CoRL)*, 2020.

[63] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks. In *IEEE Robotics and Automation Letters (RA-L)*, 2022.

[64] 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. In *arXiv preprint arXiv:2009.12293*, 2020.

[65] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su. ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills. In *International Conference on Learning Representations (ICLR)*, 2023.

[66] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T. kai Chan, Y. Gao, X. Li, T. Mu, N. Xiao, A. Gurha, Z. Huang, R. Calandra, R. Chen, S. Luo, and H. Su. ManiSkill3: GPU Parallelized Robotics Simulation and Rendering for Generalizable Embodied AI. *arXiv preprint arXiv:2410.00425*, 2024.

[67] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. RLBench: The Robot Learning Benchmark and Learning Environment. In *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[68] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Robotics: Science and Systems (RSS)*, 2024.

[69] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine. BridgeData V2: A Dataset for Robot Learning at Scale. In *Conference on Robot Learning (CoRL)*, 2023.

[70] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding DINO: Marrying DINO with Grounded Pre-training for Open-set Object Detection. In *European Conference on Computer Vision (ECCV)*, 2024.

[71] Y. Kuang, J. Ye, H. Geng, J. Mao, C. Deng, L. Guibas, H. Wang, and Y. Wang. RAM: Retrieval-Based Affordance Transfer for Generalizable Zero-Shot Robotic Manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

[72] J. Borràs, G. Alenyà, and C. Torras. A Grasping-Centered Analysis for Cloth Manipulation. *IEEE Transactions on Robotics*, 2020.

[73] A. Longhini, Y. Wang, I. Garcia-Camacho, D. Blanco-Mulero, M. Moletta, M. Welle, G. Alenyà, H. Yin, Z. Erickson, D. Held, J. Borràs, and D. Kragic. Unfolding the Literature: A Review of Robotic Cloth Manipulation. *Annual Review of Control, Robotics, and Autonomous System*, 2024.

[74] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, S. Levine, J. Wu, C. Finn, H. Su, Q. Vuong, and T. Xiao. Evaluating Real-World Robot Manipulation Policies in Simulation. In *Conference on Robot Learning (CoRL)*, 2024.

[75] NVIDIA Corporation. NVIDIA Isaac Sim. https://developer.nvidia.com/isaac/sim, 2025. Version 4.5.0.

[76] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[77] C. Moenning and N. A. Dodgson. Fast Marching Farthest Point Sampling. Technical report, University of Cambridge, Computer Laboratory, 2003.

[78] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint arXiv:2201.11903*, 2022.

[79] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, B. Li, P. Luo, T. Lu, Y. Qiao, and J. Dai. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. *arXiv preprint arXiv:2312.14238*, 2023.

[80] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966*, 2023.

[81] L. Y. Chen, B. Shi, D. Seita, R. Cheng, T. Kollar, D. Held, K. Goldberg, K. Goldberg, K. Goldberg, K. Goldberg, K. Goldberg, and K. Goldberg. AutoBag: Learning to Open Plastic Bags and Insert Objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[82] L. X. Shi, Z. Hu, T. Z. Zhao, A. Sharma, K. Pertsch, J. Luo, S. Levine, and C. Finn. Yell At Your Robot: Improving On-the-Fly from Language Corrections. In *Robotics: Science and Systems (RSS)*, 2024.

[83] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Robotics: Science and Systems (RSS)*, 2023.

[84] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. SAM 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714*, 2024.

[85] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[86] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

[87] W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song, J. Xu, B. Xu, J. Li, Y. Dong, M. Ding, and J. Tang. CogVLM: Visual Expert for Pretrained Language Models. *arXiv preprint arXiv:2311.03079*, 2023.

[88] T. GLM, A. Zeng, B. Xu, B. Wang, C. Zhang, D. Yin, D. Rojas, G. Feng, H. Zhao, H. Lai, H. Yu, H. Wang, J. Sun, J. Zhang, J. Cheng, J. Gui, J. Tang, J. Zhang, J. Li, L. Zhao, L. Wu, L. Zhong, M. Liu, M. Huang, P. Zhang, Q. Zheng, R. Lu, S. Duan, S. Zhang, S. Cao, S. Yang, W. L. Tam, W. Zhao, X. Liu, X. Xia, X. Zhang, X. Gu, X. Lv, X. Liu, X. Liu, X. Yang, X. Song, X. Zhang, Y. An, Y. Xu, Y. Niu, Y. Yang, Y. Li, Y. Bai, Y. Dong, Z. Qi, Z. Wang, Z. Yang, Z. Du, Z. Hou, and Z. Wang. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *arXiv preprint arXiv:2406.12793*, 2024.

[89] Z. Chen, W. Wang, H. Tian, S. Ye, Z. Gao, E. Cui, W. Tong, K. Hu, J. Luo, Z. Ma, et al. How Far Are We to GPT-4V? Closing the Gap to Commercial Multimodal Models with Open-Source Suites. *arXiv preprint arXiv:2404.16821*, 2024.

[90] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[91] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023.

[92] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[93] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[94] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual Instruction Tuning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

[95] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved Baselines with Visual Instruction Tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[96] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

[97] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Wyatt, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Guzmán, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Thattai, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Zhang, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Prasad, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, K. Lakhotia, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Tsimpoukelli, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, N. Zhang, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan,

R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Maheswari, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Albiero, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. Wang, X. E. Tan, X. Xia, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Srivastava, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Teo, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Dong, A. Franco, A. Goyal, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Liu, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, C. Gao, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E.-T. Le, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Kokkinos, F. Ozgenel, F. Caggioni, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Inan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, H. Zhan, I. Damlaj, I. Molybog, I. Tufanov, I. Leontiadis, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Lam, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Jagadeesh, K. Huang, K. Chawla, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. Liu, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. Mehta, N. P. Laptev, N. Dong, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Parthasarathy, R. Li, R. Hogan, R. Battey, R. Wang, R. Howes, R. Rinott, S. Mehta, S. Siby, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Mahajan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Patil, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Deng, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Koehler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wu, X. Wang, X. Wu, X. Gao, Y. Kleinman, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Zhao, Y. Hao, Y. Qian, Y. Li, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, Z. Zhao, and Z. Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[98] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[99] A. Steiner, A. S. Pinto, M. Tschannen, D. Keysers, X. Wang, Y. Bitton, A. Gritsenko, M. Minderer, A. Sherbondy, S. Long, et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

[100] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[101] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Appendix

Table of Contents

A Additional Details about Table 1

In Table 1, we compare ManipBench with 9 leading vision-language benchmarks or datasets [8, 9, 10, 11, 12, 13, 14, 15, 16] in the following 7 dimensions:

1. *Low-level Manipulation Reasoning*, assessing whether included evaluations demand precise action-centric reasoning rather than solely high-level judgments (e.g., object attributes, task planning, or spatial relationships).
2. *Manipulation Task Factorization*, indicating whether evaluations are decomposed along orthogonal axes (e.g., required manipulation skills, fabric state understanding, spatial reasoning) to enable systematic analysis.
3. *Performance Transfer Evaluation*, denoting the presence of dedicated splits for measuring generalization to unseen task variations.
4. *Real World Data*, referring to the inclusion of evaluation questions sourced from real-world robot manipulation setups.
5. *Fabric Manipulation*, referring to the inclusion of evaluations on fabric manipulation tasks.
6. *Dynamic Manipulation*, referring to the inclusion of evaluations on dynamic manipulation tasks.
7. *Model Diversity*, measuring the breadth of models evaluated: "low" if fewer than five, "high" if more than ten.

For datasets such as PhysObjects [12], Octopi [10], and GemBench [16], several fields remain blank because these resources were introduced as data collections rather than comprehensive benchmarks.

B Additional Details about Data Preparation

B.1 From Public Robotic Manipulation Datasets

To effectively utilize data from the existing public robotic manipulation datasets of DROID [68] and Bridge [69] for ManipBench, we require accurate gripper positions to obtain the picking and placing points. Although DROID provides RGBD images and the camera intrinsic matrix for calculating the gripper's position, we found a notable empirical difference from the calculation to the ground truth. Hence, we utilize the annotated DROID subset from [71], retaining only the successful rollouts for question generation. Additionally, we employ the Bridge-V2 [69] dataset from the Open-X collection [48], sampling 450 successful rollouts. We exclude 96 rollouts in total, of which 45 is due to how the Grounding DINO model fails to detect the key object O_K. In the other 51 rollouts, either the task description does not match the rollout video or it is not a pick-and-place task.

We further separate the DROID data into two subsets based on the manipulation task. In the "articulate manipulation" subset, the gripper trajectories are complete and thus we directly use them. However, in the "pick-and-place manipulation" subset, only the gripper trajectory in the picking phase is provided. To complete the trajectories from [71], we fine-tune Grounding DINO [70] to detect the gripper, as shown in Fig. 5. We annotate about 50 episodes (each with 2 to 10 frames) to fine-tune the Grounding DINO model. For each episode, we manually provide the gripper mask in one frame and then use Segment Anything 2 (SAM2) [84] to track the gripper throughout the episode to efficiently generate image-mask pairs. We extract the gripper trajectories in the Bridge data in a similar fashion.

B.2 From In-house Fabric Manipulation Setup

As described in Table 5, we break down common aspects of fabric manipulation into ten distinct dimensions. This list is not exhaustive and does not encompass the entire range of fabric manipulation tasks [73]. However, we believe a strong understanding of these dimensions correlates with fabric manipulation performance.

For instance, understanding how different objects in a scene interact with each other is crucial when performing fabric manipulation tasks in cluttered environments. Furthermore, the nature of this

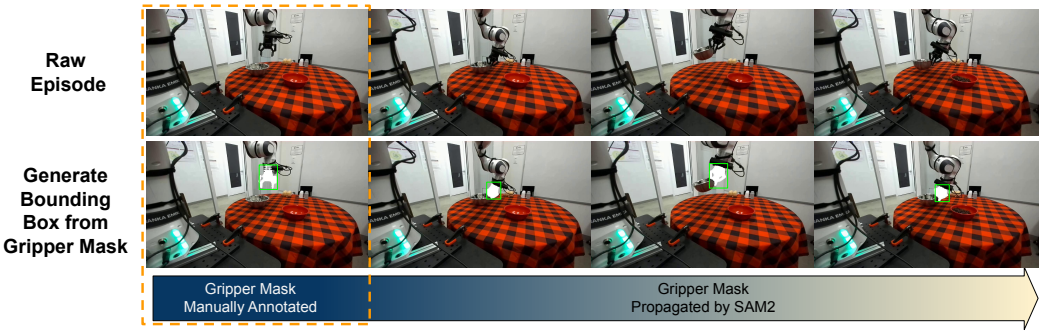


Figure 5: Raw frames (top) are paired with gripper masks (bottom), initially annotated manually (see orange box) and then propagated using SAM2 [84]. Bounding boxes derived from the masks are used for fine-tuning Grounding DINO [70].

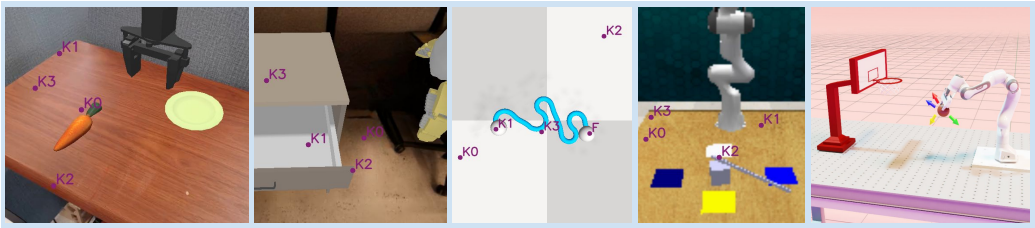


Figure 6: Simulated tasks from ManipBench: *Place Carrot*, *Close Drawer*, *Straighten Rope*, *Sweep Object*, *Ball Shooting*. For the first four tasks, we use the keypoints (`K0, K1, K2, K3`) generated from demonstrations to encode robot actions and enable multi-choice question design. For the last task, we use colored arrows (red, green, yellow, blue) to encode robot actions.

interaction also varies significantly between rigid objects and deformable objects. To evaluate this, we collect image observations of multiple scenes with certain objects randomly placed in the scene. Given an image, the robot agent should reason about whether or not it is possible to achieve the desired fabric configuration (described via natural language) and if so, then how.

