OWMM-Agent: Open World Mobile Manipulation With Multi-modal Agentic Data Synthesis

Anonymous Author(s)
Affiliation
Address
email

Abstract

The rapid progress of navigation, manipulation, and vision models has made mobile manipulators capable in many specialized tasks. However, the open-world mobile manipulation (OWMM) task remains a challenge due to the need for generalization to open-ended instructions and environments, as well as the systematic complexity to integrate high-level decision making with low-level robot control based on both global scene understanding and current agent state. To address this complexity, we propose a novel multi-modal agent architecture that maintains multi-view scene frames and agent states for decision-making and controls the robot by function calling. A second challenge is the hallucination from domain shift. To enhance the agent performance, we further introduce an agentic data synthesis pipeline for the OWMM task to adapt the VLM model to our task domain with instruction fine-tuning. We highlight our fine-tuned OWMM-VLM as the first dedicated foundation model for mobile manipulators with global scene understanding, robot state tracking, and multi-modal action generation in a unified model. Through experiments, we demonstrate that our model achieves SOTA performance compared to other foundation models including GPT-4o and strong zero-shot generalization in real world. The project page is at `https://owmm-vlm-project.github.io`.

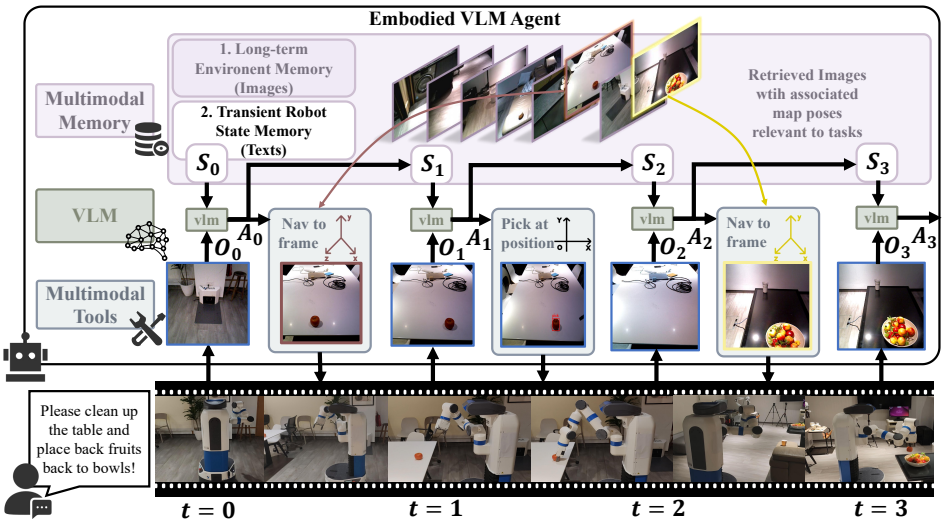


Figure 1: **OWMM-Agent Operates Fetch Robot for Tidying Task.** *OWMM-Agent* receives natural language instructions and leverages both long-term environment memory (scene images) and transient robot state memory (textual summary) to generate sequential multi-modal actions to finish the task. By **multi-turn, multi-image, and multi-modal** VLM reasoning, the agent conducts global scene aware reasoning, updates robot state memory, and actuates itself to desired coordinates without any other learning-based models in unstructured environments.

1 Introduction

The vision of generalist home assistant robots has brought open-world mobile manipulation (OWMM) to the forefront of embodied AI research [38, 34, 37, 42, 26]. OWMM tasks require mobile manipulators to interpret open-ended natural language instructions and operate in unstructured, previously unseen environments. Although advancements in navigation, manipulation, and vision models have effectively enabled mobile manipulators to perform many specialized tasks under constraints, achieving robust autonomy in these settings remains challenging.

