2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3315-4393-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IROS60139.2025.11247026

Open-World Task Planning for Humanoid Bimanual Dexterous Manipulation via Vision-Language Models

Zixin Tang[1], Zhihao Li[1], Junjia Liu[1], Zhuo Li[1], and Fei Chen[†1], *Senior Member, IEEE*

Abstract— **Open-world task planning, characterized by handling unstructured and dynamic environments, has been increasingly explored to integrate with long-horizon robotic manipulation tasks. However, existing evaluations of the capabilities of these planners primarily focus on single-arm systems in structured scenarios with limited skill primitives, which is insufficient for numerous bimanual dexterous manipulation scenarios prevalent in the real world. To this end, we introduce OBiMan-Bench, a large-scale benchmark designed to rigorously evaluate open-world planning capabilities in bimanual dexterous manipulation, including task-scenario grounding, workspace constraint handling, and long-horizon cooperative reasoning. In addition, we propose OBiMan-Planner, a vision-language model-based zero-shot planning framework tailored for bimanual dexterous manipulation. OBiMan-Planner comprises two key components, the scenario grounding module for grounding open-world task instructions with specific scenarios and the task planning module for generating sequential stages. Extensive experiments on OBiMan-Bench demonstrate the effectiveness of our method in addressing complex bimanual dexterous manipulation tasks in open-world scenarios. The code, benchmark, and supplementary material are released at `https://github.com/Zixin-Tang/OBiMan`.**

I. Introduction

Task and Motion Planning (TAMP) framework [1], [2] has been widely used in manipulation skill learning approaches to reduce the difficulties in handling long-horizon tasks by leveraging task planners to decompose complex tasks into sequential stages. Classical task planners are mainly based on predicate logic solvers, which require complete definitions using standardized domain languages such as the Planning Domain Definition Language (PDDL) [3], [4]. Recently, benefitting from the excellent capabilities in reasoning and common sense understanding of large language models (LLMs) and vision-language models (VLMs) [5], [6], [7], large model-based task planners have emerged [8], [9], [10], [11], [12], [13], [14]. These advances alleviate the need for labor-intensive PDDL definitions and pave the way for open-world task planning in handling unstructured and dynamic environments.

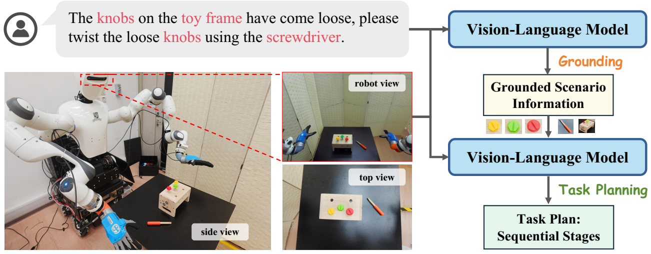
The knobs on the toy frame have come loose, please twist the loose knobs using the screwdriver.
robot view
top view
side view
Vision-Language Model
Grounding
Grounded Scenario Information
Vision-Language Model
Task Planning
Task Plan: Sequential Stages

Fig. 1: VLMs-based framework for open-world task planning. The framework leverages VLMs to ground task instructions with specific scenario observations, generating grounded scenario information about task-relevant objects and their properties. This information is then processed by the VLM to propose feasible task plans.

However, current manipulation tasks designed to validate the planning performance in many works remain insufficient in several critical aspects. First, they primarily focus on single-arm systems with limited skill primitives such as pick-and-place [15], [16], while neglecting numerous bimanual dexterous skills prevalent in the real world, such as wiping, inserting, and handover, have not yet been fully explored. Second, these tasks heavily rely on well-defined scenario descriptions to specify critical properties of task-relevant objects, such as names and locations [17], [11], which constrain adaptability in unstructured or dynamic scenarios. Third, they also lack large-scale and systematic benchmarks to identify the quantitative performance of task planners. Addressing these gaps is essential for advancing the field toward developing open-world task planners capable of handling bimanual dexterous manipulation.

To this end, we propose **OBiMan-Bench**, a large-scale benchmark to rigorously evaluate open-world task planners for bimanual dexterous manipulation. This benchmark comprises 18 skill primitives and 100 open-world tasks. Previous works [18], [19], [20] have established taxonomies for bimanual dexterous manipulation considering characteristics coordinated/uncoordinated, coupled/uncoupled, and symmetric/asymmetric for categorization. Although these factors provide valuable insights into analyzing coordination levels of tasks, excessive subdivision makes identification less intuitive and harder to interpret. In OBiMan-Bench, we simplify bimanual manipulation into two patterns and develop a skill-oriented bimanual coordination taxonomy, facilitating the classification of bimanual manipulation tasks into three coordination levels.

OBiMan-Bench emphasizes the critical capability of task planners to handle open-world tasks. For this purpose, the difficulty of tasks is enhanced by requiring planners to infer

This work was supported in part by the Research Grants Council of the Hong Kong SAR under Grant 14211723, 14222722, 24209021 and C7100-22GF, and in part by InnoHK of the Government of Hong Kong via the Hong Kong Centre for Logistics Robotics.

[1] Zixin Tang, Zhihao Li, Junjia Liu, Zhuo Li, and Fei Chen are with the Department of Mechanical and Automation Engineering, T-Stone Robotics Institute, The Chinese University of Hong Kong, Hong Kong SAR (email: `zxtang@mae.cuhk.edu.hk`, `zhihaoli@mae.cuhk.edu.hk`, `jjliu@mae.cuhk.edu.hk`, `zli@mae.cuhk.edu.hk`, `f.chen@ieee.org`.)

979-8-3315-4393-8/25/$31.00 ©2025 IEEE

object properties from visual observations. An example is shown in Fig. 1. Given the task objective *please twist the loose knobs using the screwdriver*, the planner must analyze the image to determine the number of knobs in the scenario, which dictates how many times the *twist* skill should be executed in the task plan. Overall, our benchmark is characterized as open-world tasks with long-horizon, ungrounded-object, diverse-skill, and multi-level coordination, which provides a comprehensive testbed for bimanual dexterous manipulation.

