LoHoRavens: A Long-Horizon Language-Conditioned Benchmark for Robotic Tabletop Rearrangement

Anonymous ACL submission

Abstract

The integration of embodied agents with foundation models has led to notable progress in embodied instruction following. Specifically, the advanced reasoning capabilities of large language models (LLMs) and the visual perception skills of vision-language models (VLMs) enable robots to tackle complex, long-horizon tasks without requiring costly annotated demonstrations. However, there is still a lack of public benchmarks for evaluating the long-horizon reasoning capabilities of language-conditioned robots across different scenarios. To address this gap, this work introduces *LoHoRavens*, a simulation benchmark designed for tabletop rearrangement tasks. It includes 40 challenging tasks and addresses various aspects of long-horizon reasoning such as color, size, spatiality, arithmetic, reference, shape construction, commonsense, and occlusion. We evaluate two prevalent methods with current advanced VLMs (such as GPT-4o and Gemini 2.0 Flash) on this benchmark and conduct a thorough analysis of their reasoning performance. Our findings indicate that both methods struggle with numerous tasks, shedding light on the most challenging contexts that the community should be focusing on, as well as underscoring the need for continued effort to bridge gaps between modalities and improve current models.

1 Introduction

In embodied instruction following, an embodied agent such as a robot receives a language-based instruction and is expected to follow the instruction to complete the designated task. Of particular interest is long-horizon instruction following: how to endow embodied agents with long-horizon instruction following capabilities attracts more and more attention, as it mirrors real-world scenarios that are of practical importance in robotics. Long-horizon tasks involve high-level instructions that cannot be accomplished in just a few steps. Thus, the embodied agent must not only comprehend the language instruction well but also demonstrate advanced capabilities in long-horizon memorizing and complex reasoning. Thanks to the emergent abilities of LLMs and VLMs, embodied agents are able to borrow the rich knowledge and commonsense about the world and the strong reasoning capabilities from LLMs and VLMs, reducing the need for large expensive datasets of expert-annotated demonstrations.

With the rapid progress of LLMs and VLMs, robots are demonstrating increasingly impressive capabilities (Ahn et al., 2022; Driess et al., 2023; Brohan et al., 2023; Zitkovich et al., 2023; Ahn et al.; Black et al., 2024; Team et al., 2025); still, they struggle to solve some tasks that are relatively simple for a human child such as arranging objects on a table into a circle. Unlike the recent progress in NLP and computer vision, there are **unique challenges specific to robotics** that prevent robots from developing near-human behavior and intelligence, such as intensive interaction with environments, gaps between different modalities and difficulty of annotating domain data for deep learning based solutions. Long-horizon tasks further exacerbate these difficulties since they require multi-step complex reasoning across various aspects (e.g., commonsense, spatiality, color). Additionally, they require overcoming errors accumulated over multiple action steps, and bridging the modality gap between visual feedback, action planning, and action execution. Each of these challenges is crucial to the system's performance.

Despite its clear significance for autonomous language-conditioned robots, there is little prior work that systematically evaluates and quantifies these unique challenges. Existing benchmarks for long-horizon tasks fall into two main categories. (i) Real-world evaluations, e.g., Language-Table (Lynch et al., 2023) and LHManip (Ceola et al.), incorporates real-world uncertainties but at the cost of a quite limited scale

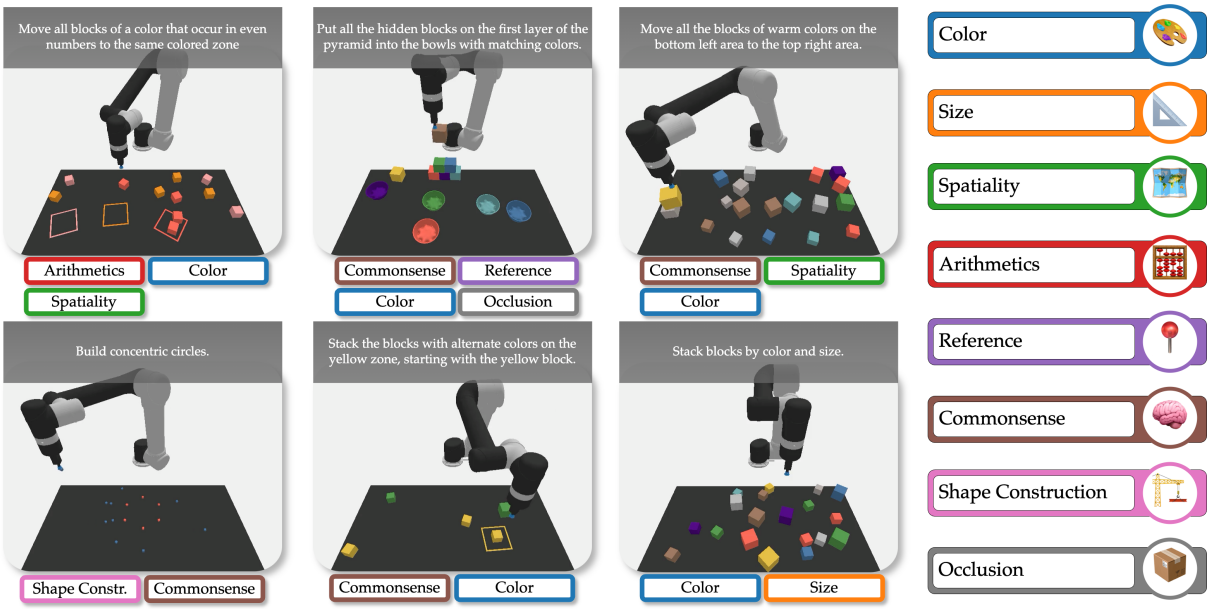

Figure 1: Examples of long-horizon tasks in LoHoRavens, highlighting the requirements of varying combinations of multiple reasoning capabilities for reasoning and planning to complete the tasks.