Additionally, understanding inverse dynamics is important when reasoning about feasible pick-and-place action(s) to perform goal-based tasks like fabric folding. For this dimension, we collect image observations of a fabric on a table and record the initial and final fabric configurations after performing a robot action like folding or unfolding. An agent should be able to generate the correct robot action from these images across diverse settings.

Similarly, it is essential for a robot agent to reason about the configuration of the fabric from its observation and about the consequences of an action if the scene were to slightly change. Moreover, it might not be trivial for a robot agent to demonstrate spatial reasoning capabilities, a foundation for numerous tasks [9]. For evaluating VLMs along these dimensions, we collect a single image observation per scene and manually label them with information required for further question generation. For instance, we might label the locations of different corners of a flat fabric on a table.

While our main focus is on emphasizing low-level reasoning capabilities, we also study some high-level reasoning. This refers to the ability to decompose complex tasks into smaller and/or more manageable sub-tasks. To this end, we consider certain aspects of high-level task planning and maintain a list of common sense statements (e.g., folding a fabric will be more effective if it is flattened first) which are used later to generate multiple-choice questions.

B.3 From Existing Simulation Environments

As discussed in Section 4, for each task category, we either reuse simulation environments from prior benchmarks (Fig. 6, first four images) or develop custom simulation environments (Fig. 6,

Dimension	Description
Task Planning Understanding	Assesses physical reasoning based on human intuition, requiring VLMs to select the correct answer.
Fabric State Understanding	Tests VLMs' ability to identify the correct fabric state from an image and four given options.
Spatial Reasoning Abilities	Evaluates VLMs' ability to locate fabric corners (e.g., "bottom-right") from an image.
Key-point Mapping Abilities	Assesses VLMs' accuracy in mapping key points from an image to a grid location.
Temporal Understanding of Action Sequence	Tests VLMs' ability to reorder shuffled images of a fabric manipulation sequence correctly.
Action Length Understanding	Evaluates VLMs' understanding of how short vs. long pick-place actions affect fabric configuration.
Inverse Dynamics Understanding	Requires VLMs to predict the correct pick-place action from four choices based on initial and final images.
Fabric-Solid Body Interaction Understanding	Tests VLMs' ability to choose the correct pick-place action when fabric interacts with solid objects.
Fabric-Fabric Interaction Understanding	Assesses VLMs' understanding of multi-fabric interactions, especially in bi-manual actions.
Counterfactual Understanding	Evaluates VLMs' reasoning on how changes in a scene or action alter outcomes.

Table 5: Description of the different dimensions for fabric manipulation, evaluated as a part of ManipBench. See Table 6 for the number of corresponding questions in ManipBench.

last image). Demonstrations are then generated using rollouts of pre-trained policies or via manual annotations, following a task-specific data collection procedure.

1. *Place Carrot (pick-and-place).* We use the simulation environment and pre-trained Octo policy [33] and RT-1 policy [85] from SimplerEnv to generate a set of robot demonstrations. We then identify keypoints representing the ground-truth pick-and-place actions using contact information provided by the simulator.
2. *Close Drawer (articulated object manipulation).* Similar to the Place Carrot task, we use SimplerEnv to generate robot demonstrations. For this task, we identify keypoints that represent the robot-drawer contact point and the robot movement direction following contact.
3. *Straighten Rope (deformable object manipulation).* We use SoftGym to simulate rope dynamics and generate demonstrations using a heuristic: the robot grippers pull two endpoints of the rope apart. In this task, keypoints represent the robot-rope contact point, as well as the robot movement direction following contact.
4. *Sweep Object (tool manipulation).* We use the environment from RLBench and pre-trained policy from PerAct [86] to generate robot demonstrations. The keypoints represent the tool-robot contact point, tool-object contact point, and the robot movement direction following contact.
5. *Ball Shooting (dynamic manipulation).* We create our own ball-shooting simulation environments in IsaacSim [75]. We manually annotate colored arrows on the images to denote different ball-shooting directions for the robot.

C Additional Details about Question Generation

Using the different data sources as described in Sec. 4, we generate **12617** multiple-choice questions. See Table 6 for the question statistics.

C.1 From Public Robotic Manipulation Datasets

We generate questions across 612 tasks as described in Section 5.2. Our preprocessing phase will sample 18 incorrect points for each episode and we only need 3 for building a question (with the 4th

Task Type	Total Questions
From Public Robotic Datasets	
Question Type 1	
DROID pick and place (p&p)	2010
DROID articulated (art.)	1640
Bridge	2470
Question Type 2	
DROID pick and place (p&p)	1005
DROID articulated (art.)	820
Bridge pick and place	1235
For Evaluating Fabric Manipulation	
Task Planning Understanding	240
Fabric State Understanding	234
Spatial Reasoning Abilities	325
Keypoint Mapping Abilities	312
Temporal Understanding of Action Sequence	240
Action Length Understanding	240
Inverse Dynamics Understanding	240
Fabric-Solid Body Interaction Understanding	282
Fabric-Fabric Interaction Understanding	280
Counterfactual Understanding	269
From Existing Simulation Environments	
Place Carrot (pick-and-place task)	277
Close Drawer (articulated manipulation task)	83
Straighten Rope (deformable manipulation)	140
Sweep Object (tool manipulation task)	194
Ball Shoot (dynamic manipulation task)	81
All Tasks Combined	12617

Table 6: Summary of tasks and the number of questions in each category. ManipBench has **12617** questions in all. See Sec. 4 and Sec. 5 for more details about the data sources and the question generation process, respectively.

Model Family	Params
OpenAI GPT [1] (closed-source)	
o1	N/A
GPT-4.1	N/A
GPT-4o	N/A
GPT-4o-mini	N/A
Google Gemini [2] (closed-source)	
Gemini-2.5-pro	N/A
Gemini-2.0-flash	N/A
Gemini-1.5-pro	N/A
Gemini-1.5-flash	N/A
GLM-4V [87, 88] (open-source)	
GLM-4V-9B	13.9B
InternVL-2 [89, 79] (open-source)	
InternVL-2-1B	0.94B
InternVL-2-2B	2.21B
InternVL-2-4B	4.15B
InternVL-2-8B	8.08B
InternVL-2-26B	25.50B
InternVL-2-40B	40.10B
InternVL-2-76B	76.30B
InternVL-2.5 [90] (open-source)	
InternVL-2.5-1B	0.94B
InternVL-2.5-2B	2.21B
InternVL-2.5-4B	3.71B
InternVL-2.5-8B	8.08B
InternVL-2.5-26B	25.50B
InternVL-2.5-38B	38.40B
InternVL-2.5-78B	78.40B
Qwen-VL [91] (open-source)	
Qwen-VL-Chat-Int4	4.05B
Qwen-VL-Chat	9.60B
Qwen2-VL [92] (open-source)	
Qwen2-VL-2B	2.21B
Qwen2-VL-7B	8.29B
Qwen2-VL-72B	73.40B
Qwen2.5-VL [93] (open-source)	
Qwen2.5-VL-3B	3.75B
Qwen2.5-VL-7B	8.29B
Qwen2.5-VL-32B	33.50B
Qwen2.5-VL-72B	73.40B
LLaVA-NeXT [94, 95, 96] (open-source)	
LLaVA-NeXT-7B	7.57B
Llama3.2-Vision [37, 97] (open-source)	
Llama3.2-11B-Vision-Instruct	10.60B

Table 7: Summary of model families evaluated.

point being the "correct" choice). Thus, to further exploit the dataset we have, we perform "question augmentation" by randomly sampling different sets of 3 incorrect points to build several question versions from the same episode. We also randomly sample 3 pairs of starting and ending tiles that are distinct from the ground-truth tiles as the incorrect options.

C.2 From In-house Fabric Manipulation Setup

We describe how we generate the MCQs for evaluating fabric manipulation in Sec. 5.3. On having all the data mapped in terms of affordances, we formulate different types of multiple-choice ques-

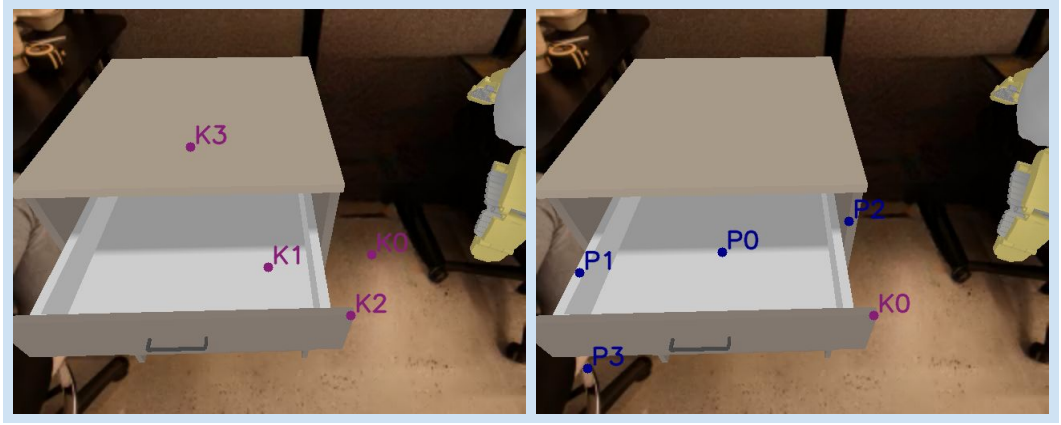


Figure 7: Two annotated images illustrating evaluation question generation for VLMs in the *Close Drawer* task. Left: Keypoints denote potential contact points for initiating the drawer-closing action. Right: Keypoints indicate possible movement directions for closing the drawer while maintaining contact at point K_0.

tions (see Table 5). We use the collected image observations to formulate the questions and have the obtained affordances act as the correct answers. The incorrect options for these questions are generated manually, making sure that they are clearly distinct from the answer. For instance, for the question types whose choices are based on the possible grid cells, we make sure that the cells neighboring the cell corresponding to the correct answer are not included in the possible choices.

C.3 From Existing Simulation Environments

After generating robot demonstrations from simulation (Sec. 4), we create evaluation questions for VLMs. In each demonstration, we identify the frames f_s and f_e where manipulation "starts" and "ends," respectively. We then annotate the candidate contact points and after-contact movements on the frames to generate evaluation questions. VLMs are required to select from the candidate keypoints to complete the given manipulation tasks. For instance, in Fig. 7, they must choose one contact point from `K0` to `K3` to initiate contact with the drawer and one point from `P0` to `P3` to push the drawer.

In our ball-shooting task (Fig. 6, right), instead of keypoint-based action annotations, we use colored arrows to encode robot shooting actions. VLMs must choose one arrow as the robot shooting direction. We designed three types of questions to evaluate the dynamic manipulation reasoning capabilities of VLMs in the ball-shooting task (see Sec. J.3.5).