In addition, we present **OBiMan-Planner**, a VLM-based zero-shot planning framework tailored for bimanual dexterous manipulation in open-world scenarios. Our approach includes two key components: First, a hierarchical grounding module is designed to bridge open-world tasks with specific scenario images. Specifically, the VLM layer employs a VLM within a feedback validation loop to extract task-relevant objects and infer their properties, followed by the rule layer to ground constraints and objectives. Second, a fully VLM-based task planning module is introduced to generate task plans based on grounded scenario information, eliminating the reliance on external solvers. To improve planning performance, we integrate a rule-based skill world model into the validation process, which encapsulates all knowledge about the effect of all skills. This model-based validation helps assess the feasibility of the proposed plan, providing critical feedback to the VLM. In summary, a series of experiments are conducted on OBiMan-Bench, demonstrating the strong capability of our method in addressing open-world bimanual dexterous manipulation tasks.

II. Related Works

A. Bimanual Dexterous Manipulation

Bimanual dexterous manipulation, leveraging coordination between hands, has shown promising results in enhancing stability and efficiency, especially in workspace-constrained and contact-rich task scenarios, such as wiping a plate, twisting knobs, and moving a heavy box [21], [22]. Previous works have established taxonomies for bimanual dexterous manipulation by analyzing the kinematic patterns of both hands from bimanual manipulation datasets [18], [19]. Recently, Grotz et al. [20] extended the taxonomy by considering the force interaction between both hands, determining the task complexity based on motion characteristics in temporal, spatial, and physical coupling, as well as symmetric and synchronous coordination. Drawing insights from them, we develop a skill-oriented bimanual coordination taxonomy and provide a large-scale benchmark called OBiMan-Bench, tailored for open-world bimanual dexterous manipulation planning.

B. Open-world Task Planning in Robotics

Recent advances in large language models (LLM) and vision-language models (VLM) demonstrate remarkable capability in logical reasoning with common sense knowledge [23], spurring the development of numerous LLM-based [15], [16], [13], [14] and VLM-based planners [8], [9], [10], [11], [12]. Several studies [13], [11], [14] have integrated them with PDDL [3] to automatically generate task domains, reducing the need for manual definition in classical predicate logic-based planning. For example, Yang et al. [11] proposed a hierarchical framework that leverages VLMs to decompose task instructions into valid sub-goals expressed in the PDDL format based on labeled observation images. Despite the progress, such methods remain constrained by their reliance on predefined scenario descriptions and/or precise semantic segmentation, limiting their adaptability in open-world scenarios. Recently, Wang et al. [12] proposed a VLM-based zero-shot task planner for bimanual manipulation. However, they only adopted low-level coordinate skill primitives such as pick and place. OBiMan-Planner, in contrast, introduces a zero-shot framework that fully leverages VLMs without external solvers, excelling in generating plans based on 18 bimanual dexterous skill primitives in open-world scenarios.

III. Method

A. OBiMan-Bench

We create a large-scale benchmark named OBiMan-Bench to evaluate open-world planning performance for bimanual dexterous manipulation, which includes **18** skill primitives commonly used in human bimanual household activities. These skills include *move*, *handover*, *approach*, *hold*, *align*, *grasp*, *place*, *open*, *close*, *shake*, *stir*, *peel*, *pour*, *cut*, *twist*, *mash*, *insert* and *wipe*.

1) Skill-oriented Bimanual Taxonomy: Previous works [18], [19], [20] classified bimanual manipulation by analyzing the kinematic motion patterns of both hands, distinguishing characteristics such as coordinated/uncoordinated, coupled/uncoupled, and symmetric/asymmetric. While these characteristics offer critical insights for identifying the coordination level of tasks, they are not sufficiently intuitive or evident to facilitate direct comparisons. To address this limitation, we develop a skill-oriented bimanual coordination taxonomy.

Specifically, according to the movement characteristics of both hands in a skill, we categorize bimanual manipulation into two patterns: Two-Hand Coupling (THC) and One-Hand Combination (OHC). In the THC pattern, both hands are required to perform a skill. For example, *handover* transfers an object between hands, and *move* focuses on fixed-offset synchronous movements like carrying a box or kneading dough. In contrast, the OHC pattern functionally differentiates the dominant hand performing manipulation from the non-dominant hand, which either remains uninvolved or slightly assists by stabilizing the operation object. Based on the patterns, we further classify bimanual coordination into three levels:

- High-level: Skills are in the THC pattern, including *move* and *handover*, which require highly dynamic coordination of hands.
- Middle-level: Skills are in the OHC pattern, and the non-dominated hand is required to assist hold the object for stabilization. We define skills, including *insert*, *cut*,