of evaluations. These evaluations are neither reproducible nor publicly accessible – making it difficult to verify, reproduce, or build upon previous results. (ii) Public simulated long-horizon benchmarks, e.g., RLBench (James et al., 2020) and CALVIN (Mees et al., 2022), either lack language-conditioning or require step-by-step human-provided sub-instructions. Category (ii) prevents autonomous long-horizon reasoning and instead evaluates short-horizon execution guided by human intervention, limiting its ability to test true high-level reasoning capabilities.[1]

To address this gap, this work introduces **LoHoRavens**, an open-source simulated benchmark designed specifically for long-horizon, language-conditioned robotic tabletop rearrangement tasks. LoHoRavens enables large-scale autonomous evaluations and provides a comprehensive analysis framework for the robotics community. Unlike prior benchmarks, LoHoRavens tasks require profound semantic understanding of high-level instructions and complex multistep reasoning capabilities without external step-by-step guidance. LoHoRavens covers a wide array of long-horizon reasoning and planning aspects including color, size, spatiality, arithmetic, reference, commonsense, shape construction and occlusion (see examples in Fig. 1). To solve each task, a robot must integrate multiple reasoning capabilities and formulate a comprehensive coherent long-horizon plan accordingly.

We further evaluate two prevalent methods on LoHoRavens benchmark, an imitation learning-based method and a Planner-Actor-Reporter method, using current state-of-the-art VLMs like GPT-4o and Gemini 2.0 Flash. We observe that these methods exhibit varied performance levels depending on the specific reasoning capabilities required by each task. Furthermore, both methods struggle greatly with long-horizon tasks, underscoring the need for continued improvement in long-horizon language-conditioned robotics tasks. To support ongoing research in this field, we publicly release the benchmark, trained models, and the corresponding codebase.

2 Related Work

2.1 Robotic Manipulation Benchmarks and Datasets

The interest in training language-conditioned models for robot manipulation has been growing in recent years thanks to the enormous advances in language processing techniques. As a result, many researchers proposed robotic manipulation datasets and benchmarks. RLBench (James et al., 2020), Ravens (Zeng et al., 2021; Shridhar et al., 2022), Robosuite (Zhu et al., 2020) introduce manipulation tasks in the household or tabletop-environment household tasks with their corresponding natural language instructions. VIMA-Bench (Jiang et al., 2023) is a robot manipulation learning benchmark supporting multimodal-prompting tasks. VLMbench (Zheng et al., 2022) contains 3D manipulation tasks with compositional

[1]Because this limitation of short-horizon scenarios, we limit our comparison in Table 1 to long-horizon benchmarks.

Benchmarks	type of environment	publicly released	replicable	large-scale train & eval	pick action	place action	language conditioned	# LoHo tasks	type of reasoning	max # primitive steps	color reasoning	size reasoning	spatial reasoning	arithmetic reasoning	reference reasoning	commonsense reasoning	shape construct reasoning	occlusion reasoning
FurnitureBench (Heo et al., 2023)	sim+real	✓	✓	✗	✓	✗	✗	8	auto	-	✗	✓	✓	✗	✓	✓	✗	✓
Behaviour-1K (Li et al., 2024)	sim	✓	✓	✗	✓	✓	✗	1000	auto	16	✓	✓	✓	✗	✓	✓	✗	✓
CALVIN (Mees et al., 2022)	sim	✓	✓	✗	✓	✓	✓	34	guided	5	✓	✗	✓	✗	✗	✓	✗	✗
Language-Table (Lynch et al., 2023)	sim+real	✗	✗	✗	✓	✓	✓	11	guided	8	✓	✗	✓	✓	✓	✓	✓	✗
LHManip (Ceola et al.)	real	✗	✗	✗	✓	✓	✓	20	auto	5	✓	✓	✓	✗	✓	✓	✗	✓
CLIPort (Shridhar et al., 2022)	sim+real	✓	✓	✓	✓	✓	✓	10	-	7	✓	✗	✓	✗	✓	✗	✗	✗
LoHoRavens (ours)	sim	✓	✓	✓	✓	✓	✓	40	auto	15	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of LoHoRavens with other long-horizon robotics benchmarks.

language instructions. RM-PRT (Ren et al., 2023) designs four progressive reasoning tasks and integrates the instruction parsing capabilities of LLMs. ARNOLD (Gong et al., 2023b) addresses the challenge of understanding continuous object states in complex tasks. OpenD (Zhao et al., 2022) addresses language-driven door and drawer opening. Open X-Embodiment (O'Neill et al., 2024) is a robotic manipulation dataset that contains 1M+ robot trajectories from 22 robot embodiments. Robo360 (Liang et al., 2023), D3IL (Jia et al., 2023), LEMMA (Gong et al., 2023a), and RoboScript (Chen et al., 2024a) are robotic manipulation benchmarks focusing on specific scenarios like evaluating closed-loop sensory feedback, multi-robot collaboration, or code generation. None of these benchmarks focuses on long-horizon tasks.

FurnitureBench (Heo et al., 2023) and Behaviour (Li et al., 2024) introduce simulated long-horizon benchmarks but they are not language-conditioned and thus do not focus on understanding semantic information of complex and ambiguous task instructions. Inner Monologue (Huang et al., 2023), Code as Policies (CaP; Liang et al. (2022)), and Language-Table (Lynch et al., 2023) build datasets for long-horizon language-conditioned manipulation tasks, but all of their long-horizon datasets are not open-source even though their code is partially released. LHManip (Ceola et al.) contains 20 real-world long-horizon manipulation tasks in cluttered tabletop environments; each task has a pair of natural language instructions and 10 demonstrations collected via teleoperation. However, their real-world scenarios limit the benchmark only to enable small-scale training and evaluations.