D Additional Details about Human Evaluation

As described in Sec. 6, we perform a web-based human evaluation of our MCQs. We design the website using Python flask and host it using `ngrok` operated servers. We make the website easy for the volunteers to navigate through the questions, along with incorporating a feature that allows them to save their responses for later (see Fig. 8). We also provide some demo questions which will have to be correctly answered by the volunteers before we proceed with the test. We consider people with a robotics background from our research institution to volunteer for this evaluation. To reduce biases in the analysis, the person(s) responsible for designing the questions have not partaken in their evaluations. We plan to release this website in the future when ManipBench becomes public, which will also serve as a demo for our questions.

On top of the results described in Sec. 6, we also perform a small scale human evaluation for the questions created from simulation environments. Specifically, we ask human volunteers to answer 10 random questions for the *Place Carrot* and *Sweep Object* tasks, and observe an accuracy of 100% for both the cases. Due to the relatively smaller scale of these experiments, we do not include them as part of the discussion in the main paper.

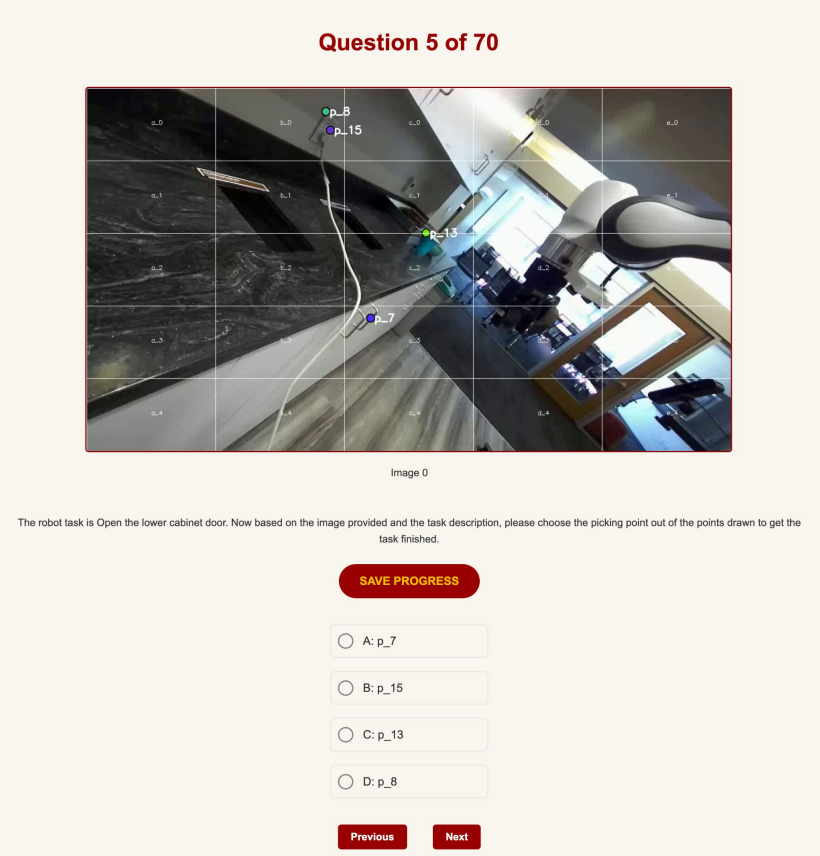


Figure 8: Visualization of the website used for human evaluation. The human selects one of the given choices.

E Additional Details about Real-World Experiments

We list the description of each task in the real-world experiments, which we report in Sec. 6.

- **Color Specific p&p**: Pick up the orange cube and place it in the bowl.
- **Spatial Specific p&p**: Pick up the cube to the right of the container and place it into the container.
- **Fit Apple**: Pick up the apple and place it in the cardboard box that best fits it.
- **Pick Crumpled Cloth**: Pick up the crumpled fabric and place it in the cardboard box.
- **Move Fabric Easy**: Move the fabric slightly to the right while keeping it flat. To do this, lift the rightmost corner and slightly drag it to the right.
- **Move Fabric Intermediate**: There are two pieces of fabric. Pick up the one covering the other without disturbing the second fabric, and move it slightly to the left.
- **Move Fabric Hard**: Move fabric slightly to the right while keeping it flat throughout the process.

For these experiments, we introduce an adaptive point sampling scheme that samples fewer points on smaller objects, minimizing overlap among the sampled points. Additionally, we adjust the number of points sampled per object based on the task requirements to simplify tasks for VLMs. For tasks that require VLMs to reason about selecting an object among multiple options, and if the exact picking point is not critical, we sample the center point of each object's mask. For the MCQs, we discretize the image into a 4x7 grid to match the rectangular dimensions of the robot workspace. The VLMs select one grid tile as the target location for gripper placement, and the specific placement point within the selected tile is randomly sampled following a truncated multivariate Gaussian distribution with a mean at the tile center. See Fig. 9 for the real-world experiment pipeline.

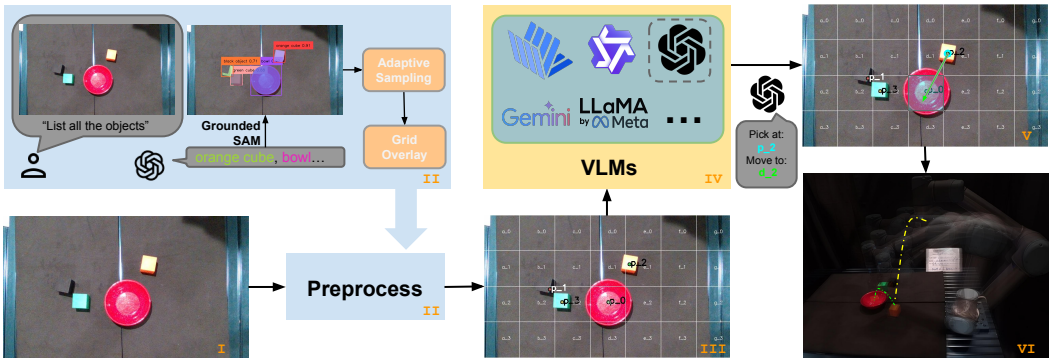


Figure 9: Real-world robot experiment pipeline. Given the task description and the top-down RGBD image (**I**), we first preprocess its RGB channels (**II**). We pass the raw RGB image to GPT and ask it to list all the objects, and then use GroundedSAM to get their masks for sampling candidate picking points. We draw the sampled points and a white grid overlay on the raw RGB image and obtain the processed image (**III**). For each task, we test different VLMs (**IV**). The chosen VLM outputs the picking point and an ending tile to complete the task. We sample a placing point inside the ending tile to ultimately get a pick-and-place action (**V**). After obtaining the pick-and-place action, and utilizing the depth information, we execute the action (**VI**).

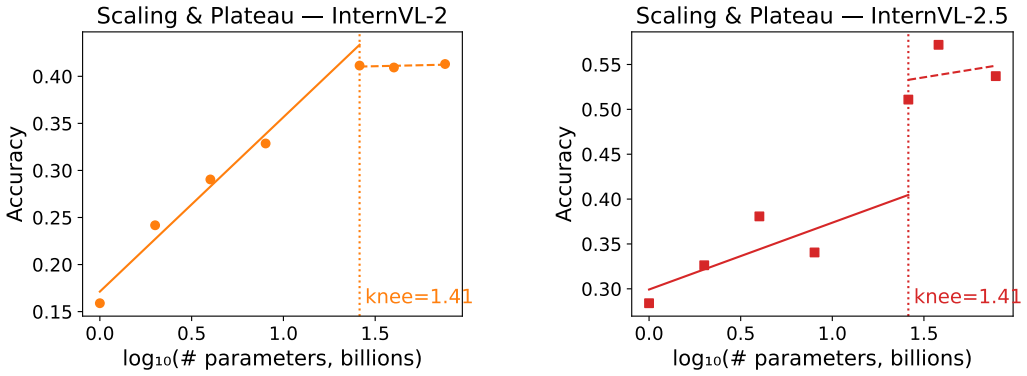


Figure 10: Log-scaling curves for the InternVL2 (left) and InternVL2.5 (right) families. Each panel plots overall model accuracy on ManipBench against $\log_{10}$(model size). A "knee" (vertical dotted line) splits the linear growth (solid fit) regime from the post-knee plateau (dashed fit). See App. I for more details.

F Additional Results from Public Robotic Manipulation Datasets

We provide the MCQ answering accuracy for the questions in Table 2 formed from the existing datasets. Each Q1 question is an MCQ with 4 options and each Q2 question has two MCQ sub-questions and is scored as correct if the VLM answers both of them correctly:

1. Picking Point Prediction: Select a pick point from the candidate picking points.
2. Ending Tile Prediction: Given the ground-truth pick point and the task description, select the destination tile.

We report the overall pick-place success rate as Q2 accuracy in Table 2. Due to the design of Q2, we also report the picking point prediction accuracies, ending tile prediction accuracies, and overall pick-place success rates from Q2 for a more detailed comparison of various VLMs. See Table 8. We use red to represent the highest accuracy, orange for the second-highest, and yellow for the third-highest. The same color code is used in the previous tables as well.

For the second sub-question of Q2 question, distractor tiles are sampled uniformly from all incorrect tiles, so duplicates can appear, and fewer than four options may be shown. Because the option set is random, the theoretical random performance for Q2 cannot be calculated; we instead estimate it empirically with a random policy and report it in Table 2 and Table 8.

Model	Bridge			DROID (art.)			DROID (p&p)		
	Picking Point Acc.	Ending Tile Acc.	Pick Place Success	Picking Point Acc.	Ending Tile Acc.	Pick Place Success	Picking Point Acc.	Ending Tile Acc.	Pick Place Success
Closed-Source									
o1	0.836	0.559	0.458	0.878	0.427	0.384	0.881	0.602	0.542
GPT-4.1	0.822	0.494	0.401	0.828	0.383	0.301	0.871	0.467	0.433
GPT-4o	0.785	0.395	0.309	0.843	0.479	0.396	0.861	0.448	0.401
GPT-4o-mini	0.546	0.330	0.183	0.670	0.243	0.156	0.658	0.313	0.210
Gemini-2.5-pro	0.923	0.431	0.403	0.876	0.407	0.362	0.903	0.493	0.453
Gemini-2.0-flash	0.883	0.326	0.287	0.868	0.448	0.387	0.821	0.279	0.235
Gemini-1.5-pro	0.797	0.309	0.237	0.840	0.139	0.126	0.779	0.179	0.158
Gemini-1.5-flash	0.458	0.347	0.287	0.818	0.149	0.123	0.741	0.246	0.193
Open-Source									
GLM-4V-9B	0.742	0.343	0.249	0.679	0.571	0.381	0.619	0.466	0.292
InternVL2-1B	0.249	0.135	0.028	0.262	0.428	0.105	0.261	0.254	0.066
InternVL2-2B	0.327	0.151	0.052	0.361	0.363	0.120	0.291	0.201	0.052
InternVL2-4B	0.445	0.317	0.134	0.481	0.576	0.357	0.549	0.434	0.242
InternVL2-8B	0.645	0.403	0.258	0.642	0.357	0.228	0.651	0.390	0.258
InternVL2-26B	0.725	0.423	0.326	0.765	0.379	0.293	0.759	0.397	0.296
InternVL2-40B	0.756	0.446	0.338	0.778	0.358	0.272	0.751	0.412	0.313
InternVL2-76B	0.879	0.588	0.509	0.849	0.279	0.235	0.820	0.477	0.411
InternVL2.5-1B	0.347	0.035	0.016	0.379	0.252	0.099	0.387	0.165	0.057
InternVL2.5-2B	0.475	0.045	0.026	0.457	0.182	0.096	0.461	0.056	0.027
InternVL2.5-4B	0.681	0.290	0.199	0.650	0.280	0.181	0.549	0.434	0.242
InternVL2.5-8B	0.713	0.462	0.330	0.687	0.307	0.229	0.728	0.355	0.272
InternVL2.5-26B	0.884	0.423	0.376	0.817	0.527	0.433	0.838	0.458	0.389
InternVL2.5-38B	0.875	0.606	0.528	0.838	0.373	0.316	0.823	0.503	0.425
InternVL2.5-78B	0.876	0.624	0.541	0.823	0.418	0.348	0.840	0.500	0.431
QwenVL-Chat	0.291	0.265	0.067	0.309	0.326	0.141	0.259	0.302	0.077
Qwen2VL-2B	0.310	0.171	0.045	0.404	0.411	0.156	0.372	0.216	0.089
Qwen2VL-7B	0.591	0.329	0.191	0.611	0.593	0.366	0.650	0.504	0.329
Qwen2VL-72B	0.802	0.505	0.398	0.826	0.473	0.395	0.845	0.527	0.460
Qwen2.5-VL-3B	0.469	0.414	0.190	0.527	0.265	0.144	0.657	0.298	0.197
Qwen2.5-VL-7B	0.684	0.500	0.344	0.712	0.517	0.372	0.774	0.508	0.408
Qwen2.5-VL-32B	0.795	0.540	0.428	0.782	0.484	0.399	0.788	0.555	0.459
Qwen2.5-VL-72B	0.811	0.595	0.470	0.809	0.462	0.390	0.830	0.561	0.481
LLaVA-NeXT-7B	0.258	0.243	0.060	0.245	0.359	0.094	0.300	0.305	0.100
Llama3.2-11B-VI	0.388	0.298	0.118	0.429	0.276	0.115	0.413	0.270	0.111
Random	0.247	0.279	0.061	0.257	0.287	0.063	0.250	0.287	0.084