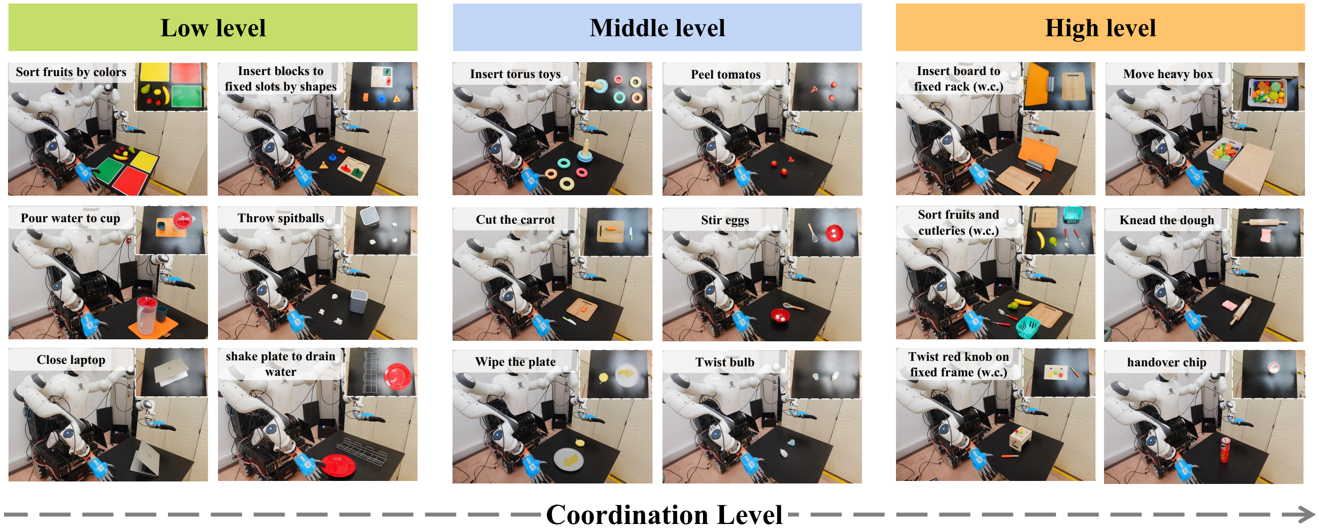

Fig. 2: Example tasks in OBiMan-Bench across multi-level coordination based on the skill-oriented bimanual taxonomy. Manipulating fixed objects or having workspace constraints (w.c.) can shift the coordination level of tasks from high to low or from low to high. OBiMan-Bench validates critical open-world planning capabilities in task-scenario grounding, multi-level coordination handling, and long-horizon bimanual cooperative reasoning.

peel, *stir*, *twist*, *mash*, and *wipe*, as belonging to this level when applying to unfixed objects.

- Low-level: Skills are in the OHC pattern, and the non-dominated hand is free. Typical skills includes *approach*, *hold*, *align*, *grasp*, *place*, *shake*, and *pour*.

Note that, due to the ambiguity in defining the level of collaboration for *open* and *close* in open-world scenarios, these skills in OBiMan-Bench are simplified and categorized as low-level coordination. Based on skill-oriented bimanual coordination taxonomy, the coordination level of a task is determined by the skill at the highest level of all skills necessarily involved in completing the task. Examples of multi-level coordination tasks are shown in Fig. 2.

2) Open-world Tasks: OBiMan-Bench emphasizes the critical capability of task planners to handle open-world tasks, challenging planners in three key aspects:

Task-scenario grounding. Unlike previous works that explicitly defined object properties (*e.g.*, name, type, quantity, and location) [17], we increase the complexity by requiring the task planner to infer these properties directly from the observation images. For example, given the task objective $\mathcal{G}$: *The knobs on the toy frame have come loose, please twist the loose knobs using the screwdriver*, the planner must analyze the image to determine the number of knobs in the scenario, which dictates how many times the *twist* skill should be executed. Besides, because the toy frame is not described as fixed in $\mathcal{G}$, the planners must also capture this information and ask the non-dominated hand cooperatively *hold* the frame.

Multi-level coordination handling. Research [12], [11] on VLM-based or LLM-based task planning primarily employs *grasp*, *place*, *open*, and *close* as their skill primitives to address bimanual manipulation tasks, all of which lie in low-level coordination. Although some works [15] try to enhance coordination by defining the workspace of each robot hand, such as requiring one hand to place an object within a shared area for subsequent manipulation by the other hand, it is still loosely coordinated and not ideal for real-world bimanual manipulation, where *handover* is more commonly used to achieve seamless coordination with higher efficiency. To this end, we expand the skill set with 18 commonly used skill primitives and define various tasks across all three levels of coordination taxonomy. We also incorporate the definition of the workspace to create coordination variants of the same scenario, which requires adaptive planning strategies to select appropriate skills.

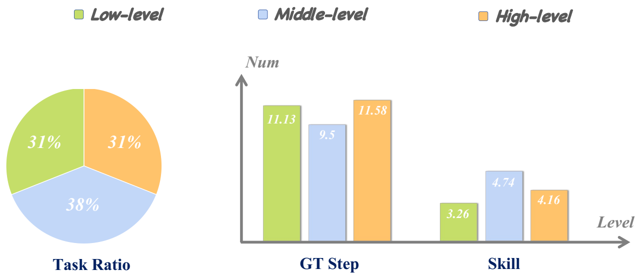

Fig. 3: Statistical charts for OBiMan-Bench. The pie chart represents the task division ratio. The bar charts illustrate the average optimal number of steps (GT) and the required number of skill primitives for task completion.

Long-horizon bimanual cooperative reasoning. OBiMan-Bench defines **100** open-world tasks for evaluating long-horizon bimanual cooperative reasoning, with a breakdown of 31 low-level tasks, 38 middle-level tasks, and 31 high-level tasks. Specifically, each task involves the execution of 2 to 6 skills and requires reasoning over up to 10 object