A central difficulty in OWMM is the need for comprehensive global scene understanding and reasoning conditioned on natural language instructions and agent state. On one hand, prior approaches often construct 2D semantic maps [28] or 3D semantic fields with CLIP-based features [19, 26], retrieving targets by computing embedding distances between the semantic map and language instructions. While these methods have enabled progress, they are limited by the capacity of embedding models, which can struggle with complex, compositional instructions, compared to foundational generative models like large language models (LLM) or vision-language models (VLM). Additionally, they often require time-consuming dense 3D reconstruction, making them less suitable for complex, open-ended and dynamic environments. On the other hand, the recent advances in LLMs and VLMs, with strong generalization capability, versatility, and reasoning capability, offer promising opportunities and potentially a fundamental pathway to solve all sorts of scene understanding, task planning, and robot control issues in open-world intelligent robot systems [20, 15].

Based on the aforementioned observations, we propose a novel VLM agent framework, *OWMM-Agent*, to address these challenges and leverage the power of VLMs for OWMM task. More specifically, we formulate the high-level OWMM task for the internal VLM model as a multi-turn, multi-image, and multi-modal reasoning problem. The VLM model generates end-to-end chain-of-thought (CoT) thinking process, tracked agent states, and multi-modal actions with coordinates based on all raw multi-modal input. Then the agent calls the coordinate-based planners to actuate the robot. Our approach is built on two insights: 1) We do not need the detailed geometric representation of the environment for instruction-conditioned global scene understanding, and we could easily access precise and even dynamic geometric information when the robot moves to the task-relevant local region. 2) By leveraging the strong vision-language grounding capabilities, we can effectively bridge the high-level reasoning process in language and low-level robot control targets in coordinates, with the help of 2D-to-3D reverse projection.

However, directly applying pre-trained VLMs to our embodied agent presents challenges of domain shift: 1) **Rare grounding tasks**: Robotic planners and controllers require multi-modal inputs, including both tools and coordinates in the visual space for robot control. The base models could be powerful for object-centric grounding such as detecting novel objects, but they suffer in other grounding tasks including detecting non-blocked navigable areas in the ego-centric image. 2) **State tracking**: The agent must infer and track its own state from observations and history records to make contextually appropriate decisions. 3) **Embodiment priors**: Effective decision-making in egocentric settings demands strong embodiment-dependent priors, such as knowledge of the robot's kinematic constraints, such as maximum reach for picking actions.

To address the problem of domain adaptation, we further introduce an agentic data synthesis pipeline tailored for OWMM, to generate large-scale and instruction-driven episodes that teach the VLM agent to track its state, reason over multi-view observations, and generate multi-modal action affordances grounded in both the global scene and the agent's embodiment. This pipeline minimizes human annotation effort by utilizing predefined task sequence templates and ground-truth symbolic world representations from simulation. With extensive experiments in simulation, we demonstrate that OWMM-VLM consistently outperforms baseline models. In the real-world experiment, we find that our model has strong zero-shot generalization to real-world observations, with $27/30 = 90\%$ action generation success rate on our fetch robot in the lab environment, even being fine-tuned on the simulated data. We also provide ablation studies on models and dataset analysis to provide insights into the model design and training data construction. In summary, our contributions are as follows:

- We propose *OWMM-Agent*, a unified VLM-based agent architecture for open-world mobile manipulation, capable of global scene understanding, state tracking, and end-to-end action generation.
- We introduce a simulation-based agentic data synthesis pipeline that enables scalable data collection for instruction fine-tuning for domain adaptation with minimized human effort, with detailed analysis on the quality of the generated dataset.

- We introduce a foundation model for OWMM, capable of multi-image reasoning and executable multi-modal action generation, with extensive experiments analyzing the model's performance.

2 Related Works

Open World Mobile Manipulation

Open-vocabulary Mobile Manipulation (OVMM) focuses on navigating and manipulating novel objects in unseen environments with language instructions. Referred to as *Open Vocabulary Mobile Manipulation* (OVMM) by[37, 42, 19] or *Open World Mobile Manipulation* (OWMM) by[27, 38, 34], we use the term OWMM for this paper.

The original OWMM baseline and Melnik et al. [23] assume the agent starts without scene observation and must explore to build a representation for decision-making. Recent works Liu et al. [19], Qiu et al. [26], Zhi et al. [42] suggest a two-stage approach: first using SLAM [7] to create 3D semantic maps, then performing OVMM using open-vocabulary models like GPT-4V and GPT-4o [11]. Zhi et al. [42] introduces *COME-robot*, a closed-loop OVMM framework using GPT-4V for reasoning and replanning, producing code for preset functions and object captions as in *Code-as-Policy* [17]. Unlike relying on pre-trained skill models requiring inputs like skill names and object captions, our model directly produces target positions for position-based motion planners and controllers.