The works most similar to our proposed LoHoRavens are CALVIN (Mees et al., 2022) and GenSim (Wang et al., 2023a). CALVIN is also a simulated long-horizon language-conditioned manipulation benchmark. However, CALVIN provides step-by-step instructions and depends on the corresponding step-by-step evaluations to proceed. Therefore, the robot does not need to reason and plan for each step by itself to complete tasks. Furthermore, depending on the step-by-step evaluations severely limits the freedom of the benchmark. There are no alternative planning choices, even neglecting the step-by-step instructions.

Indeed, up to today, existing replicable[2] benchmarks, as summarized in Table 1, either neglect language-conditioned instructions (the main topic of this benchmark) or fail to account for autonomous long-horizon reasoning (relying on provided step-by-step sub-instructions). These limitations hinder progress in developing robots capable of fully autonomous, high-level reasoning in complex tasks. Instead, LoHoRavens allows for high-level instruction and evaluates policies based on the final states, thus is able to test a robot's long-horizon reasoning and planning capabilities. GenSim is an approach to generate robotic simulation tasks with LLMs. We make use of it to generate tasks. However, even with the most powerful commercial LLMs such as GPT-4 (OpenAI, 2024a), we still need much effort to check the code and correct the errors manually.

2.2 Foundation Models and Methods for Robot Learning

The emergent abilities of LLMs such as GPT-4 (OpenAI, 2024a), PaLM (Chowdhery et al., 2023), Gemini (Reid et al., 2024), Llama (Touvron et al., 2023; Dubey et al., 2024), Mixtral (Jiang et al., 2024), Claude (Anthropic, 2024), Qwen (Bai et al., 2023; Yang et al., 2024; Bai et al., 2025) have brought significant breakthroughs to many fields, including robotics, due to their rich knowledge and strong reasoning capabili-

[2]Meaning it can validated, used, or expanded, e.g., (Mees et al., 2022; Shridhar et al., 2022; Heo et al., 2023; Li et al., 2024).

ties. At the same time, there has been remarkable progress in the development of vision-language models as well, such as CLIP (Radford et al., 2021), BLIP-2 (Li et al., 2023), InstructBLIP (Dai et al., 2023), Flamingo (Alayrac et al., 2022), LLaVA (Liu et al., 2023), MiniGPT-4 (Zhu et al., 2024), CogVLM (Wang et al., 2025), Chameleon (Team, 2024), PaliGemma (Beyer et al., 2024), Molmo (Deitke et al., 2024), InternVL (Chen et al., 2024b) whose capabilities can be extended to robotic closed-loop control, enabling new levels of generalization. Moreover, there are also some foundation models such as SayCan (Ahn et al., 2022), PaLM-E (Driess et al., 2023), RT-1 (Brohan et al., 2023), and vision-language-action models such as RT-2 (Zitkovich et al., 2023), AutoRT (Ahn et al.), RT-2-X (O'Neill et al., 2024), Octo (Mees et al., 2024), OpenVLA (Kim et al., 2024), and π_0 (Black et al., 2024) which are especially designed for robot learning. With them, robots show more and more impressive capabilities and better generalization to new scenarios. Our work uses some of these LLMs and VLMs as baselines such as GPT-4o, Gemini 2.0 Flash, and Qwen2.5-VL to explore solutions to the hard challenge of long-horizon language-conditioned tasks.

Besides the two methods we use as baselines to test long-horizon tasks, there is also some work trying other ways for long-horizon manipulation tasks, such as Language-Table (Lynch et al., 2023) and VADER (Ahn et al., 2024), which explore using real-time interaction to complete long-horizon tasks. These methods can also be tested on our LoHoRavens benchmark.

3 LoHoRavens Benchmark

As far as we know, LoHoRavens is the first public benchmark supporting large-scale automatic evaluation for long-horizon language-conditioned robotic tabletop manipulation tasks, without requiring step-by-step instructions and evaluations for the high-level goal of each task (see the comparison with other long-horizon benchmarks in Table 1). In this section, we give details about the composition of the benchmark, as well as its inherent structure, design, and evaluation framework.

3.1 Simulation Environment

LoHoRavens is built on the **Ravens** robot simulator (Zeng et al., 2021; Shridhar et al., 2022) by extending it to **Lo**ng-**Ho**rizon tasks. We chose Ravens as the base simulator because it is a well-established simulator and widely used for robotic manipulation tasks such as in CLIPort (Shridhar et al., 2022), VIMA-Bench (Jiang et al., 2023). We use the main pick-and-place action primitive supported by Ravens to construct LoHoRavens. Though pick-and-place seems simple, its combinations cover a wide range of manipulation tasks and can be used to test very complex reasoning capabilities of agents: see Table 1. In the LoHoRavens simulation environment, there are a UR5e robot arm with a suction gripper and some objects on the table. Given a high-level language based instruction (e.g., "stack all the blocks of the same size"), the robot is supposed to rearrange these objects to a desired state. The input to the robot is language instructions and visual observations in the form of top-down RGB-D images from three cameras positioned around a rectangular table. The action space of the robot consists of a language-conditioned pick-and-place motion primitive which is parameterized by two end-effector poses at each time step. Moreover, to simulate disturbance in the real world, we add noise and perturbations to the robot's environment at test time. Following Inner Monologue (Huang et al., 2023), we add Gaussian noise $\mathcal{N}(0, 3)$ to pixel observations and $\mathcal{N}(0, 2.5)$ to policy primitive outputs.