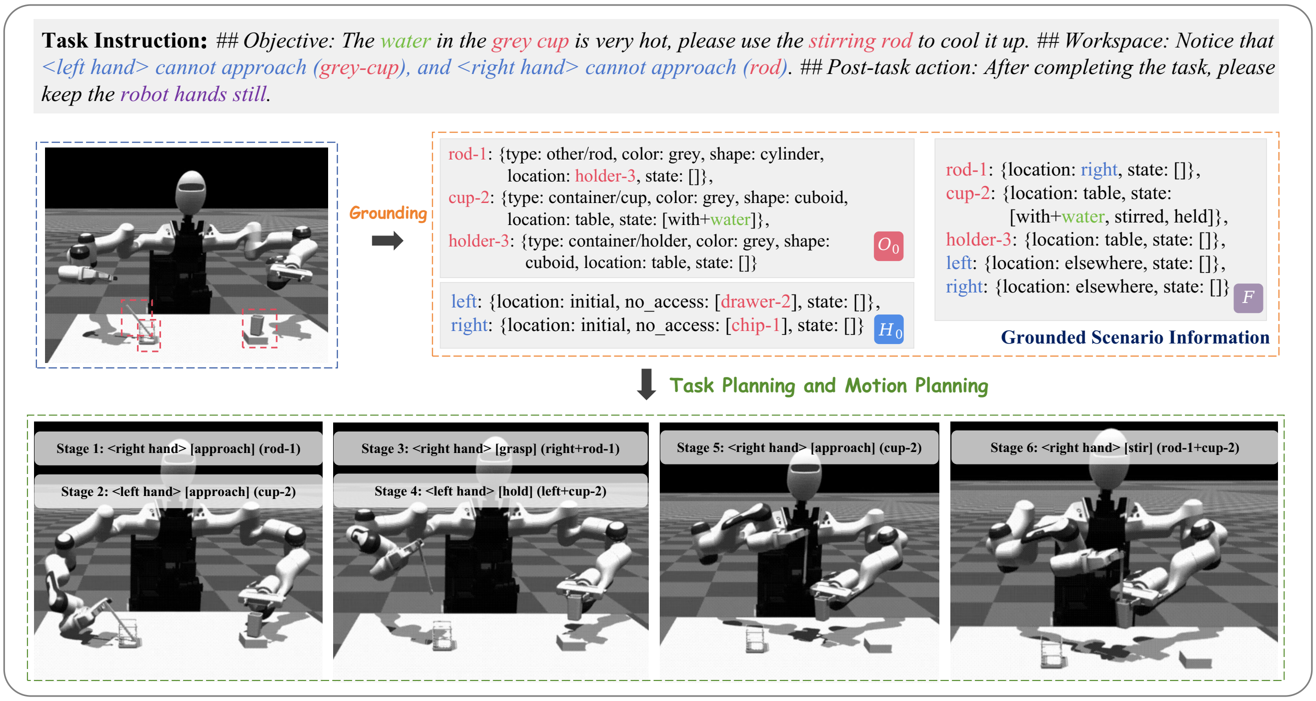
Task Instruction: ## Objective: The water in the grey cup is very hot, please use the stirring rod to cool it up. ## Workspace: Notice that <left hand> cannot approach (grey-cup), and <right hand> cannot approach (rod). ## Post-task action: After completing the task, please keep the robot hands still.
Grounding
rod-1: {type: other/rod, color: grey, shape: cylinder, location: holder-3, state: []},
cup-2: {type: container/cup, color: grey, shape: cuboid, location: table, state: [with+water]},
holder-3: {type: container/holder, color: grey, shape: cuboid, location: table, state: []}
O_0
left: {location: initial, no_access: [drawer-2], state: []},
right: {location: initial, no_access: [chip-1], state: []}
H_0
rod-1: {location: right, state: []},
cup-2: {location: table, state: [with+water, stirred, held]},
holder-3: {location: table, state: []},
left: {location: elsewhere, state: []},
right: {location: elsewhere, state: []}
F
Grounded Scenario Information
Task Planning and Motion Planning
Stage 1: <right hand> [approach] (rod-1)
Stage 2: <left hand> [approach] (cup-2)
Stage 3: <right hand> [grasp] (right+rod-1)
Stage 4: <left hand> [hold] (left+cup-2)
Stage 5: <right hand> [approach] (cup-2)
Stage 6: <right hand> [stir] (rod-1+cup-2)

Fig. 4: Pipeline of OBiMan-Planner in the Task Planning and Motion Planning (TAMP) framework, exemplified with the middle-level task of stirring using a rod. Given an open-world task (top row), OBiMan-Planner first grounds the task instruction with a specific scenario observation, generating grounded scenario information (middle row), including grounded object properties O_0, hand constraints H_0, and final objective dictionary F. Then, OBiMan-Planner proposes a task plan with sequential stages based on VLM, where each stage is executed by a short-horizon skill primitive, whether it is acquired rule-based or learning-based (bottom row).

interactions, with the longest task sequence spanning 28 steps. On average, the optimal number of steps (GT) for task completion is 10.65, and the required number of skill primitives for it is 4.10. More details are illustrated in Fig. 3. The varying task complexities, from low-level to high-level coordination, ensure a comprehensive assessment of open-world planning ability to handle long-horizon, ungrounded-object, multi-skill interactions for bimanual dexterous manipulation.

3) Task Definition: In OBiMan-Bench, each task is defined by a triplet that includes an open-world task instruction and objective dictionary, with a scenario-specific observation image for grounding.

Open-world task instruction is a natural language text, including three components: task objective, workspace, and post-task action.

- Task objective: Describes the overall purpose of the task and follows the format "*## Objective: please*", where the *please* part, as exemplified in $\mathcal{G}$.
- Workspace: Defines the workspace of each hand. It follows the format "*## Workspace: Notice that <h1 hand> can approach all objects/cannot approach (constraint1), (constraint2), . . . and <h2 hand> . . .*". Here, `h1` represents arbitrary hand (left/right), with `h2` being the other.
- Post-task action: indicates the final movements of both hands after the goal is reached, following the format "*## Post-task action: After completing the task, please move the robot hands to the initial poses/keep still*".

Note that, `constraint` defined in the workspace part supports three templates, `object`, `color-object`, or `shape-object`, *e.g.*, knob, red-apple, and triangle-block. This stipulates the no-access list of `h1` at the beginning of tasks, where *approach* skill cannot be employed by `h1` to all objects in the list. However, the list can be dynamically managed as the interlocation among objects changes.

Open-world objective dictionary represents target final status of concerned entities (objects and two robot hands) based on *Objective* and *Post-task action* in the task instruction. The final status of each entity contains two fields: location and state. Specifically, the location field indicates the final position it should be in, for robot hands, it varies from 'initial' to 'elsewhere' based on the post-task action requiring *initial poses* or *fixed*. For *initial poses*, all hands should move to initial poses without objects in their hands, which means they should *place* in-hand objects on the table if applicable. The state field for objects is a list of non-kinematic states (e.g., fixed, with+water, shaken, cut, opened, held, mashed, twisted, wiped, peeled) resulting from the execution of corresponding skills. This dictionary plays an important role in constructing feedback for task planners, indicating whether the planned stage sequence is correct.