Large Foundational Models for Robotics

Recent advances in large fundamental models show significant potential in robotic control and generalization. One major research focus is to adapt pre-trained Visual Language Model (VLM) to robot scenario. RoboPoint [39] introduces a synthetic data pipeline for instruction-tuning VLMs in robotics, supporting accurate spatial affordance prediction in object manipulation and navigation. MOKA [18] uses a novel VLM approach in robotic manipulation with point-based affordance and motion representation, using visual prompts to turn key points and waypoint predictions into visual question-answering tasks for VLMs. Our proposed model *OWMM-VLM* also falls into this category.

The other popular research topic is Vision-Language-Action (VLA) models, focusing on using relatively smaller transformer backbones to directly generate robotic actions with high freqeuncy. OpenVLA [14] is a 7B-parameter open-source model trained on 970,000 real-world demonstrations using Llama 2 [32] architecture, excelling in general manipulation tasks. Octo [31] advances generalist robot policies, handling language commands and goal images while adapting quickly to new inputs and actions with standard GPUs. π_0 [2] presents a flow-matching architecture based on a pre-trained VLM, excelling in dexterous tasks. These models mark significant progress in making robotic systems more versatile, scalable, and adaptable to different trajectories.

Embodied LLM/VLM Agents and Post-training Adaptation

Large Language Models (LLMs) and Vision-Language Models (VLMs) have achieved remarkable success across various agent applications. In web navigation, agents like AutoWebGLM [16] leverage curriculum learning and reinforcement learning to surpass GPT-4 on realistic browsing tasks. Similarly, AppAgent [40] and AssistGUI [9] successfully adapt multimodal LLMs to effectively interact with smartphone and desktop environments, enabling complex GUI manipulation through imitation learning. In embodied AI, foundational models have recently begun to be integrated into physical or simulated robotic agents. Approaches like Steve-Eye [41] provide integrated multimodal perception and planning capabilities for open-world tasks, while TANGO [44] demonstrates that pretrained LLMs, combined with basic robot primitives, can solve diverse embodied tasks without task-specific fine-tuning.

However, direct application of foundational models in embodied settings often causes hallucinations and grounding issues. To address this, recent works introduce specialized post-training methods to adapt models to embodied domains. For instance, KNOWAGENT [43] employs external knowledge bases to constrain model-generated plans, significantly reducing unrealistic outputs. Factually Augmented RLHF [30] utilizes reinforcement learning from human feedback enhanced by factual grounding to align model outputs with reality. Similarly, AdaVIB [1] incorporates adaptive information bottlenecks to suppress irrelevant visual features, thereby mitigating visual hallucinations and improving task accuracy.

3 Methodology

In this section, we introduce the definition of OWMM in section 3.1. After that, we elaborate on the agent framework in section 3.2. Finally, the training method of OWMM-VLM is presented in section 3.3. The overview of our method is shown in Figure 2.