Table 8: Performance comparison of various VLMs on our MCQs for the Bridge and DROID datasets on Question Type 2 (Q2). We report the Pick Place Success as the Q2 accuracy in Table 2. Details of Q2 questions can be found in Section F.

Error Mode Analysis.

From Table 8, the Ending Tile Accuracy is significantly lower than the Picking Point Accuracy. Our argument is that placing the object at the target location requires a more advanced and comprehensive reasoning ability of manipulation, which includes spatial reasoning, object identification, and even implicit understanding, whereas choosing the correct picking point in the questions from public robotic manipulation datasets relies more on object identification.

Question Length Analysis.

Our question/answer pairs are very long and they require the model to associate between the text and the image many times. To investigate, we create a small sample of 100 binary multiple choice questions to simplify the stated text-image association. These questions are designed from the Bridge [69] dataset and have the images annotated with two differently colored points (red and blue), with one of the points being a correct answer for the question under consideration.

Each question has two sub-questions with different color palettes for the points. In sub-question 1, the red point is the correct picking point; In sub-question 2, the blue point is the correct picking point. The position of those 2 candidate picking points are identical on both sub-questions. This is done to mitigate possible color bias exhibited by the VLMs while answering these questions. See Fig. 11 for a sample question, and Table 9 for the results. We see that the results with these questions are consistent with the results stated in Section 7. For example, the best VLM is still Gemini-2.5-pro

Figure 11: Sample question for performing question length analysis. The task here is "Put the banana on the green cloth," and the correct choice would be the red point.

with the overall accuracy of 0.98, while the smaller open-source VLMs demonstrate color bias and thus have relatively poor performance overall.

We also include models of the PaliGemma family [98, 99, 100] in this smaller experimental study. Specifically, the original PaliGemma model with 3 billion parameters (Paligemma-V1-3B), the PaliGemma V2 model with 3 billion parameters (Paligemma-V2-3B), and the Gemma 3 model with 4 billion parameters (Gemma-3-4B). We observe that Gemma 3 performs similar to open-source models of similar size, whereas PaliGemma V2 refuses to answer the questions, resulting in a 0.00 accuracy. Since these models are not evaluated on any other questions in ManipBench, we do not include them in Table 7.

G Additional Results from the In-house Fabric Manipulation Setup

We provide comparison for the MCQ answering accuracies of selected VLMs in Figure 4 for the questions formed from the in-house fabric manipulation setup. The MCQ answering accuracy for all the models can be found in Table 10. The models of GLM-4V-9B and QwenVL-Chat do not support multi-image reasoning hence we do not include any accuracies for these models on the tasks of *Temporal Sequence Understanding* and *Inverse Dynamics Understanding* as they require reasoning with two or more images.

H Additional Results from the Statistical Analysis

As discussed in Section 7.1, we compute the Pearson's and the Spearman's coefficients to quantify the correlation between the performance of VLMs on the questions of ManipBench and their effectiveness as real-world robotic agents. See Table 11. On performing the analysis by aggregating over the task categories, we observe that the questions from for evaluating Fabric Manipulation have the strongest correlation with the real-world success rates with the Pearson's coefficient of 0.950 ($p = 0.001$), Spearman's coefficient of 0.986 ($p = 0.001$), and Kendall's Tau of 0.9090 ($p = 0.002$). These values indicate an extremely strong linear relationship that maintains rank-order consistency. The questions from existing simulation data demonstrate a relatively weaker correlation, with the

Model	Accuracy Red Correct	Accuracy Blue Correct	Accuracy Both Correct
Closed-Source			
Gemini-2.5-pro	1.00	0.98	0.98
GPT-4o	0.60	0.76	0.50
Open-Source			
InternVL2-1B	0.34	0.20	0.08
InternVL2-2B	1.00	0.00	0.00
InternVL2-4B	0.12	1.00	0.12
InternVL2-8B	0.22	0.82	0.12
InternVL2-26B	0.56	0.66	0.26
InternVL2.5-1B	0.92	0.00	0.00
InternVL2.5-2B	1.00	0.12	0.12
InternVL2.5-4B	0.10	0.98	0.10
InternVL2.5-8B	0.48	0.74	0.28
InternVL2.5-26B	0.72	0.76	0.48
Qwen2VL-2B	1.00	0.12	0.12
Qwen2VL-7B	0.98	0.58	0.56
Qwen2.5-VL-3B	0.31	0.82	0.22
Qwen2.5-VL-7B	0.51	0.94	0.49
Paligemma-V1-3B	0.76	0.36	0.26
Paligemma-V2-3B	0.00	0.00	0.00
Gemma-3-4B	0.88	0.37	0.32
Random	0.50	0.50	0.25

Table 9: Performance comparison of various VLMs on the simpler binary MCQs, which comprise of red and blue points annotated on the image with one of them being the correct answer for the task in hand. We report the question answering accuracies for the case where the red point is the correct answer, and where the blue point is the correct answer in the first two columns respectively. We also take an intersection of these and report the accuracies of solving both the sub-questions correctly in the third column.

Pearson's coefficient being 0.779 ($p = 0.023$), Spearman's coefficient being 0.638 ($p = 0.009$), and Kendall's Tau being 0.691 ($p = 0.018$). The reason behind this could be attributed to the weaker correlations observed for the Sweep Object task as shown in Table 11.

I Scaling Laws Analysis

Scaling laws [101] describe how a model's performance rises as more compute, data, or parameters are added. Quantifying these relationships lets practitioners decide whether a large model is worth the extra cost and guides researchers toward regimes where architectural innovation, rather than brute scale, is likely to deliver the next accuracy jump.

We studied the scaling behaviors of 4 open-source model families: InternVL2, InternVL2.5, Qwen2VL, and Qwen2.5-VL. By computing the Pearson correlation coefficient (r), which measures how closely accuracy increases follow a straight-line relationship with $\log_{10}(\text{model size})$ we find strong scaling in every family: InternVL-2 (r=0.969), InternVL-2.5 (0.937), Qwen-VL-2 (0.998), Qwen-VL-2.5 (0.890).

Furthermore, we also performed a local, post-knee analysis on the two 2 InternVL families (Fig. 10) to study their scaling plateauing behaviors. Using a knee-finder that keeps at least three checkpoints on each side of the split, we fitted separate lines before and after the knee. Focusing only on the few checkpoints that lie just beyond each knee, we find that the growth slope nearly vanishes.

J Sample Multiple Choice Question Examples

This section contains one sample question for each question type in ManipBench. We include the questions with the exact text instruction given as prompt to the VLMs during the evaluation. The figure captions are self-explanatory.

Model	High Level Planning	Fabric State	Spatial Reasoning	Keypoint Mapping	Temporal Sequence	Action Length	Inverse Dynamics	Fabric-Solid Interaction	Fabric-Fabric Interaction	Counter Factual
Closed-Source										
o1	0.879	0.713	0.886	0.855	0.812	0.650	0.487	0.712	0.625	0.799
GPT-4.1	0.825	0.589	0.587	0.551	0.745	0.512	0.291	0.347	0.282	0.565
GPT-4o	0.770	0.624	0.606	0.558	0.829	0.754	0.395	0.347	0.243	0.632
GPT-4o-mini	0.662	0.436	0.409	0.336	0.537	0.387	0.358	0.184	0.171	0.469
Gemini-2.5-pro	0.887	0.726	0.975	0.990	0.962	0.487	0.775	0.762	0.616	0.820
Gemini-2.0-flash	0.770	0.556	0.673	0.750	0.858	0.533	0.458	0.298	0.336	0.544
Gemini-1.5-pro	0.712	0.449	0.452	0.445	0.725	0.475	0.283	0.230	0.296	0.527
Gemini-1.5-flash	0.675	0.448	0.378	0.461	0.625	0.475	0.208	0.220	0.182	0.435
Open-Source										
GLM-4V-9B	0.496	0.423	0.541	0.711	X	0.216	X	0.305	0.146	0.422
InternVL2-1B	0.312	0.167	0.202	0.313	0.222	0.333	0.083	0.327	0.151	0.192
InternVL2-2B	0.475	0.372	0.452	0.426	0.225	0.221	0.358	0.326	0.150	0.293
InternVL2-4B	0.604	0.393	0.534	0.641	0.241	0.279	0.383	0.184	0.157	0.356
InternVL2-8B	0.612	0.406	0.455	0.448	0.354	0.266	0.446	0.262	0.186	0.414
InternVL2-26B	0.562	0.389	0.612	0.570	0.259	0.254	0.404	0.273	0.214	0.343
InternVL2-40B	0.645	0.479	0.566	0.539	0.296	0.308	0.470	0.234	0.132	0.402
InternVL2-76B	0.762	0.594	0.748	0.724	0.500	0.512	0.487	0.177	0.243	0.364
InternVL2.5-1B	0.346	0.252	0.295	0.295	0.304	0.233	0.316	0.245	0.232	0.259
InternVL2.5-2B	0.446	0.359	0.517	0.500	0.150	0.221	0.437	0.372	0.100	0.364
InternVL2.5-4B	0.608	0.350	0.575	0.708	0.279	0.304	0.400	0.549	0.168	0.368
InternVL2.5-8B	0.658	0.402	0.504	0.570	0.433	0.358	0.471	0.316	0.182	0.473
InternVL2.5-26B	0.671	0.589	0.628	0.574	0.658	0.354	0.487	0.284	0.296	0.439
InternVL2.5-38B	0.758	0.529	0.772	0.721	0.733	0.400	0.548	0.213	0.282	0.594
InternVL2.5-78B	0.800	0.487	0.741	0.753	0.808	0.687	0.512	0.344	0.332	0.586
QwenVL-Chat	0.316	0.295	0.311	0.311	X	0.187	X	0.138	0.068	0.209
Qwen2VL-7B	0.616	0.389	0.372	0.340	0.216	0.604	0.358	0.255	0.114	0.376
Qwen2.5-VL-3B	0.562	0.364	0.320	0.317	0.262	0.204	0.383	0.294	0.211	0.343
Qwen2.5-VL-7B	0.637	0.389	0.401	0.439	0.262	0.400	0.408	0.397	0.157	0.322
Random	0.237	0.226	0.240	0.275	0.246	0.250	0.267	0.269	0.246	0.280
Human	0.852	0.960	0.980	1.000	1.000	1.000	0.920	0.910	0.940	0.742

Table 10: Performance comparison of various VLMs on our MCQs for evaluating fabric manipulation. We report accuracies for the dimensions discussed in Section B.2. For tasks requiring multi-image reasoning, the performance of models that do not support it is denoted by X in the table.