The details of all fields are elaborated in the supplementary materials.

Scenario-specific observation is an RGB image, which captures the scenario from the robot view. Three open-world parts in tasks, including keywords of objects, workspace of hands, and the objective dictionary, should be grounded with this image, as shown in Fig. 4. Notably, to support the potential expanded applications of OBiMan-Bench in bimanual manipulation, we additionally provide multi-view observation images for each scenario (robot view, top view, and side view) as shown in Fig. 1, though only the robot view image is utilized for task planning. Meanwhile, the ground truth of grounded scenario information is also provided for ablation. For more details please refer to the supplementary material.

B. OBiMan-Planner

To address bimanual dexterous tasks in open-world scenarios defined in OBiMan-Bench, we introduce OBiMan-Planner, a novel zero-shot planning framework based on vision-language models (VLM). The schematic framework is shown in Fig. 1. Compared to other methods [13], [11], OBiMan-Planner eliminates the reliance on external solvers such as PDDL solver [3], enabling a more flexible and self-contained approach to task planning. Specifically, two key components are comprised: the scenario grounding module and the task planning module.

1) Scenario Grounding Module: This module bridges open-world tasks with specific scenario images, generating grounded object properties O_0, hand constraints H_0, and final objective dictionary F, as shown in Fig. 5. Instead of directly prompting the VLM to output all components at once, we propose a hierarchical pipeline that processes O_0 and other elements, respectively.

Specifically, the VLM layer employs a VLM within a feedback validation loop to generate grounded O_0, extracting task-relevant objects and inferring their property fields, including name, type, color, shape, location, and state, based on the observation image $\mathcal{I}$. Considering the specific image for $\mathcal{G}$ shown in Fig. 1, the task-relevant objects `knobs`, `frame`, `screwdriver` can be grounded as follows:

> toy frame-1: {type: toy/frame, color: white, shape: cuboid, location: table, state: []},
> knob-2: {type: toy/knob, color: yellow, shape: cylinder, location: toy frame-1, state: []},
> knob-3: {type: toy/knob, color: green, shape: cylinder, location: toy frame-1, state: []},
> knob-4: {type: toy/knob, color: red, shape: cylinder, location: toy frame-1, state: []},
> screwdriver-5: {type: tool/screwdriver, color: orange, shape: irregular, location: table, state: []}

The process begins with an initial grounding prompt composed of the task instruction, the image I, and important notes for legally defining the property fields of objects. Then, the VLM processes this prompt and generates a proposal for O_0. A validation block is conducted to verify the format and field legality of the proposed O_0. If any check fails, the failure feedback is appended to the dialogue history, guiding the VLM to refine its proposal. This loop keeps iterating until the legal proposal is generated or the maximum number of iterations is exceeded.

Note that the legal proposal does not ensure to be consistent with the ground truth of O_0, which means it may be inaccurate for subsequent processes. The reasons are mainly from two aspects: First, the output of VLM is uncertain, which is controlled by the temperature factor. Second, not all fields can be properly validated due to the missing ground truth during open-world task planning, which results in some key states, such as fixed or closed, that may not be identified in O_0. This effect will be discussed in Sec. IV.

Based on the grounded O_0, the rule layer takes the task instruction and objective dictionary as inputs and applies rule-based grounding to derive grounded hand constraints H_0 and final objective dictionary F. Specifically, it replaces the open-world `object`, `color-object`, or `shape-object` with corresponding grounded object names in O_0 that match these patterns. For example, `knobs` may be replaced with `knob-2`, `knob-3` and `knob-4`.

2) Task Planning Module: Given the grounding scenario information O_0, H_0, and F, the task planning module is utilized to derive a long-horizon cooperative plan $\mathcal{S}$, ensuring that the desired location and additional states in F can be fulfilled based on O_0 and H_0.

Formally, the task plan contains sequential stages, where each stage A_i clarifies which hand or both hands use which skill to manipulate which object. According to the skill primitives, four stage formats are defined:

1. `<h1 hand> [approach] (app_target_name)`
2. `<h1 hand> [dex_skill] (tool_name+dex_target_name)`
3. `<both hands> [move] (obj_inhand_name+move_target_name)`
4. `<both hands> [handover] (active_name+passive_name)`

where `dex_skill` is in the set of skills in OHC except *approach*. Note that for the second format, only one exception skill is *twist*, whose format adopts `(tool_name+dex_base_name+dex_target_name)`, where `tool_name+dex` is in the set of `h1` and all object names in O_0, `dex_base_name` and `dex_target_name` are in the set of all object names in O_0. For example, "*left hand using left hand [twist] lamp bulb-2 into lamp holder-4*" should be `<left hand> [twist] (left+lamp bulb-2+lamp holder-4)`. Similarly, "*right hand using screwdriver-5 [twist] knob-3 into toy frame-1*" should be `<right hand> [twist] (screwdriver-5+knob-3+toy frame-1)`. More examples for other cases please refer to the supplementary material.