3.2 Tasks and Dataset

Currently, LoHoRavens contains 40 long-horizon tasks. To support complex long-horizon reasoning, there are three low-level pick-and-place primitives that can be used by the foundation model planner: (i) the vanilla *pick-and-place-with-color primitive*, e.g., "pick up the red block and place it on the yellow block", (ii) the *pick-and-place-with-size primitive*, e.g., "pick up the smaller red block and place it on the bigger yellow block", (iii) the *pick-and-place-with-spatiality primitive*, e.g., "pick up the red block and place it on the top right area". In addition to the three pick-and-place primitives, LoHoRavens contains 30 manually implemented tasks and 10 tasks automatically generated with the help of GenSim (Wang et al., 2023a).[3]

LoHoRavens covers three kinds of basic objects: block, bowl and zone (see Fig. 1). We made this choice because we do not intend to study the robot's generalization capability to new or unseen object

[3]We use GenSim with GPT-4o to generate tasks. For each task, we check the automatically generated code and modify it if necessary.

types in this work. Instead, we focus on the long-horizon reasoning capabilities that are related to the general attributes of objects like size, color and spatial position. Such reasoning capabilities can be generalized to other objects as well. In addition to these general object attributes, we are also interested in the reasoning capabilities related to attributes of multiple objects. So we include several tasks to test arithmetic and reference reasoning capabilities (e.g., "Move all blocks of a color that occur in even numbers to the same colored zone."). Moreover, one of the most important reasoning capabilities is commonsense reasoning. The tasks in LoHoRavens range from simple color commonsense reasoning (e.g., "stack the blocks of warm colors") to complex shape construction (e.g., "construct concentric circles"). The most complex task requires commonsense reasoning about what the shape to be constructed is first, then manipulating as many as sixteen objects where each object has to be positioned precisely. Another interesting reasoning capability is to find hidden objects which are out of sight (e.g., "pick up the blue block on the bottom layer of the pyramid"). To solve this kind of task, the agent must move all the objects on top of the target object first, which poses further challenges to the agent's reasoning capabilities. Fig. 2 shows the proportion of each reasoning capability in the 40 tasks.

To understand how the agent performs on the tasks, we provide a large-scale dataset for training and automatic evaluation. The simulator of LoHoRavens can generate large-scale expert demonstrations automatically with the scripted oracle program as used in CLIPort and VIMA-Bench. The oracle agent has access to ground-truth pick and place poses and uses pre-specified heuristics to complete the tasks. All the tasks can be instantiated into thousands of task instances with different random seeds. The generated large-scale expert demonstrations can be used for further imitation learning or video related research. To ensure we have good enough pick-and-place primitives, 20,000 demonstrations are generated for training each primitive. Then they are trained together with multi-task training for 12,000,000 steps. The final trained multi-task primitive achieves a performance of 91.83%.

To build the benchmark, we generate, for each long-horizon task, 1,000 demonstrations as the train set, 100 demonstrations as the validation set, and 200 demonstrations as the test set. Note that there are 16 colors for each object in the benchmark,

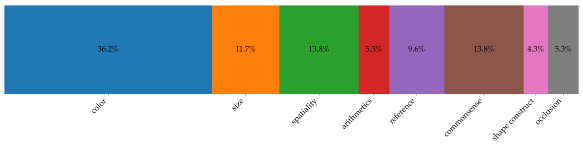

Figure 2: Reasoning capability frequency across LoHoRavens tasks. Tasks that combine reasoning types are multiply counted. See Fig. 1.

and the colors of objects are chosen randomly, so they are generally different in training, validation and test sets. We split the tasks into 20 seen tasks and 20 unseen tasks. The seen tasks are used for training and writing prompts. The unseen tasks are used to evaluate the model's generalization abilities to new tasks. Most of the task instances need at least five steps to complete. Some tasks need 15 steps to get to the correct final state.

3.3 Evaluation

For each task, there are one or more manually-defined ground-truth final states. Depending on the task, there are two different match methods for evaluating whether the states of the objects are correct compared to the ground-truth states. One is *pose match*: an object's position and rotation are the same as ground truth. The other is *zone match*: the overlap area of two objects is larger than a threshold.

LoHoRavens uses two measures to evaluate the success rate of a task. The first one is *binary success rate*. If the final state of objects is the same as the ground truth, the score is 1, otherwise, it is 0. The other evaluation measure is a *partial reward-based score*, in the range $[0, 1]$. The score assigns the partial rewards according to the proportion of successful pick-and-place steps. For example, if a task needs ten pick-and-place steps to complete, and the test model finishes eight of them, the score is $8/10 = 80\%$.

4 Experiments

4.1 Baseline Methods

Imitation Learning Based Model (IL) We use the same architecture and training recipe as CLIPort for the imitation learning baseline. Using multi-task training, the CLIPort model is trained with the train sets of all 20 seen tasks along with the three pick-and-place primitives for 100K steps. Because the vanilla CLIPort does not know when to stop the execution, following Inner Monologue and CaP, we use an oracle termination variant that

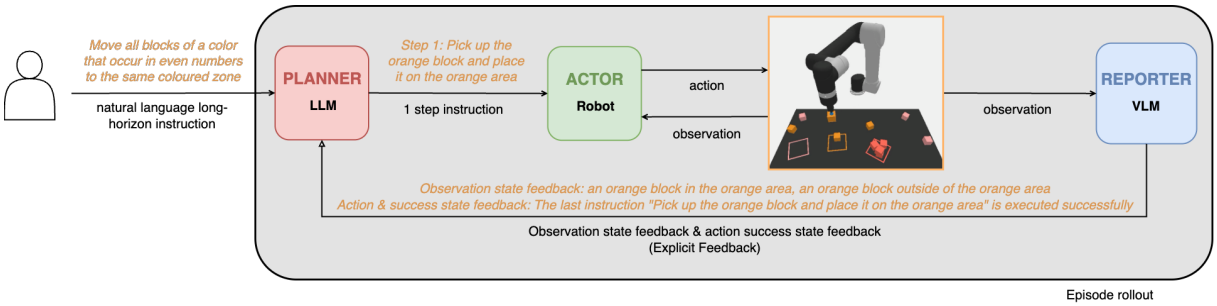

Figure 3: The Planner-Actor-Reporter (PAR) baseline takes the human input and asks an LLM to create the next step that needs to be done in order to achieve the task.

uses the oracle information from the simulator to detect the success and stop the execution process.