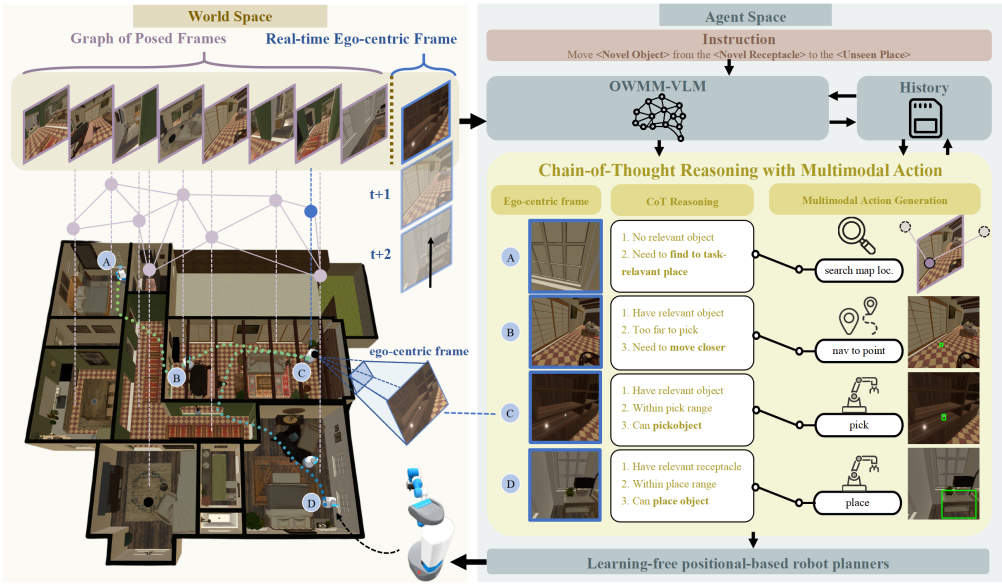


Figure 2: **The Overview of OWMM Agent Framework.** The left panel represents the world space, including a graph of posed frames generated during the pre-mapping phase and a real-time egocentric frame captured by the robot. The right panel showcases the Agent Space, where OWMM-VLM processes task instructions, robot history, and visual inputs to perform chain-of-thought reasoning and generate high-level actions with region coordinates, which are then sent to robot planners for navigation and manipulation.

3.1 OWMM Task Definition

Following the common OVMM/OWMM problem setting [37, 19, 34], the robot needs to follow the instruction in the pattern of `"Move ⟨A⟩(in ⟨B⟩) and place it on/in ⟨C⟩"`, where `⟨A⟩⟨B⟩⟨C⟩` are novel objects/initial receptacles/goal receptacles in the unseen environment from the training data. Following the problem setting in [19, 26], we assume a pre-mapping phase separating active exploration and the SLAM module from the OWMM task focus. This is practical, as most robotic vacuums automate room mapping before cleaning.

Thus, we introduce a pose graph G and associated RGB images $\mathcal{I}$ as the output of the pre-mapping stage on the basis of [37], and define our OWMM problem as follows: In an OWMM task episode of max timestep T, at each timestep $t, 0 \leq t \leq T$, an agent takes inputs composed of 1) a natural language instruction $\mathcal{L}$; 2) a pre-mapping camera pose graph $G = \{V, E\}$ of n poses, where $V = v_0, \ldots, v_n$ edges are not used; 3) and associated RGB images $\mathcal{I} = \{I_0, \ldots, I_n\}$, each image $I_i \in \mathbb{R}^{3\times w\times h}$ are taken at head camera view pose v_i in G; The pre-mapping camera 4) the agent's current head camera RGB image I_t^c and depth image D_t^c.

With these inputs, an agent needs to generate a low-level continuous action a_t that directly actuates the robot kinematically, including joint velocities of the robot arm and the base velocity of the robot. Let's F_{agent} note the logical function of the agent policy model, and we have

$$\mathbf{a}_t = F_{agent}(\mathcal{L}, G, \mathcal{I}, I_t^c, D_t^c, \mathbf{x}_t), \tag{1}$$

where $\mathbf{x}_t$ stands for the robot state at time t.

3.2 OWMM Agent

Running large VLM models at 25Hz and gathering sufficient data for training a generalist VLA model from open-set language and visual observations remain challenging. To address this latency issue in the OWMM agent, we have the large VLM to produce high-level actions. The agent employs a unified VLM model F_{vlm} to convert visual and lingual inputs into action types and positional commands, using a classical planner for navigation and a motion planner for manipulation, similar to Rekep[10]. The model's output represents a high-level action A_t spanning several simulation steps, while planners resolve trajectories and low-level actions $\mathbf{a}_t$ for each step.

$$A_t, \mathcal{H}_t = F_{vlm}(\mathcal{L}, G, \mathcal{I}, I_t^c, \mathcal{H}_{t-1}), \tag{2}$$

$$\mathbf{a}_t = A_t(\mathbf{x}_t, D_t^c), \tag{3}$$

where $\mathcal{H}_t, \mathcal{H}_t$ are the high-level robot history, updated by the VLM model by itelf. $a_t = A_t(\mathbf{x}_t, D_t^c)$ indicates that the high-level action itself can be converted to executable code with the action handle linked to different planners and positional targets. In this regard, part of the high-level action A_t can be seen as a special type of language model program, as proposed in [17]. Then the linked planner takes the state of the robot $\mathbf{x}_t$, and point clouds converted from depth map D_t^c as an additional input to calculate the low-level action a_t.