Our method adopts the centralized framework and proposes a complete plan proposal at once based on VLM. Specifically, the initial planning prompt for this module

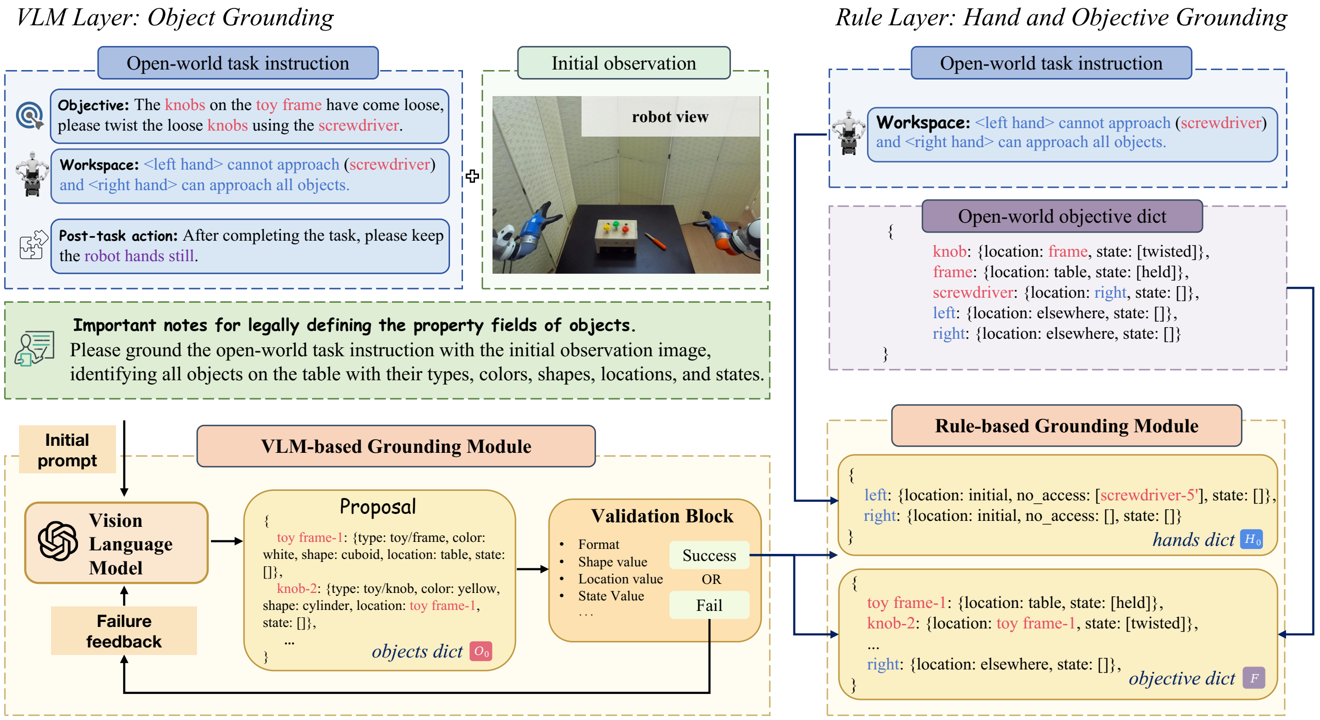
VLM Layer: Object Grounding
Open-world task instruction
Objective: The knobs on the toy frame have come loose, please twist the loose knobs using the screwdriver.
Workspace: <left hand> cannot approach (screwdriver) and <right hand> can approach all objects.
Post-task action: After completing the task, please keep the robot hands still.
Initial observation
robot view
Important notes for legally defining the property fields of objects.
Please ground the open-world task instruction with the initial observation image, identifying all objects on the table with their types, colors, shapes, locations, and states.
Initial prompt
VLM-based Grounding Module
Vision Language Model
Failure feedback
Proposal
{
toy frame-1: {type: toy/frame, color: white, shape: cuboid, location: table, state: []},
knob-2: {type: toy/knob, color: yellow, shape: cylinder, location: toy frame-1, state: []},
...
}
objects dict O0
Validation Block
Format
Shape value
Location value
State Value
...
Success
OR
Fail
Rule Layer: Hand and Objective Grounding
Open-world task instruction
Workspace: <left hand> cannot approach (screwdriver) and <right hand> can approach all objects.
Open-world objective dict
{
knob: {location: frame, state: [twisted]},
frame: {location: table, state: [held]},
screwdriver: {location: right, state: []},
left: {location: elsewhere, state: []},
right: {location: elsewhere, state: []}
}
Rule-based Grounding Module
{
left: {location: initial, no_access: [screwdriver-5'], state: []},
right: {location: initial, no_access: [], state: []}
}
hands dict H0
{
toy frame-1: {location: table, state: [held]},
knob-2: {location: toy frame-1, state: [twisted]},
...
right: {location: elsewhere, state: []},
}
objective dict F

Fig. 5: Illustration of the hierarchical scenario grounding module in OBiMan-Planner. Given open-world task instructions and specific scenario observations, the VLM layer first employs VLM within a feedback validation loop to generate the grounded objects dictionary (dict). If any validation check fails, the failure feedback is append to the prompt for the next refinement. Then, the legal objects dictionary is utilized for rule-based hand and objective grounding in the rule layer.

incorporates the grounded scenario information (O_0, H_0, F), the image $\mathcal{I}$, and essential notes for skill primitives and stage generation. Similarly to the high-level layer of the grounding module, this module also employs the VLM as the proposer within a feedback validation loop to generate the task plan $\mathcal{S}$. Beyond the basic format checking in the validation block, it integrates a skill world model manually crafted by rules, which encapsulates all knowledge about the effect of all skills on the location and state fields. For example, for stage `<right hand> [twist] (screwdriver-5+knob-3+toy frame-1)`, the preconditions of *twist* skill include `screwdriver-5` is in `right hand`, `right hand` has approached `toy frame-1`, and `toy frame-1` is held if it is not fixed. Given $\mathcal{S} = \{A_1, A_2, \cdots, A_T\}$ as actions, O_0 and H_0 as the initial state, F as target state, this model-based validator iteratively simulates each A_i to assess its feasibility and verify whether the final updated O_T and H_T align with F, as shown in Fig. 6. If any failure is detected, it returns failure feedback to the proposer for the next refined plan. This model-based validator is demonstrated to greatly improve planning performance.