Planner-Actor-Reporter Based Model (PAR) The Planner-Actor-Reporter paradigm is frequently used in robotics (Dasgupta et al., 2022; Huang et al., 2023; Wang et al., 2023b). Usually, as shown in Fig.3, LLMs serve as the Planner due to their impressive planning and reasoning capabilities, and humans or VLMs play the role of Reporter to provide necessary language feedback for the Planner's planning. The Actor is the agent that interacts with the environment. Specifically, we use Llama-3 8B (Dubey et al., 2024) and the trained pick-and-place CLIPort primitives as the Planner and Actor, respectively. For the Reporter, we use the VLM CogVLM2 (Wang et al., 2025). We also conduct smaller-scale experiments using the powerful commercial model GPT-4o and Gemini 2.0 Flash as the Planner and Reporter in §4.3.

We create 10-shot examples for both LLM and VLM prompts and use them for both seen and unseen tasks. When a step's action has been executed, there will be a top-down RGB image rendered by the simulator. The VLM as the Reporter module will generate the caption feedback based on the current image or the whole image history. This caption feedback is sent to the LLM for its next-step planning. The Planner-Actor-Reporter closed-loop process will be iteratively executed until the high-level goal is achieved or the maximum number of trial steps has been exceeded.

4.2 Experimental Results

Our experiments are designed to evaluate the whole simulated robotic policy's performance, rather than evaluating the components of LLM planner and visual feedback separately. On one hand, evaluating the whole system is more in line with real practical needs. On the other hand, LLMs cannot read the visual observation directly, therefore it's hard to evaluate the planning capabilities of LLMs in

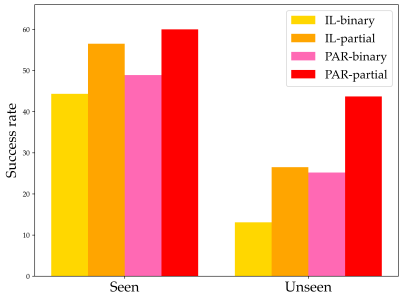

Figure 4: Performance of the imitation learning-based model (IL) and the Planner-Actor-Reporter based model (PAR) on the seen tasks and unseen tasks of LoHoRavens.

isolation on a large scale.

We aim to answer the following questions from the experiments and analysis in §4.3: (i) How do the two baselines perform on the long-horizon tasks in the LoHoRavens benchmark? (ii) How do the models perform under different combinations of reasoning capabilities? (iii) How do the gaps between the modalities language, vision, and action influence the performance of models?

Fig. 4 shows how the two baselines perform on all seen and unseen tasks. Numbers are averages over tasks. We can see that the imitation learning-based CLIPort model (IL) performs a little worse than the Planner-Actor-Reporter based model (PAR) on seen tasks. However, when generalizing to the unseen tasks, the IL model drops quite a lot while the PAR counterpart is relatively less affected. The binary success rate of both models is quite low, indicating it is hard for them to finish all the steps of the long-horizon tasks.

We then investigate the effects of different reasoning capabilities. Due to the low binary success rates, we only use the partial reward based metric to study model performance under different combinations of reasoning capabilities. As we can see from Fig. 5, the overall tendency is that model performance drops as the number of reasoning capabilities required increases. This observation fits with our intuition that the more reasoning capabilities

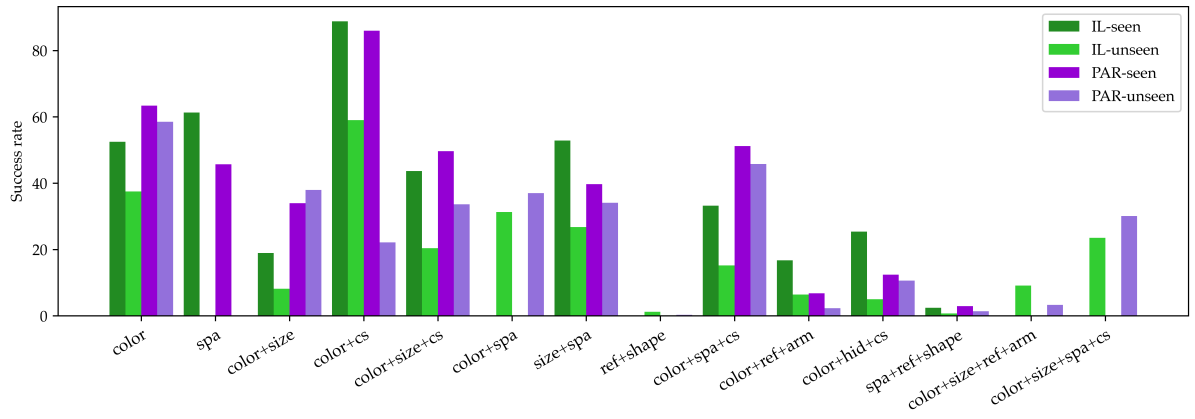

Figure 5: Performance of the imitation learning-based model (IL) and the Planner-actor-reporter-based model (PAR) under different combinations of reasoning capabilities. (spa = spatiality, arm = arithmetics, cs = commonsense, hid = occlusion, ref=reference, shape = shape construction)

VLMs	VLM as P		VLM as P & R	
	Partial	Binary	Partial	Binary
GPT-4o	23.96	10.50	6.76	0.2
GPT-4o-mini	23.41	7.70	7.04	0.4
Gemini 2.0 Flash	25.20	10.30	8.97	1.1
Qwen-2.5-VL-7B	11.63	2.80	3.41	0.1
Qwen-2.5-VL-72B	13.11	4.17	3.50	0.1

Table 2: Performance of different VLMs in the PAR framework.

are required, the harder the tasks become.