J.1 From Public Robotic Manipulation Datasets

J.1.1 Type 1 (Q1)

The left-most question in Fig. 1 is an example question in this category. Another sample question (with the exact VLM prompts) can be found in Fig. 12. This question corresponds to the DROID articulated (art.) task.

J.1.2 Type 2 (Q2)

A sample question with the exact VLM prompts for this question type is in Fig. 13. This question corresponds to the DROID pick and place (p&p) task.

J.2 From In-house Fabric Manipulation Setup

J.2.1 Task Planning Understanding

See Fig. 14.

J.2.2 Fabric State Understanding

See Fig. 15.

J.2.3 Spatial Reasoning Abilities

See Fig. 16.

Task Type	Pearson		Spearman		Kendall's Tau	
	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value
From Public Robotic Datasets						
DROID pick and place (Q1)	0.887	0.003	0.862	0.006	0.691	0.018
DROID articulated (Q1)	0.905	0.002	0.898	0.002	0.764	0.009
Bridge (Q1)	0.628	0.095	0.683	0.062	0.618	0.034
DROID pick and place (Q2)	0.811	0.015	0.874	0.005	0.691	0.018
DROID articulated (Q2)	0.426	0.293	0.323	0.435	0.182	0.533
Bridge (Q2)	0.605	0.112	0.657	0.077	0.519	0.079
For Evaluating Fabric Manipulation						
Task Planning Understanding	0.922	0.001	0.934	0.001	0.837	0.004
Fabric State Understanding	0.905	0.002	0.934	0.001	0.837	0.004
Spatial Reasoning Abilities	0.868	0.005	0.850	0.007	0.691	0.018
Keypoint Mapping Abilities	0.655	0.078	0.575	0.136	0.473	0.105
Temporal Understanding of Action Sequence	0.886	0.003	0.934	0.001	0.837	0.004
Action Length Understanding	0.729	0.040	0.707	0.050	0.473	0.105
Inverse Dynamics Understanding	0.767	0.026	0.719	0.045	0.618	0.034
Fabric-Solid Body Interaction Understanding	0.788	0.020	0.850	0.007	0.764	0.009
Fabric-Fabric Interaction Understanding	0.808	0.015	0.801	0.017	0.667	0.024
Counterfactual Understanding	0.908	0.002	0.922	0.001	0.837	0.004
From Existing Simulation Environments						
Place Carrot (pick and place task)	0.679	0.064	0.719	0.045	0.473	0.105
Close Drawer (articulated manipulation task)	0.862	0.006	0.807	0.015	0.667	0.024
Straighten Rope (deformable manipulation task)	0.710	0.048	0.743	0.035	0.473	0.105
Sweep Object (tool manipulation task)	0.270	0.518	0.204	0.629	0.255	0.383
Ball Shoot (dynamic manipulation task)	0.815	0.014	0.814	0.014	0.764	0.009

Table 11: Pearson's, Spearman's, and Kendall's Tau coefficients, along with their corresponding p-values, computed with respect to the experiments described in Section 7.1.

J.2.4 Key-point Mapping Abilities

See Fig. 17.

J.2.5 Temporal Understanding of Action Sequence

See Fig. 18.

J.2.6 Action Length Understanding

See Fig. 19.

J.2.7 Inverse Dynamics Understanding

See Fig. 20.

J.2.8 Fabric-Solid Body Interaction Understanding

See Fig. 21.

J.2.9 Fabric-Fabric Interaction Understanding

See Fig. 22.

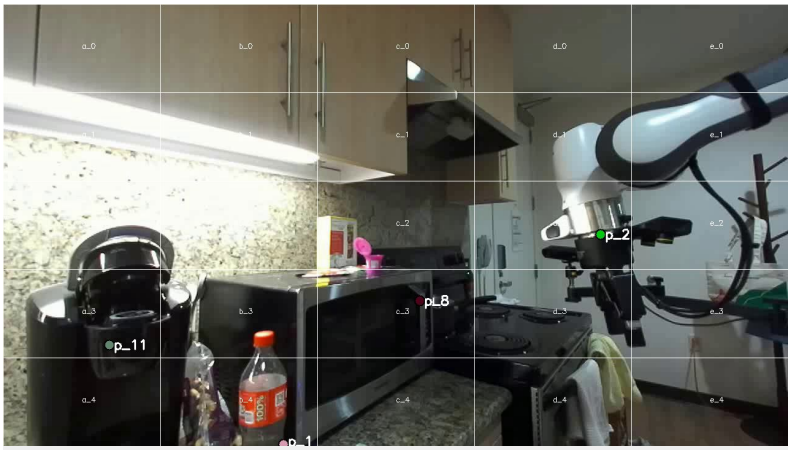
J.2.10 Counter Factual Understanding

See Fig. 23.

J.3 From Existing Simulation Environments

J.3.1 Place Carrot

See Fig. 24.



Prompt: The robot task is Open the microwave. Now based on the image provided and the task description, please choose the correct choice from the following 4 choices that has the correct picking point and the trajectory to get the task finished.

Option A: Picking point: p_8, Trajectory of the gripper: c_3,d_3
Option B: Picking point: p_11, Trajectory of the gripper: a_3,b_2
Option C: Picking point: p_1, Trajectory of the gripper: b_4,e_4
Option D: Picking point: p_2, Trajectory of the gripper: d_2,b_3

Figure 12: Sample question for *Type 1 (Q1)* from Existing Robotic Manipulation Datasets. Answer: Option A.

J.3.2 Close Drawer

See Fig. 25.

J.3.3 Straighten Rope

See Fig. 26.

J.3.4 Sweep Object

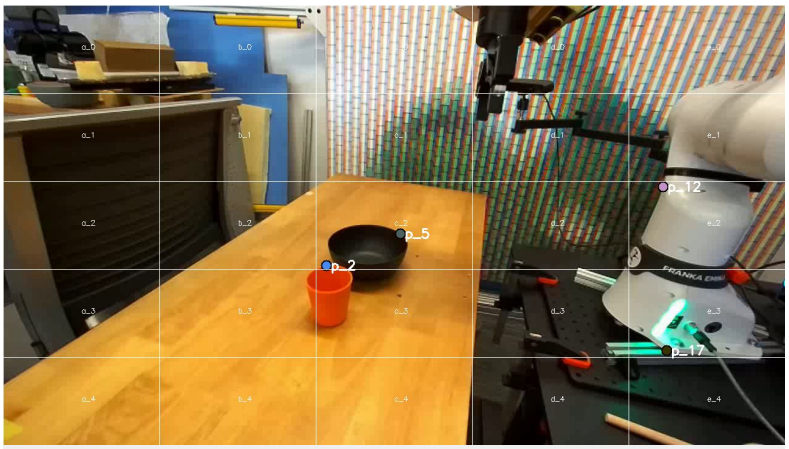
See Fig. 27.

J.3.5 Ball Shooting

Type 1. See Fig. 28.

Type 2. See Fig. 29.

Type 3. See Fig. 30.



Prompt for the first question: The robot task is Pick up the orange cup and put it in the black bowl. Now based on the image provided and the task description, please choose the picking point out of the points drawn to get the task finished.
Option A: p_17; Option B: p_2; Option C: p_12; Option D: p_5

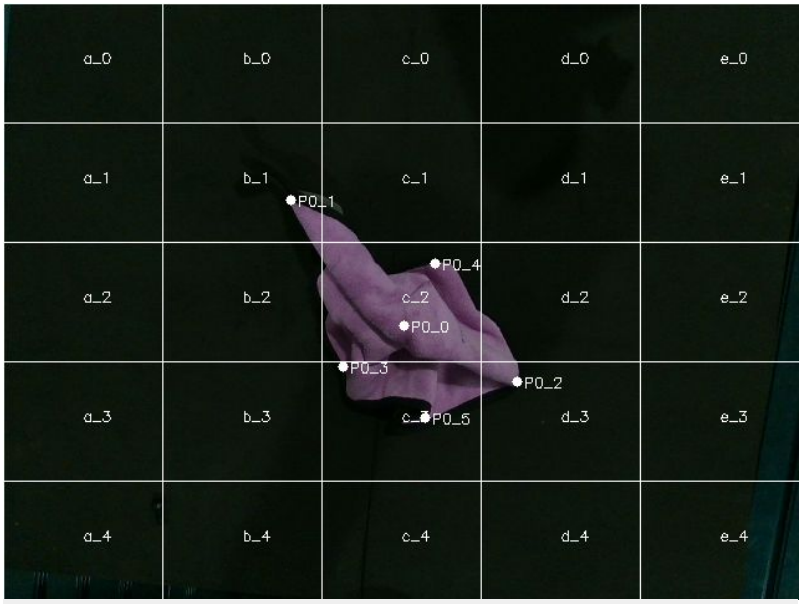
Prompt for the second question: The robot task is Pick up the orange cup and put it in the black bowl. Now based on the image provided and the task description, suppose the robot grasped at p_2, please choose the image tile the gripper should move to for finishing the task.
Option A: d_4; Option B: c_2; Option C: e_3; Option D: e_2

Figure 13: Question for *Type 2 (Q2)* from Existing Robotic Manipulation Datasets. The answer to the first question is B and the answer to the second question is B. The VLM has to answer both questions correctly.

Prompt: Which of the following actions will probably not result in a fabric fold? Select one of the four options below
Option A: Aligning two adjacent corners
Option B: Moving a corner to meet the center
Option C: Aligning adjacent corners to meet near the center
Option D: Lifting and dropping the cloth

Return your output in the below format only:
- Answer: One of the four choices - A, B, C, D
- Explanation: A justification for your choice by evaluating all the options thoroughly

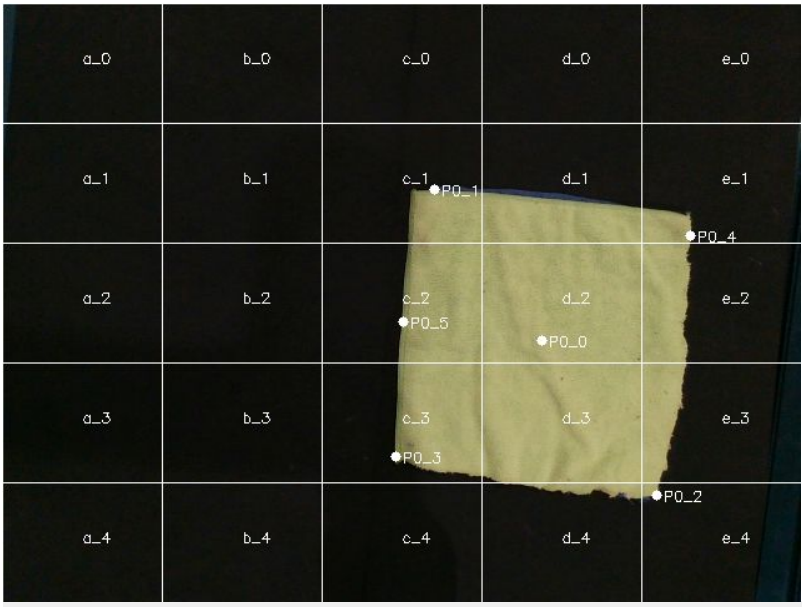
Figure 14: Question for *Task Planning Understanding*. Answer: Option D.