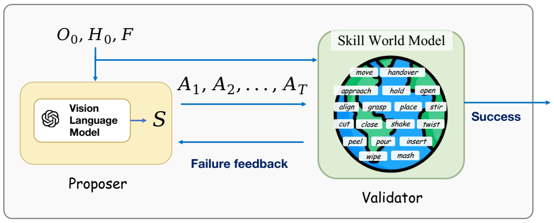

Fig. 6: Illustration of the validation process of the task planning module in OBiMan-Planner. Given grounded scenario information, the VLM proposer generates a complete task plan, validating with a skill world model.

IV. Experiments

We design a series of experiments on OBiMan-Bench to validate our task planner OBiMan-Planner. In this section, we try to answer the following key questions: 1) How effective is the planning performance to handle long-horizon bimanual dexterous manipulation tasks with multi-skill and multi-level coordination in open-world scenarios? 2) What are the enablers and bottlenecks for open-world bimanual dexterous manipulation planning? 3) How can the planner help in developing open-world bimanual manipulation policies?

A. Experiment Setting

We utilize GPT-4o [24] as the VLM for task planning. For each task in OBiMan-Bench, the optimal number of steps (GT) for completion is manually counted. Each planning loop is allowed to receive feedback up to 15 times, and the total step number in the proposed task plan $\mathcal{S}$ should be below twice GT. If any condition is violated, the iteration

TABLE I: Quantitative comparison of different methods on OBiMan-Bench.

Method	Low-level		Middle-level		High-level		Average	
	SR↑	AS↓	SR↑	AS↓	SR↑	AS↓	SR↑	AS↓
Grounded Info w/o Image	0.900	13.516	0.974	10.105	0.903	14.258	0.930	12.450
Grounded Info w/o Feedback	0.800	14.484	0.368	16.632	0.533	21.839	0.480	17.580
OBiMan-Planner (Ours)	0.900	13.258	0.974	9.921	1.000	12.387	0.960	11.720
GT Info w Image	1.000	12.032	1.000	9.605	1.000	12.161	1.000	11.150
GT Info w/o Image	1.000	12.548	1.000	9.684	1.000	13.516	1.000	11.760
Ground Truth (GT)	-	11.129	-	9.500	-	11.581	-	10.650

breaks, then this planning is treated as a failure. Otherwise, if the final updated O_T and H_T based on S match F, the iteration ends with success. For all experimental methods, they ran all tasks in OBiMan-Bench with the same temperature factor of 0.2 and as similar prompts as possible for GPT-4o, ensuring a fair comparison.

B. Compared methods

To answer the second question, we ablate OBiMan-Planner into four variants:

- **Grounded Info w/o Image:** This variant only tasks the grounded scenario information and essential notes as the initial planning prompt, without the image I.
- **Grounded Info w/o Feedback:** This variant only receives the format check of its validation block without the model-based feasibility assessment for each stage. If the format check passes, it ends the iteration and determines the success or failure.
- **GT Info w Image:** This variant takes the ground truth of grounded scenario information, the image, and essential notes as prompt, eliminating planning failures from inaccurate O_0, H_0, and F.
- **GT Info w/o Image:** This variant takes the ground truth of grounded scenario information and essential notes as prompt, eliminating planning failures from inaccurate O_0, H_0, and F.

For this purpose, the first two variants are designed to explore the enablers for OBiMan-Planner in addressing task planning for bimanual dexterous manipulation, while the last two variants are used to investigate the scenario grounding capability and bottlenecks in open-world task planning. OBiMan-Planner is the one 'Grounded Info w Image w Feedback'.

C. Evaluation Metric

We adopt Success Rate (*SR*) and Average Step (*AS*) as the evaluation metrics. For *SR*, we count the total number of successful planning tasks and divide by the total number of tasks $N = 100$. For *AS*, we summarize the total step number of all tasks and divide by N, where the step number is counted as twice *GT* plus one for the failed tasks for uniform measurement. As such, higher *SR* and lower *AS* are preferred.

D. Results

The quantitative results are reported in Table I. By analyzing the results, the answers to the key questions are discussed as follows:

1) Planning performance on open-world tasks for bimanual dexterous manipulation: Based on the experimental results of *OBiMan-Planner*, our method demonstrates promising performance in tacking long-horizon and high-level coordination tasks, achieving the success rate (*SR*) of **0.96** among 100 complex open-world planning tasks for bimanual dexterous manipulation. Additionally, the average step (*AS*) required for task completion is slightly higher than the ground truth (GT) value, indicating that our method is capable of generating efficient plans for long-horizon tasks.

2) Enablers and bottlenecks for open-world planning: To investigate the enablers and bottlenecks for open-world planning, we compare the results between *OBiMan-Planner* and its variants, which disentangle potential factors influencing the performance. Several insights are discovered.

Specifically, the performance of the variants *Grounded Info w/o Image* and *Grounded Info w/o Feedback* is inferior to our method on all metrics across all coordination levels, which uncovers the enabling factors for *OBiMan-Planner*. First, including the observation image in the initial planning prompt provides essential visual context for the VLM, improving its reasoning performance. This conclusion is further supported by the comparison between *GT Info w Image* and *GT Info w/o Image*, where the former shows lower *AS*, indicating higher efficiency of task plans. However, this conclusion is neglected by previous works. Second, the variant of *Grounded Info w/o Feedback* shows a significant performance drop compared to *OBiMan-Planner*, which highlights the critical role of the introduced skill world model by providing model-based feedback.

On the other hand, both *GT Info w Image* and *GT Info w/o Image* variants achieve *SR=1* across all coordination levels, validating the effectiveness of our task planning module. However, this also reveals that the primary bottleneck in *OBiMan-Planner* lies in the scenario grounding module, where inaccuracies in the generated grounded scenario information lead to performance degradation. Specifically, two primary issues are identified through failure analysis: First, the names of objects in O_0 cannot match with the open-world hand constraints or objective dictionary, causing failures in the rule layer of the scenario grounding module. For example, the instruction mentions 'knife' while the grounding module named the object after 'cutlery'. Second, inaccurate object properties in O_0, such as location and state, may result in unnecessary stages in the plan or infeasibility. For example, it failed to detect that a board is already on the shelf but placed it again, which results in the total step number exceeding the prescribed one. Another case is presented in a task with constraints, where an object is detected as an incorrect color, causing both hands to be unable to access it.