Another interesting finding is that the two baselines perform differently regarding different reasoning capabilities. On the seen tasks requiring spatial reasoning capability, the IL model usually performs better. It is probably because current LLMs and VLMs do not have good spatial understanding. In contrast, the PAR model usually outperforms the IL model on tasks requiring commonsense. Another observation is that the PAR model cannot deal with tasks requiring reference since LLMs cannot indicate the objects accurately if there are more than one object with the same size and color. This also prevents the PAR model from solving tasks requiring arithmetic reasoning since these tasks usually comprise multiple objects of the same kind.

The experiments also show that some tasks are extremely hard for both models. For tasks that contain occluded objects, both models struggle to reason to remove the top object that blocks the bottom target objects. Moreover, they are almost completely unable to solve shape construction tasks.

4.3 Analysis and Case Study

Performance of Different VLMs in PAR Different VLMs have different strengths and weaknesses. Therefore, besides CogVLM2, we further test some other prominent VLMs, including Gemini 2.0 Flash (DeepMind, 2025), GPT-4o (OpenAI, 2024b), and Qwen-2.5-VL-7B/72B (Bai et al., 2025). We select ten hard tasks in LoHoRavens (i.e., the baseline method Llama3+CogVLM2 shows subpar performance) and test two settings for these VLMs: (i) using the VLM solely as a Planner with image inputs; (ii) using the VLM as both Planner and Reporter, where it first generates textual observations as a Reporter, then uses these descriptions for planning as a Planner. As shown in Table 2, the performance of the popular commercial VLMs such as GPT-4o and Gemini 2.0 Flash is quite close, but the current prominent open-sourced VLM Qwen-2.5-VL is far behind. Moreover, even the most powerful commercial VLMs still struggle to solve the challenging long-horizon tasks in LoHoRavens, indicating the necessity of developing better models for such long-horizon reasoning tasks.

Case Study We further perform case studies with the powerful GPT-4o model on the 10 hard LoHoRavens tasks. Table 3 demonstrates that GPT-4o, functioning solely as Planner, surpasses GPT-4o serving both as Planner and Reporter in nearly all tasks. This suggests that intermediate observation descriptions will bring information loss and further interrupt the Planner's strategy. Moreover, the GPT-4o Reporter severely struggles with object enumeration, which likely hinders its ability to aid the Planner in the arithmetic reasoning task. While GPT-4o clearly outperforms Llama 3 8B+CogVLM2, it is still completely incapable of solving tasks involving occlusion and shape construction. Many actions in these tasks are not explicitly described in instructions, requiring the model to infer them based

Tasks	BM	GPT-4 as P		GPT-4 as P&R	
	Prt	Prt	Bin	Prt	Bin
Move blocks between absolute positions. (cl+spa)	20.30	48.58	14.00	16.79	1.00
Move blocks between absolute positions by size. (size+spa)	20.20	37.62	10.00	16.55	0.00
Move blocks between absolute positions by color. (cl+spa+cs)	25.20	38.36	7.00	17.40	0.00
Move blocks between absolute positions by color and size. (cl+size+spa+cs)	18.50	31.53	3.00	16.51	1.00
Move all blocks of a color that occur in even numbers to the same colored zone. (cl+ref+arm)	8.70	78.81	67.00	0.31	0.00
Stack blocks by absolute position and color in size order. (cl+size+spa+cs)	0.00	4.67	4.00	0.00	0.00
Put all the hidden objects in 3-layer stacked towers into the bowls with matching colors. (cl+hid+cs)	0.00	0.02	0.00	0.03	0.00
Put the hidden objects in the pyramid into the bowls with matching colors. (cl+hid+cs)	0.00	0.01	0.00	0.00	0.00
Build concentric circles. (ref+shape+cs)	0.00	0.00	0.00	0.00	0.00
Build a rectangle on the zone. (ref+shape+cs)	0.00	0.00	0.00	0.00	0.00

Table 3: Results of GPT-4o on 10 hard LoHoRavens tasks. We test the performance of GPT-4o as only Planner (P) and as both Planner&Reporter (P&R) against the baseline (BM = Llama3+CogVLM2) on 100 instances per task. *Partial* (Prt): the success of intermediate steps. *Binary* (Bin): the success of finishing the whole task. cl = color. See Fig. 5 for other task abbreviations.

Tasks	GPT-4 as P		GPT-4 as P & R		
	Planner	Actor	Planner	Actor	Reporter
Move blocks between absolute positions by color.	58.1	35.8	68.5	37.5	60.7
Move blocks between absolute positions by size and color.	73.3	59.8	100.0	-	64.3
Move all blocks of a color that occur in even numbers to the same colored zone.	40.9	32.7	-	-	100.0
Put the hidden objects in the pyramid into the bowls with matching colors.	83.3	-	-	-	100.0
Build a rectangle on the zone.	100.0	-	-	-	100.0

Table 4: Error analysis for five tasks with GPT-4o-mini (GPT4) models. We use precision ($n_{\text{correct\_outputs}}/n_{\text{all\_outputs}}$) to analyze errors of Planner and Reporter. We use plan-conditioned precision ($n_{\text{successful\_exec}}/n_{\text{correct\_plans}}$) to analyze Actor errors. We don't report other modules' error rates if the error rate of Planner or Reporter is too high.

on commonsense knowledge. This indicates the need for more refined prompts and alternative approaches like Code as Policies (Liang et al., 2022) for these complex challenges.