Prompt: I shall be providing you with an image of a fabric lying on the table in any possible configuration.
This image is divided into 25 grid cells with the labels of a_0, b_0,...,d_4, e_4. There are also some key-points annotated on the fabric with the corresponding label.
Given the fabric image and four options A to D representing the possible descriptions of the fabric configuration, your task is to choose one of these options which would be describing the fabric configuration correctly.
Option A: The fabric is folded in a diagonal manner
Option B: The fabric on the table is highly crumpled
Option C: More than one corners of the fabric are folded inward
Option D: The fabric is lying flat on the table

Return your output in the below format only:
- Answer: One of the four choices - A, B, C, D
- Explanation: A justification for your choice by evaluating all the options thoroughly

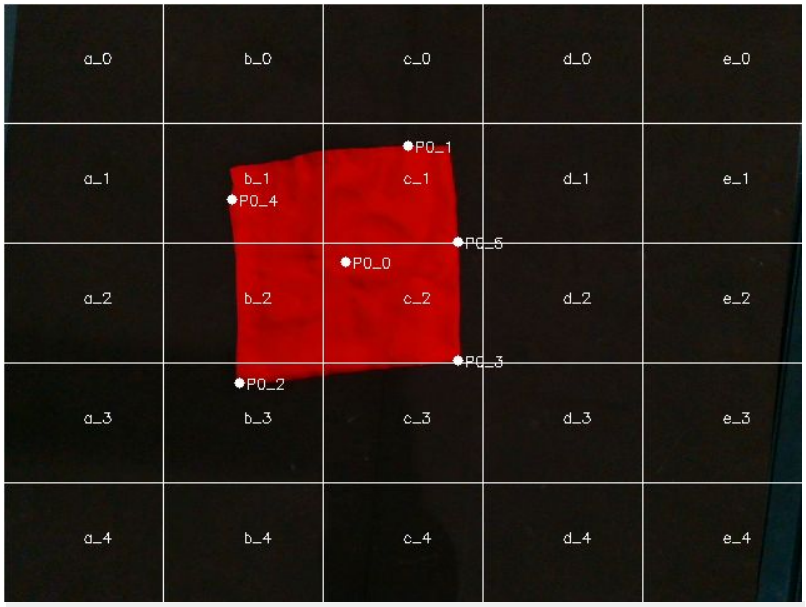
Figure 15: Question for *Fabric State Understanding*. Answer: Option B.



Prompt: I shall be providing you with an image of a fabric lying flat on the table. This image is divided into 25 grid cells with the labels of a_0, b_0,...,d_4, e_4. There are also some key-points annotated on the fabric with the corresponding label.
Give the fabric image and four options A to D corresponding to a different grid cell location, choose the correct option representing the fabric top-right corner
Option A: e_4
Option B: c_1
Option C: e_1
Option D: a_3

Return your output in the below format only:
- Answer: One of the four choices - A, B, C, D
- Explanation: A justification for your choice by evaluating all the options thoroughly

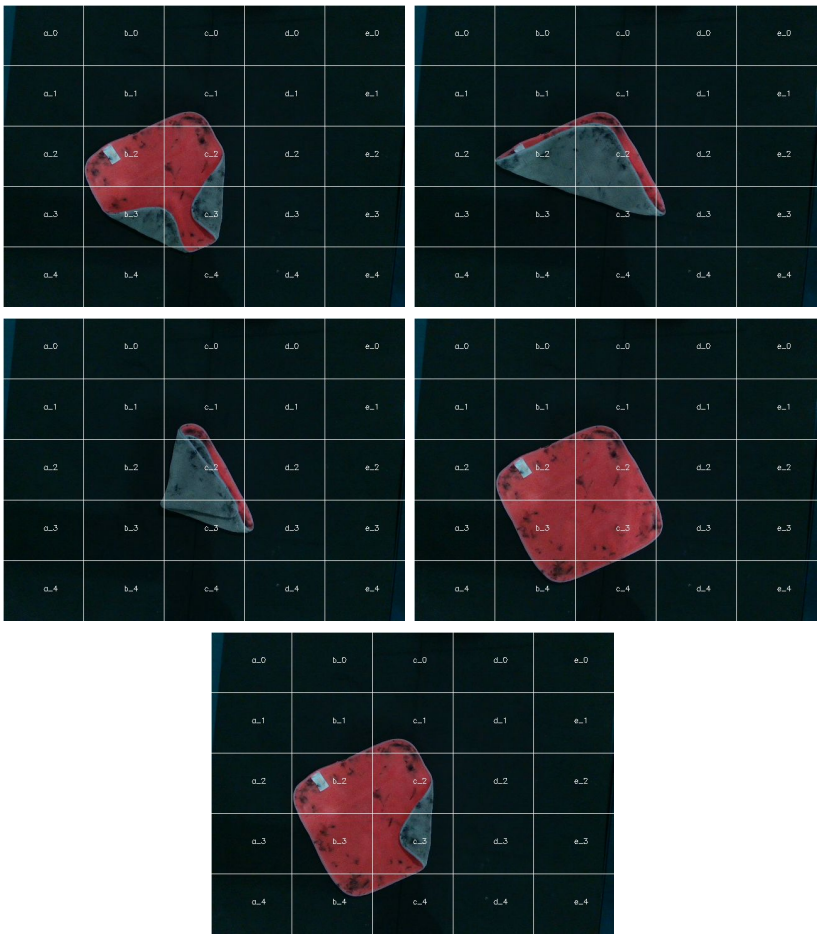
Figure 16: Question for *Spatial Reasoning Abilities*. Answer: Option C.



Prompt: I shall be providing you with an image of a fabric lying flat on the table. This image is divided into 25 grid cells with the labels of a_0, b_0,...,d_4, e_4. There are also some key-points annotated on the fabric with the corresponding label.
Give the fabric image and four options A to D corresponding to a different grid cell location, choose the correct choice for the grid cell where the keypoint P0_4 is located.
Option A: b_1
Option B: b_3
Option C: d_4
Option D: c_2

Return your output in the below format only:
- Answer: One of the four choices - A, B, C, D
- Explanation: A justification for your choice by evaluating all the options thoroughly

Figure 17: Question for *Key-point Mapping Abilities*. Answer: Option A.



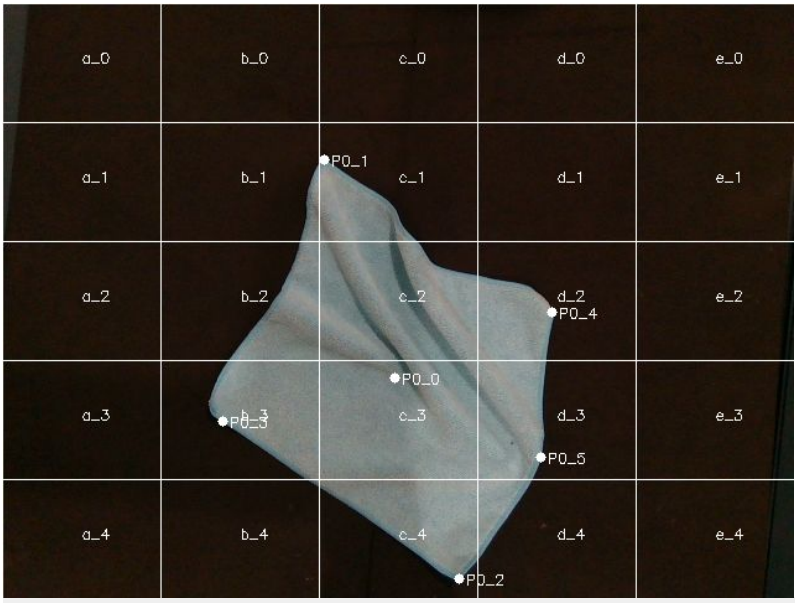
Prompt: I shall be providing you with multiple images of a fabric on a table. These images correspond to different configurations of the fabric. Each image is divided into 25 grid cells with the labels of a_0, b_0,...,d_4, e_4. These images are labeled as 0, 1, 2, and so on.
I shall be providing you with a task description and your role is to reorder these images in a way that a single pick-and-place action can lead to the transition between one image and the next. A pick-and-place action is essentially one robot arm picking a fabric point located in one of those 25 grid cells, lifting the fabric by a small height, moving to a placing location among one of those 25 grid cells, and then finally lowering and releasing its grip. It is important that the transition across all pairs of the consecutive images in the reordered list makes sense according to the task description, resulting in a final fabric configuration (i.e the last image) as specified in the task description.

You are also provided with the description of the fabric state in each of the images as:
Image 0: Two corners of the fabric are folded; Image 1: The fabric appears to have been diagonally folded; Image 2: The fabric appears to have been diagonally folded; Image 3: The fabric is lying flat; Image 4: One corner of the fabric is folded

Given four options A to D, representing a reordering of the images, your task is to pick the correct choice that is consistent with the requirements stated above. Task description: Flatten the fabric by unfolding the corners one at a time, followed by folding the resulting flat fabric repeatedly in a way that you always align the farthest corners to one another
Option A: 0, 2, 3, 1, 4; Option B: 0, 3, 2, 1, 4; Option C: 0, 4, 3, 1, 2; Option D: 0, 2, 1, 3, 4

Return your output in the below format only:
- Answer: One of the four choices - A, B, C, D
- Explanation: A justification for your choice by evaluating all the options thoroughly

Figure 18: Sample Question for *Temporal Understanding of Action Sequence*. Answer: Option C.



Prompt: I shall be providing you with an image of a fabric lying on the table that is either slightly crumpled or has one corner that is folded inward slightly. This image is divided into 25 grid cells with the labels of a_0, b_0,...,d_4, e_4. There are also some key-points annotated on the fabric with the corresponding label.

A robot arm would pick one of those fabric keypoints, lift the fabric by a small height, move to a placing location, and then finally lower and release its grip. This describes a pick-and-place action.

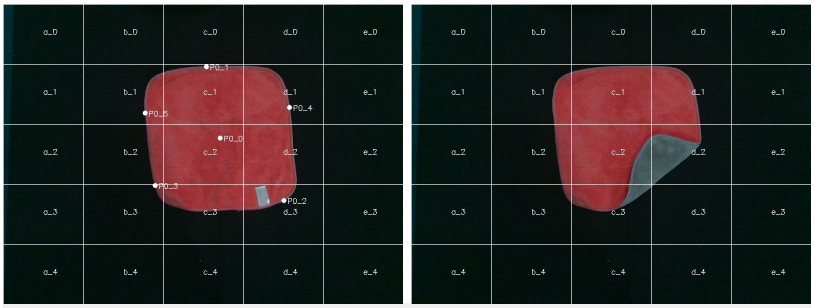
For the purpose of this question, we consider the placing location to be any one of the 25 grid cells in the scene. The given fabric configuration needs only a slight adjustment to flatten it via a pick-and-place action. Note that adjusting the fabric more by picking and placing it over longer distances might add more wrinkles to the fabric.

Given a fabric keypoint to be picked and four options A to D representing a placing location grid cell, your task is to choose one of these options such that the resulting pick-and-place action flattens the fabric without adding more wrinkles.

Given the fabric key-point P0_4 located in the grid d_2, choose a placing location among
Option A: e_0; Option B: d_2; Option C: e_4; Option D: c_0

Return your output in the below format only:
- Answer: One of the four choices - A, B, C, D
- Explanation: A justification for your choice by evaluating all the options thoroughly

Figure 19: Question for *Action Length Understanding*. Answer: Option B.