To translate high-level action A_t into low-level action a_t, the agent has a path planner [8] for navigation and a motion planner [29] for arm manipulation. These planners generate waypoints that satisfy mechanical constraints for base chassis and arm joints through sampling-based methods. There is also a gripper controller to grasp/ungrasp the object. The high-level actions that aim to actuate the robot will be associated with planners and controllers through predefined functions.

3.3 OWMM-VLM model

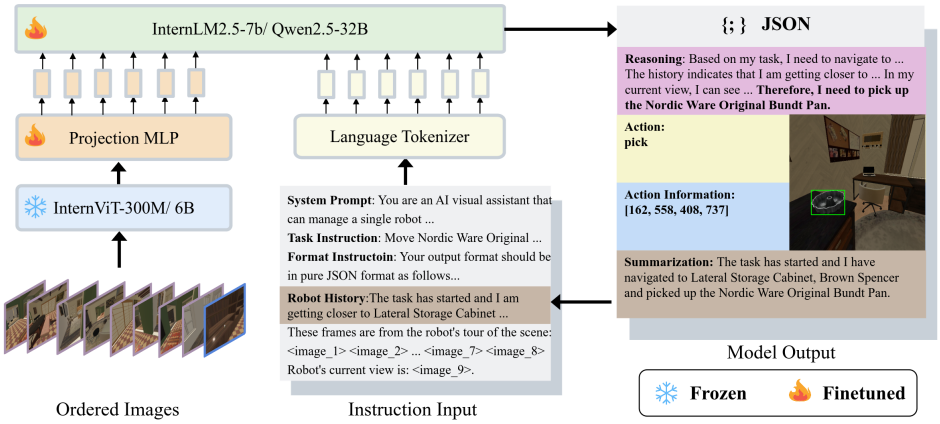
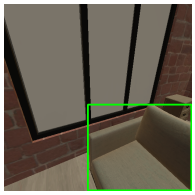
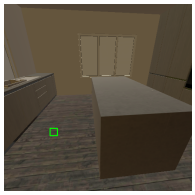
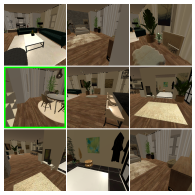


Figure 3: **Overview of OWMM-VLM.** Our model is fine-tuned on InternVL-2.5[5], comprising a ViT, a 2-layer projection MLP, and a LLM. During training, ViT parameters are frozen while the projection MLP and the LLM parameters are trainable. The model is required to generate multi-modal actions in JSON format conditioned on scene images, task instructions, and robot history.

Intuitively, a VLM model requires three core multi-modal capabilities to accomplish the OWMM task: (1) **Image Retrieval.** Given the graph of posed frames and an egocentric frame, the VLM model needs to retrieve a posed frame that contains the relative objects or receptacles that the robot needs to navigate to. (2) **Ego-centric Decision-making.** Given multiple posed frames and an egocentric frame, the VLM model needs to decide which action to conduct based on the task context, robot history, and current egocentric observation. This capability is closely associated with the idea of spatial intelligence[36], that VLM models should understand the spatial relationship between themselves and the scene objects in order to make decisions on actions. (3) **Affordance Grounding.** If the agent decides to interact with the near surroundings perceived in the egocentric frame, it should also generate the target positions that correspond to the intention of the task.

Following this insight, we train a versatile VLM model that takes the task instruction $\mathcal{L}$, multimodal observations $\mathcal{I}, I_t^c$, and history $\mathcal{H}_{t-1}$, and generates all high-level actions. We design four types of

Table 1: **Dataset Overview for Instruction Fine-tuning.** Our dataset consists of four subsets, each corresponding to one of the four primary task actions: *Pick*, *Place*, *Navigate to Point*, and *Search Scene Frame*. The dataset is designed to encompass diverse scenarios and objects, ensuring comprehensive coverage of open-world mobile manipulation tasks.