Ablation Study For 5 typical tasks, we manually examine 10 failed instances each to categorize the errors and quantify the modality gaps. We calculate the precision of Planner's planning (number correctly generated plans / total number of plans) and the plan-conditioned precision of Actor's actions (number of actions executed correctly / number correctly generated plans). Table 4 reveals that the Actor often fails, particularly in spatially related tasks, despite a high success rate depicted in § 3.2. We hypothesize that the failures are due to: (i) the Planner generating incorrect instructions not present in the primitive training set, and (ii) the primitive's inability to generalize well to entirely new situations. With GPT-4 as the Planner (column 2), issues arise when (i) it occasionally produces the incorrect format despite precise formatting prompts, and (ii) it struggles with complex instructions and managing its previous history without highly task-specific prompt design. For GPT-4 functioning as both Planner and Reporter, the primary issue is the Reporter's performance on complex tasks, highlighting the big gap between vision and language/actions; thus, we omit reporting error rates for other modules if the Reporter's error rate is excessively high. The Reporter struggles to accurately describe object positions and count objects. Sometimes it cannot even recognize the correct color of objects. Additionally, it's hard for the Planner to plan accurately for complicated long-horizon tasks. These observations suggest that the modality gaps between language, vision, and actions have a significant impact on the long-horizon performance of the models.

5 Conclusion

We introduce LoHoRaves, the first open-source long-horizon language-conditioned tabletop rearrangement benchmark. It covers diverse reasoning capabilities, such as color, size, spatiality, arithmetic, reference, shape construction, commonsense, and occlusion. Two popular baselines perform well on some subsets of the reasoning tasks. However, their performance on other tasks is poor. These findings indicate that LoHoRavens contains highly challenging tasks. We believe LoHoRavens will be beneficial for evaluating and guiding future research in robotics field.

6 Limitations

Due to the design of Ravens simulator, current LoHoRavens benchmark only contains tasks which can be evaluated based on only final states. So any tasks require detecting the states of middle process cannot be added to the benchmark. Moreover, many tasks in LoHoRavens require the final position and states of objects are quite certain and fixed. Take the tasks of stacking blocks as an example, the task should be designed as the format of stacking blocks on a specified zone. Otherwise, the simulator cannot support the evaluations. LoHoRavens only contains three very basic objects, so it does not test reasoning capabilities based on daily objects' commonsense.

Use of AI Assistance We used AI assistant tools (ChatGPT and GitHub Copilot) to aid in rewriting code and text. All AI-generated content was thoroughly reviewed and verified by the authors. AI was not used to generate new research ideas or original findings; rather, it served as a support tool to improve clarity, efficiency, and organization. In accordance with ACL guidelines, our use of AI aligns with permitted assistance categories, and we have transparently reported all relevant usage in this paper. While AI contributed to enhancing the quality of the work, no direct research outputs are the result of AI assistance.

References

Michael Ahn, Montserrat Gonzalez Arenas, Matthew Bennice, Noah Brown, Christine Chan, Byron David, Anthony Francis, Gavin Gonzalez, Rainer Hessmer, Tomas Jackson, and 1 others. 2024. Vader: Visual affordance detection and error recovery for multi robot human collaboration. *arXiv preprint arXiv:2405.16021*.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, and 1 others. 2022. Do as i can, not as i say: Grounding language in robotic affordances. In *CoRL*.

Michael Ahn, Debidatta Dwibedi, Chelsea Finn, Montserrat Gonzalez Arenas, Keerthana Gopalakrishnan, Karol Hausman, Alex Irpan, Nikhil J Joshi, Ryan Julian, Sean Kirmani, and 1 others. Autort: Embodied foundation models for large scale orchestration of robotic agents. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 35:23716–23736.

Anthropic. 2024. Claude ai. Large language model.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, and 1 others. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, and 1 others. 2024. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, and 1 others. 2023. Rt-1: Robotics transformer for real-world control at scale. In *RSS*.

Federico Ceola, Lorenzo Natale, Niko Suenderhauf, and Krishan Rana. Lhmanip: A dataset for long-horizon language-grounded manipulation tasks in cluttered tabletop environments. In *RSS 2024 Workshop: Data Generation for Robotics*.

Junting Chen, Yao Mu, Qiaojun Yu, Tianming Wei, Silang Wu, Zhecheng Yuan, Zhixuan Liang, Chao Yang, and 1 others. 2024a. Roboscript: Code generation for free-form manipulation tasks across real and simulation. *arXiv:2402.14623*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *JMLR*, 24(240).

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv:2305.06500*.

Ishita Dasgupta, Christine Kaeser-Chen, Kenneth Marino, Arun Ahuja, Sheila Babayan, Felix Hill, and Rob Fergus. 2022. Collaborating with language models for embodied reasoning. In *Second Workshop on Language and Reinforcement Learning*.

DeepMind. 2025. Gemini 2.0 flash.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, and 1 others. 2024. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv:2409.17146*.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, and 1 others. 2023. Palm-e: an embodied multimodal language model. In *ICML*, pages 8469–8488.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ran Gong, Xiaofeng Gao, Qiaozi Gao, Suhaila Shakiah, Govind Thattai, and Gaurav S Sukhatme. 2023a. Lemma: Learning language-conditioned multi-robot manipulation. *RA-L*.

Ran Gong, Jiangyong Huang, Yizhou Zhao, Haoran Geng, Xiaofeng Gao, Qingyang Wu, Wensi Ai, Ziheng Zhou, Demetri Terzopoulos, Song-Chun Zhu, and 1 others. 2023b. Arnold: A benchmark for language-grounded task learning with continuous states in realistic 3d scenes. In *ICCV*.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. 2023. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *RSS*.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, and 1 others. 2023. Inner monologue: Embodied reasoning through planning with language models. In *CoRL*, pages 1769–1782.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. 2020. Rlbench: The robot learning benchmark & learning environment. *RA-L*, 5(2).