We believe these findings reveal critical support for designing open-world task planners based on VLMs and show future directions to enhance planning performance.

3) Integrating OBiMan-Planner in developing open-world bimanual manipulation policies: Fig. 4 illustrates the execution pipeline of adopting OBiMan-Planner in the Task Planning and Motion Planning (TAMP) framework. Leveraging the capability of *OBiMan-Planner* in open-world scenario grounding and task planning, long-horizon task instructions are decomposed into sequential stages, where each stage is executed by a short-horizon skill primitive, whether it is acquired rule-based or learning-based. In future work, we aim to integrate OBiMan-Planner in a closed-loop policy learning framework, enabling real-time adaptation and continuous improvement of bimanual dexterous manipulation strategies in dynamic open-world environments.

V. CONCLUSIONS

In this paper, we present *OBiMan-Bench*, a large-scale benchmark specifically designed to evaluate the performance of open-world task planners for bimanual dexterous manipulation. We also propose *OBiMan-Planner*, a novel vision-language model (VLM)-based zero-shot planning framework, which incorporates two key components to address the challenges of task-scenario grounding and long-horizon coordination reasoning. Extensive experiments on OBiMan-Bench demonstrate the effectiveness of our method in tackling complex open-world manipulation tasks. This work provides a comprehensive testbed for open-world bimanual manipulation planning and sets a foundation for developing bimanual dexterous manipulation policies in dynamic open-world environments.

REFERENCES

[1] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *Journal of machine learning research*, vol. 22, no. 30, pp. 1–82, 2021.

[2] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, no. 1, pp. 265–293, 2021.

[3] C. Aeronautiques, A. Howe, C. Knoblock, I. D. McDermott, A. Ram, M. Veloso, D. Weld, D. W. SRI, A. Barrett, D. Christianson *et al.*, "Pddl— the planning domain definition language," *Technical Report, Tech. Rep.*, 1998.

[4] D. L. Kovacs *et al.*, "A multi-agent extension of pddl3. 1," in *ICAPS 2012 Proceedings of the 3rd Workshop on the International Planning Competition (WS-IPC 2012)*, 2012, pp. 19–37.

[5] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen *et al.*, "Simple open-vocabulary object detection," in *European conference on computer vision*. Springer, 2022, pp. 728–755.

[6] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[7] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European Conference on Computer Vision*. Springer, 2024, pp. 38–55.

[8] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "Vima: General robot manipulation with multimodal prompts," in *Fortieth International Conference on Machine Learning*, 2023.

[9] brian ichter, A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, D. Kalashnikov, S. Levine, Y. Lu, C. Parada, K. Rao, P. Sermanet, A. T. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, M. Yan, N. Brown, M. Ahn, O. Cortes, N. Sievers, C. Tan, S. Xu, D. Reyes, J. Rettinghouse, J. Quiambao, P. Pastor, L. Luu, K.-H. Lee, Y. Kuang, S. Jesmonth, K. Jeffrey, R. J. Ruano, J. Hsu, K. Gopalakrishnan, B. David, A. Zeng, and C. K. Fu, "Do as i can, not as i say: Grounding language in robotic affordances," in *6th Annual Conference on Robot Learning*, 2022.

[10] 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 *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, 2023, pp. 8469–8488.

[11] Z. Yang, C. Garrett, D. Fox, T. Lozano-Pérez, and L. P. Kaelbling, "Guiding long-horizon task and motion planning with vision language models," *IEEE International Conference on Robotics and Automation*, 2025.

[12] Z. Wang, R. Shen, and B. C. Stadie, "Wonderful team: Zero-shot physical task planning with visual llms," *Transactions on Machine Learning Research*, 2024.

[13] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "Llm+ p: Empowering large language models with optimal planning proficiency," *CoRR*, vol. abs/2304.11477, 2023.

[14] G. Chen, L. Yang, R. Jia, Z. Hu, Y. Chen, W. Zhang, W. Wang, and J. Pan, "Language-augmented symbolic planner for open-world task planning," in *Robotics: Science and Systems*, 2024.

[15] Z. Mandi, S. Jain, and S. Song, "Roco: Dialectic multi-robot collaboration with large language models," in *IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 286–299.

[16] K. Liu, Z. Tang, D. Wang, Z. Wang, B. Zhao, and X. Li, "Coherent: Collaboration of heterogeneous multi-robot system with large language models," *IEEE International Conference on Robotics and Automation*, 2025.

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

[18] F. Krebs and T. Asfour, "A bimanual manipulation taxonomy," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 031–11 038, 2022.

[19] D. Rakita, B. Mutlu, M. Gleicher, and L. M. Hiatt, "Shared control–based bimanual robot manipulation," *Science Robotics*, vol. 4, no. 30, p. eaaw0955, 2019.

[20] M. Grotz, M. Shridhar, Y.-W. Chao, T. Asfour, and D. Fox, "Peract2: Benchmarking and learning for robotic bimanual manipulation tasks," in *CoRL 2024 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*, 2024.

[21] V. Girbes-Juan, V. Schettino, Y. Demiris, and J. Tornero, "Haptic and visual feedback assistance for dual-arm robot teleoperation in surface conditioning tasks," *IEEE Transactions on Haptics*, vol. 14, no. 1, pp. 44–56, 2020.

[22] Y. Ren, Z. Zhou, Z. Xu, Y. Yang, G. Zhai, M. Leibold, F. Ni, Z. Zhang, M. Buss, and Y. Zheng, "Enabling versatility and dexterity of the dual-arm manipulators: A general framework toward universal cooperative manipulation," *IEEE Transactions on Robotics*, 2024.

[23] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *The International Journal of Robotics Research*, p. 02783649241281508, 2023.

[24] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, "Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration," *IEEE Robotics and Automation Letters*, 2024.