Task Action	Pick	Place	Nav to point	Search scene frame
				
Data Size	64.7K	68.9K	59.6K	378.8K
Task Description	Move Arm Hammer Diaper Pail Refills 12 Pack from the Brunel-style bar stool to the white 2-seater sofa.	Move Shark from the Conlay kitchen to the comfortable sofa.	Move wood block from the 7-piece dining set with grey chairs to the Low kitchen element, Natural element.	Move flat screwdriver from the Modern Industrial Dresser, Natural Material to the Stacked shelf system.
Context Description	I have embarked on my task and am steadily advancing toward the Brunel-style bar stool, where the Arm Hammer Diaper Pail Refills 12 Pack MFWkmoweejt is situated.	I have embarked on my task and successfully navigated to the Conlay kitchen, retrieving the Shark with ease. Now, I am inching closer to the cozy haven of the comfortable sofa, where I will soon place the Shark.	The task has started and I have navigated to 7-piece dining set with grey chairs and picked up the wood block , I am getting closer to Low kitchen element, Natural element where I should place wood block.	The journey has commenced, and I have successfully navigated to the Modern Industrial Dresser, Natural Material, where I have now picked up the flat screwdriver.
Action Information	[[68, 755, 239, 967]]	[[447, 539, 999, 999]]	[[246, 666, 285, 705]]	4

high-level actions: 1) Posed image retrieval, 2) Navigate to point, 3) Pick, and 4) Place, which are associated with planners and the grip controller. However, due to the extended time horizon of the OWMM task, simply generating the executable action is insufficient. We instruct the VLM model to monitor the state through robot history and to infer the subsequent action by considering both the history and the present observations. Figure 3 demonstrates our model architecture as well as its input and output. For more details on model implementation, see Appendix C.

4 Dataset

In this section, we elaborate on our data construction pipeline and quality verification method in section 4.1. A detailed analysis of the data is provided in section 4.2.

4.1 Dataset Construction

For effective OWMM-VLM model training, generating the ground truth for OWMM is essential, covering navigation, object grasping, and manipulation affordances. Previous research[39, 21] often generates question-answer pairs from images or videos, lacking comprehensive action sequence representations and necessary affordance information for contextual understanding.

To address this challenge, we developed a data collection pipeline. Using Habitat simulation[25], we first constructed task sequences to complete the OWMM task based on the Planning Domain Definition Language (PDDL)[22]. We then directed the robot to execute task sequences within the simulator, recording key information at each step. With data selection strategy and filtering pipeline, we constructed question-answer pairs based on the definition of the problem mentioned in section 3.1. We constructed the reasoning and summarization components based on predefined templates. Specifically, the summarization from each step is systematically incorporated into the "Robot's History" framework to inform the question of the next step. See Appendix D for details.

Finally, we collected scene graph frames for each episode. We first set the robot at the location of the receptacle where objects were initially located and the goal receptacle, sampling the robot's

head-view images. Subsequently, we randomly positioned the robot and captured its head-view images. More details can be found in Appendix D.

4.2 Dataset Analysis

We used 143 scenes from The Habitat Synthetic Scenes Dataset (HSSD)[13] and combined objects from YCB Objects[4] and Google Scanned Objects[6] to create a dataset with 157 unique manipulation objects. We collected 1471 receptacles from our selected scenes. In each scene, objects were randomly placed for the robot to pick and relocate to another receptacle, resulting in 400 episodes per scene for our experiments.

Based on the data collection pipeline described in 4.1, we collected episodes from each scene and ultimately gathered 21,046 valid episodes, obtaining approximately 235k annotations. As shown in Table 5, the dataset is composed of: pick action dataset of 64.7K image-text pairs, place action dataset of 68.9K image-text pairs, navigation dataset of 59.6K image-text pairs, and a search scene frame dataset with 378.8K multi-image-text pairs.

In our datasets, we also apply a re-labeling process for objects and receptacles, unlike HomeRobot's fixed criteria[37]. We kept the original object labels and used GPT-4o to rewrite receptacle labels. These labels were diverse and descriptive, suited for open-world scenarios.