Xiaogang Jia, Denis Blessing, Xinkai Jiang, Moritz Reuss, Atalay Donat, Rudolf Lioutikov, and Gerhard Neumann. 2023. Towards diverse behaviors: A benchmark for imitation learning with human demonstrations. In *ICLR*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. 2023. Vima: General robot manipulation with multimodal prompts. In *ICML*.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, and 1 others. 2024. Openvla: An open-source vision-language-action model. In *CoRL*.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, and 1 others. 2024. Behavior-1k: A human-centered, embodied ai benchmark with 1, 000 everyday activities and realistic simulation. *CoRR*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2022. Code as policies: Language model programs for embodied control. In *CoRL 2022 Workshop LangRob*.

Litian Liang, Liuyu Bian, Caiwei Xiao, Jialin Zhang, Linghao Chen, Isabella Liu, Fanbo Xiang, Zhiao Huang, and Hao Su. 2023. Robo360: A 3d omnispective multi-material robotic manipulation dataset. *arXiv preprint arXiv:2312.06686*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *NeurIPS*, 36:34892–34916.

Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. 2023. Interactive language: Talking to robots in real time. *RA-L*.

Oier Mees, Dibya Ghosh, Karl Pertsch, Kevin Black, Homer Rich Walke, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, and 1 others. 2024. Octo: An open-source generalist robot policy. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. 2022. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *RA-L*.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, and 1 others. 2024. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration. In *ICRA*, pages 6892–6903. Institute of Electrical and Electronics Engineers Inc.

OpenAI. 2024a. Gpt-4. Large language model.

OpenAI. 2024b. Hello gpt-4o.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Pengzhen Ren, Kaidong Zhang, Hetao Zheng, Zixuan Li, Yuhang Wen, Fengda Zhu, Mas Ma, and Xiaodan Liang. 2023. Rm-prt: Realistic robotic manipulation simulator and benchmark with progressive reasoning tasks. *arXiv preprint arXiv:2306.11335*.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2022. Cliport: What and where pathways for robotic manipulation. In *CoRL*, pages 894–906.

Chameleon Team. 2024. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, and 1 others. 2025. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. 2023a. Gensim: Generating robotic simulation tasks via large language models. In *ICLR*.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, and 1 others. 2025. Cogvlm: Visual expert for pretrained language models. *NeurIPS*, 37:121475–121499.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, Yitao Liang, and Team CraftJarvis. 2023b. Describe, explain, plan and select: interactive planning with large language models enables open-world multi-task agents. In *NeurIPS*, pages 34153–34189.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, and 1 others. 2021. Transporter networks: Rearranging the visual world for robotic manipulation. In *CoRL*, pages 726–747.

Yizhou Zhao, Qiaozi Gao, Liang Qiu, Govind Thattai, and Gaurav S Sukhatme. 2022. Opend: A benchmark for language-driven door and drawer opening. *arXiv preprint arXiv:2212.05211*.

Kaizhi Zheng, Xiaotong Chen, Odest Jenkins, and Xin Eric Wang. 2022. VLMbench: A compositional benchmark for vision-and-language manipulation. In *NeurIPS Datasets and Benchmarks Track*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*.

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

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, and 1 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, pages 2165–2183.

A Task List

We list the tasks in LoHoRavens in Table 5. The tasks are designed to test various reasoning capabilities, including color, size, spatiality, commonsense, reference, and arithmetic. Each task is associated with the reasoning required to complete it.

Task list	Reasoning required
Stack all the blocks on a zone.	color
Stack blocks of the same color.	color
Stack blocks in alternate colors.	color
Stack blocks by color.	color, commonsense
Stack blocks of the same size.	color, size
Stack blocks by color and size.	color, size, commonsense
Stack blocks by color in size order.	color, size, commonsense
Stack smaller blocks over bigger blocks of the same color.	color, size
Stack blocks of the same color in the zone with the same color, with the bigger blocks underneath.	color, size
Stack blocks by relative position and color.	color, spatiality, commonsense
Stack blocks by relative position and color and size.	color, size, spatiality, commonsense
Stack blocks by absolute position and color in size order.	color, size, spatiality, commonsense
Stack blocks by absolute position and color and size.	color, size, spatiality, commonsense
Move blocks between absolute positions.	spatiality
Move blocks between absolute positions by size.	size, spatiality
Move blocks between absolute positions by color.	color, spatiality
Move blocks between absolute positions by color and size.	color, size, spatiality
Move all blocks of a color that occur in even numbers to the same colored zone.	color, reference, arithmetic
Move all blocks of a color that occur in odd numbers to the same colored zone.	color, reference, arithmetic
Put the blocks into the bowls with matching colors.	color
Put the blocks into the bowls with mismatching colors.	color
Put the hidden color object under the color object into the bowls with matching colors.	color, hidden objects, commonsense
Put all the hidden objects in two-layer stacked towers into the bowls with matching colors.	color, hidden objects, commonsense
Put all the hidden objects in three-layer stacked towers into the bowls with matching colors.	color, hidden objects, commonsense
Put the hidden objects in the pyramid into the bowls with matching colors.	color, hidden objects, commonsense
(step-by-step) Put the blocks into the bowls with matching colors.	color
(step-by-step) Stack all the blocks on a zone.	color
(step-by-step) Stack blocks by relative position and color.	color, spatial, commonsense
(step-by-step) Put the hidden color object under the color object into the bowls with matching colors.	color, hidden objects, commonsense
(step-by-step) Move blocks between absolute positions.	color, spatiality
Align color boxes on line.	color, spatiality
Align size boxes on circle.	color, size, spatiality
Align size boxes on line.	color, size, spatiality
Build concentric circles.	spatiality, reference, shape
Stack most colored blocks.	color, reference, arithmetics
Put max even number blocks into same colored zone.	color, reference, arithmetics
Put max odd number blocks into same colored zone in size order.	color, size, reference, arithmetics
Construct circle with block in the middle.	spatiality, reference, shape
Construct letter M.	spatiality, reference, shape, commonsense
Build rectangular on the zone.	reference, shape

Table 5: LoHoRavens task list