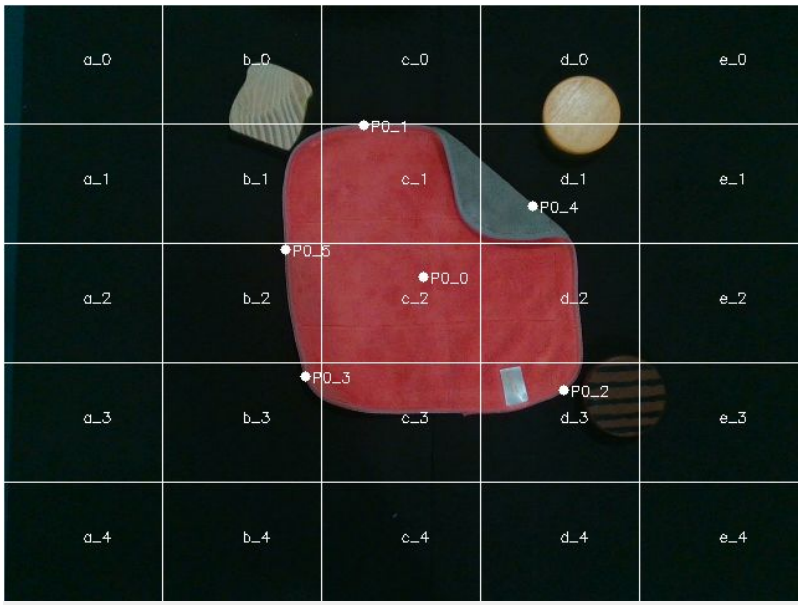
Prompt: I shall be providing you with two images of a fabric on a table. The first image corresponds to the initial configuration of the fabric and the second image corresponds to the final configuration of the fabric. Both the images are divided into 25 grid cells with the labels of a_0, b_0,...,d_4, e_4.
There are also some key-points annotated on the fabric with the corresponding label in the first image, which corresponds to the initial configuration. Note that there are no key-points annotated in the second image since they are not needed for this task.

The transition from the initial configuration to the final configuration is achieved by a single pick-and-place action. Essentially, a robot arm would pick a fabric point located in one of those 25 grid cells, lift the fabric by a small height, move to a placing location among one of those 25 grid cells, and then finally lower and release its grip. This describes a pick-and-place action. We want to help the robot to achieve this transition by providing the correct picking or placing point information that form the correct pick-and-place action.
Given the fabric key point P0_2 located in the grid d_3 being the picking point, choose the correct grid cell location for the place point among

Option A: d_4; Option B: a_2; Option C: a_3; Option D: d_2

Return your output in the below format only:
- Answer: One of the four choices - A, B, C, D
- Explanation: A justification for your choice by evaluating all the options thoroughly

Figure 20: Question for *Inverse Dynamics Understanding*. Answer: Option D.



Prompt: I shall be providing you with an image of a fabric lying on the table with certain objects on or around it. This image is divided into 25 grid cells with the labels of a_0, b_0,...,d_4, e_4. There are also some key-points annotated on the fabric with the corresponding label.

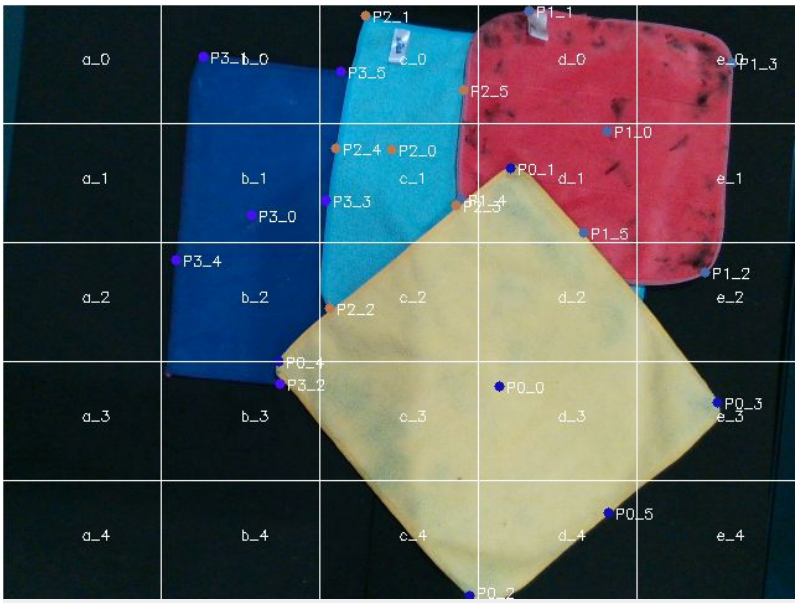
A robot arm would pick one of those fabric keypoints, lift the fabric by a small height, move to a placing location, and then finally lower and release its grip. This describes a pick-and-place action. For the purpose of this question, we consider the placing location to be any one of the 25 grid cells in the scene.

Given a fabric keypoint to be picked and four options A to D representing a placing location grid cell, your task is to choose one of these options such that the resulting pick-and-place action DOES NOT displace or cover any of the objects in the scene. Given the fabric key-point P0_4 located in the grid d_1, choose a placing location among

Option A: c_2
Option B: e_0
Option C: e_1
Option D: e_3

Return your output in the below format only:
- Answer: One of the four choices - A, B, C, D
- Explanation: A justification for your choice by evaluating all the options thoroughly

Figure 21: Sample Question for *Fabric-Solid Body Interaction Understanding*. Answer: Option A.



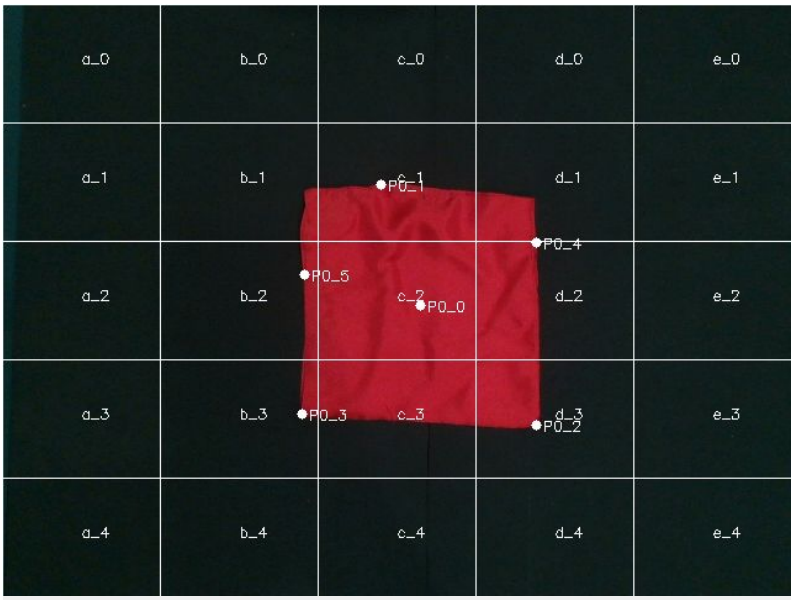
Prompt: I shall be providing you with an image of different fabrics lying flat on a table, possibly on top of one another. This image is divided into 25 grid cells with the labels of a_0, b_0,...,d_4, e_4. There are also some key-points annotated on the fabric with the corresponding label.

There are two robot arms who would each grab one of those points and lift the grasped points vertically upwards simultaneously. Given four options A to D, each representing a pair of points lifted vertically by the robot arms, your task is to choose the correct pair of points such that the given motion will displace ONLY one fabric out of all the fabrics present in the scene.

Option A: Point P3_1 in grid cell b_0 and Point P3_4 in grid cell b_2
Option B: Point P1_1 in grid cell d_0 and Point P1_0 in grid cell d_1
Option C: Point P2_1 in grid cell c_0 and Point P2_4 in grid cell c_1
Option D: Point P0_5 in grid cell d_4 and Point P0_3 in grid cell e_3

Return your output in the below format only:
- Answer: One of the four choices - A, B, C, D
- Explanation: A justification for your choice by evaluating all the options thoroughly

Figure 22: Sample Question for *Fabric-Fabric Interaction Understanding*. Answer: Option D.



Prompt: I shall be providing you with an image of a flat fabric lying on the table. This image is divided into 25 grid cells with the labels of a_0, b_0,...,d_4, e_4. There are also some key-points annotated on the fabric with the corresponding label.

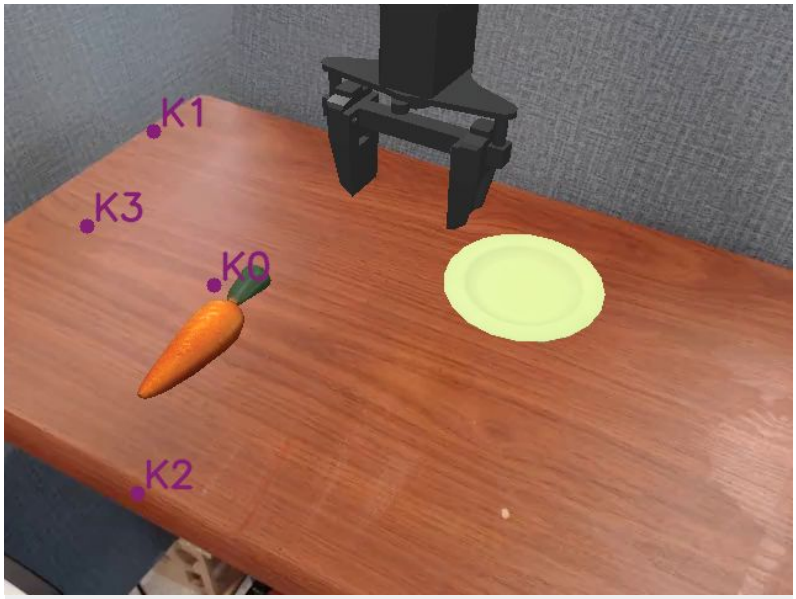
A robot arm would pick one of those fabric keypoints, lift the fabric by some height, move to a placing location, and then finally lower and release its grip. This describes a pick-and-place action. For the purpose of this question, we consider the placing location to be any one of the 25 grid cells in the scene.

Given this, if we pick the fabric point , located in the grid cell d_1, and place it at a random point in the grid cell b_3, then this action results in the fabric getting folded.
We now tweak the scene by placing a small solid object in the grid cell a_4. Which of the following is most likely to happen with the robot action specified previously?

Option A: The object is covered by the robot action
Option B: The fabric stays flat and will be dragged instead by the robot action
Option C: The object is not impacted by the robot action
Option D: The object is displaced by the robot action

Return your output in the below format only:
- Answer: One of the four choices - A, B, C, D
- Explanation: A justification for your choice by evaluating all the options thoroughly

Figure 23: Question for *Counter Factual Understanding*. Answer: Option C.

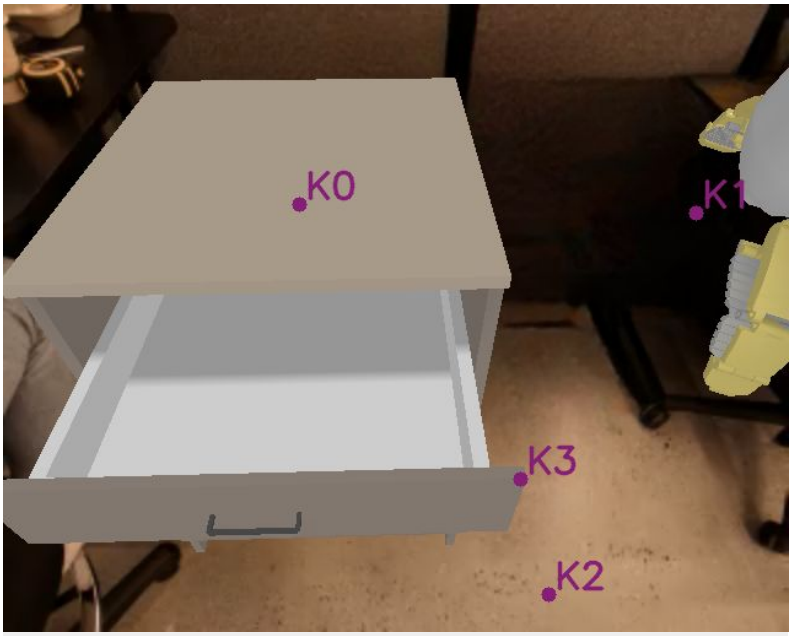


Prompt: The robot is working on a manipulation task. The high-level instruction for the manipulation. The task is put the carrot on the plate. Now you will be given an image that is annotated with multiple key-points, K0, K1, K2, K3, that are potential waypoints for the robot to grasp the carrot.

Can you tell me which keypoint the robot should move its gripper to grasp the carrot? You have four options as annotated on the image: K0, K1, K2, K3. Please return your selection. Your return should be one of the provided options.

Option A: K0
Option B: K3
Option C: K1
Option D: K2

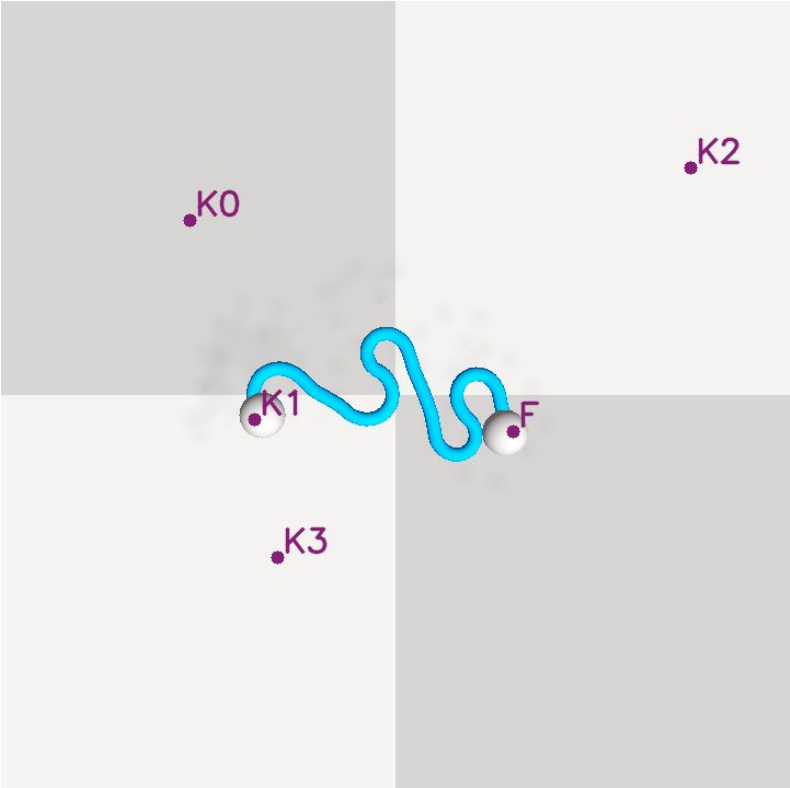
Figure 24: Question for *Place Carrot* in Simulation. Answer: Option A.



Prompt: The robot is working on a manipulation task. The high-level instruction for the manipulation task is to close the top drawer. Now you will be given an image that is annotated with multiple keypoints, K0, K1, K2, K3, that are potential waypoints for the robot to make the contact with the top drawer such that the robot can eventually close the top drawer. Can you tell me to which keypoint the robot should move its gripper to make contacts with the top drawer?

Option A: K2
Option B: K1
Option C: K0
Option D: K3

Figure 25: Question for *Close Drawer* in Simulation. Answer: Option D.

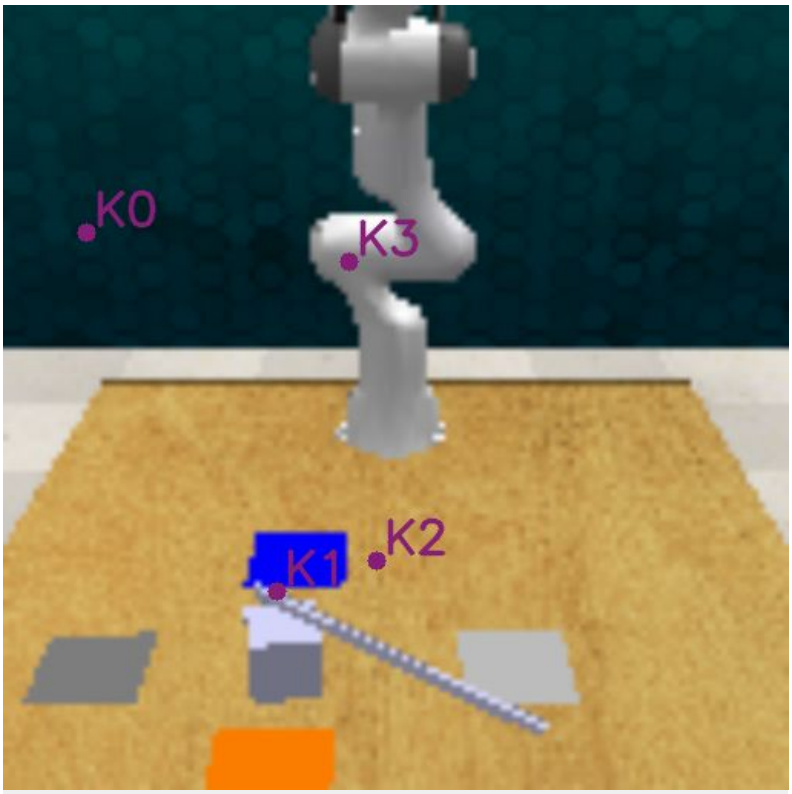


Prompt: The robot is working on a bimanual manipulation task. The high-level instruction for the task is to straighten the rope. Now you will be given an image that is annotated with multiple keypoints, F, K0, K1, K2, K3. Keypoint F is where the robot will grasp the rope with its one gripper. Keypoints K0, K1, K2, K3 are potential waypoints where the robot can grasp the rope with its other hand.

Can you tell me which keypoint the robot should move its gripper to grasp the rope? You have four options as annotated on the image: K0, K1, K2, K3. Please return your selection. Your return should be one of the provided options.

Option A: K2
Option B: K3
Option C: K1
Option D: K0

Figure 26: Question for *Straighten Rope* in Simulation. Answer: Option C.

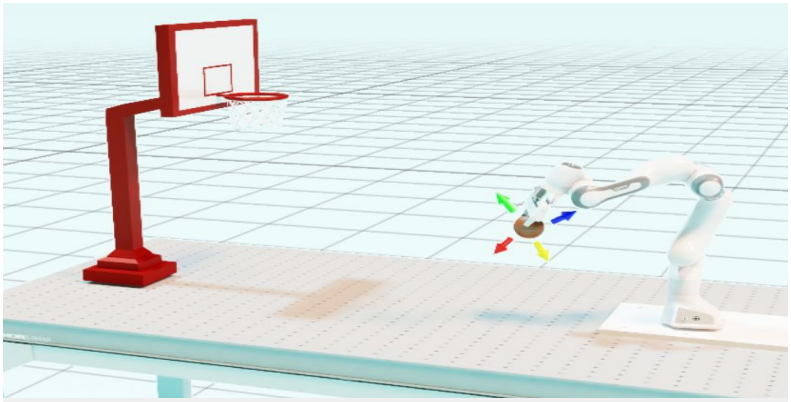


Prompt: The robot is working on a manipulation task. The high-level instruction for the manipulation task is to Pull the block towards the blue square. Now you will be given an image that is annotated with multiple keypoints, K0, K1, K2, K3, that are potential waypoints for the robot to grasp the stick.

Can you tell me at which keypoint the robot should grasp the stick? You have four options as annotated on the image: K0, K1, K2, K3. Please return your selection. Your return should be one of the provided options.

Option A: K0
Option B: K1
Option C: K2
Option D: K3

Figure 27: Question for *Sweep Object* in Simulation. Answer: Option B.

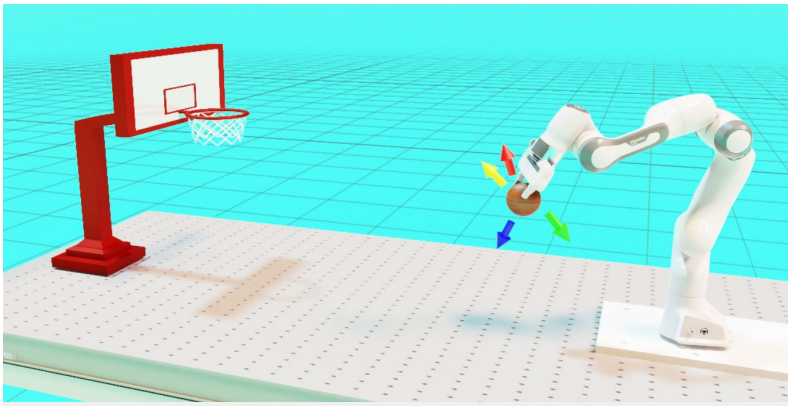


Prompt: The robot is working on a basketball shooting task. Now you will be given an image that is annotated with four arrows of different colors and directions. You need to help the robot to determine which arrow to follow in order to shoot the basketball into the basket.

The four arrows are green, red, blue, and yellow. These arrows decide the initial fly direction of the ball. Please return the color of the arrow that the robot should follow.

Option A: Red
Option B: Green
Option C: Yellow
Option D: Blue

Figure 28: Type 1 Question for the *Ball Shooting* task in simulation. Correct Answer: Option B.

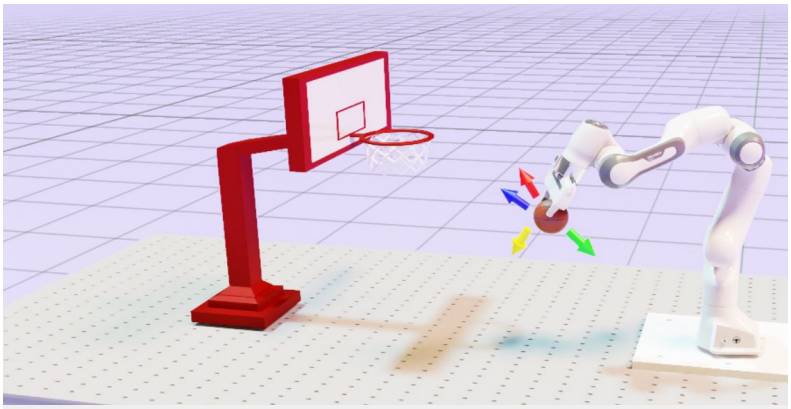


Prompt: The robot is working on a basketball shooting task. Now you will be given an image that is annotated with four arrows of different colors and directions. You need to help the robot to determine which arrow to follow in order to shoot the basketball into the basket. **Moreover, we want the ball to fly in the air as long as possible.**

The four arrows are green, red, blue, and yellow. These arrows decide the initial fly direction of the ball. Please return the color of the arrow that the robot should follow.

Option A: Red
Option B: Green
Option C: Yellow
Option D: Blue

Figure 29: Type 2 Question for the *Ball Shooting* task in simulation. Correct Answer: Option A.



Prompt: The robot is working on a basketball shooting task. Now you will be given an image that is annotated with four arrows of different colors and directions. You need to help the robot to determine which arrow to follow in order to shoot the basketball into the basket. **You should consider the effect of gravity on the ball and carefully estimate the trajectory of the ball.**

The four arrows are green, red, blue, and yellow. These arrows decide the initial fly direction of the ball. Please return the color of the arrow that the robot should follow.

Option A: Red
Option B: Green
Option C: Yellow
Option D: Blue

Figure 30: Type 3 Question for the *Ball Shooting* task in simulation. Correct Answer: Option A.