5 Experiments

In this section, we present the evaluation results in both simulation and real-world data. We present the experimental results of single-step evaluation for *OWMM-VLM* in our simulated benchmark in section 5.1 and episodic evaluation for the *OWMM-Agent* in our simulated benchmark in section 5.2. We then present the real-world evaluation in section 5.3. Due to the page limit, we discuss the data scaling law and how data diversity impacts the model performance in Appendix D.2. For the ablation study on model design, such as the choice of generating bounding boxes rather than points, please see Appendix G. We further provide the qualitative comparisons of different models in Appendix H.

Table 2: **Single-step evaluation of VLM models on OWMM core multi-modal capabilities.** The OWMM-VLM-38B model achieves the best performance across all metrics, demonstrating its superior ability to integrate scene understanding, decision-making, and action generation. *: Since PIVOT and RoboPoint are designed for a single image, we also report the single image grounding results for fairness.

Model/ Task Score	Ego-centric Decision-making↑	Image Retrieval↑	Affordance Grounding (object)↑	Affordance Grounding (receptacle)↑	Affordance Grounding (navigation)↑	Time Consumption(s)↓
OWMM-VLM-38B(ours)	**97.85%**	**87.54%**	**0.97(±0.14)**	**0.94(±0.19)**	**0.88(±0.17)**	36.58
OWMM-VLM-8B(ours)	96.72%	79.04%	0.93(±0.14)	0.91(±0.20)	0.83(±0.21)	16.58
GPT-4o[11]	48.53%	46.46%	0.56(±0.38)	0.35(±0.40)	0.07(±0.21)	160.74
Internvl2.5-8B[5]	17.52%	1.27%	0.05(±0.19)	0.18(±0.31)	0.14(±0.26)	16.06
GPT-4o+PIVOT[24]	52.72%	55.38%	0.67(±0.38)	0.45(±0.44)	0.05(±0.18)	22.91
GPT-4o+Robopoint[39]	49.56%	49.72%	0.64(±0.41)	0.38(±0.42)	0.06(±0.20)	14.19
Test of Single Image Grounding(*)						
Robopoint[39]*	—	—	0.91(±0.33)	0.83(±0.11)	0.72(±0.11)	—
PIVOT(GPT-4o)[24]*	—	—	0.86(±0.13)	0.84(±0.12)	0.74(±0.13)	—

Table 3: **Agent success rate in OWMM Task.** OWMM-VLM-38B model consistently outperforms others across all metrics.

Method	Full Task	Image Retrieval(Object)	Robot close to Object	Object Picked	Image Retrieval(Goal)	Robot close to Goal	Dead Loop
OWMM-VLM-38B(ours)	**21.90%**	**88.56%**	**84.64%**	**38.56%**	**30.39%**	**23.53%**	**0/308**
OWMM-VLM-8B (ours)	9.45%	81.43%	74.59%	17.92%	15.96%	10.42%	**0/308**
GPT-4o+PIVOT	0.33%	59.15%	10.13%	0.65%	0.33%	0.00%	195/308
GPT-4o+Robopoint	0.33%	56.86%	11.11%	1.31%	0.00%	0.00%	184/308
Experiment with more lenient distance tolerance							
OWMM-VLM-38B(ours)	**51.52%**	**89.23%**	**88.22%**	**62.96%**	**51.52%**	**44.78%**	**0/308**
OWMM-VLM-8B (ours)	38.59%	83.22%	81.21%	52.35%	39.93%	33.56%	**0/308**
GPT-4o+PIVOT	1.68%	60.27%	12.12%	5.39%	1.68%	1.35%	204/308
GPT-4o+Robopoint	3.03%	52.86%	10.10%	4.04%	2.69%	1.35%	209/308

5.1 Single-step Evaluation

In the single-step evaluation, we assess three core VLM capabilities for the OWMM task: 1) Egocentric Decision-making: We evaluate the success rate of choosing correct action categories. 2) Image Retrieval: We measure the image retrieval success rate. 3) Affordance Grounding: Instead of predicting points directly like in [39, 24], *OWMM-VLM* outputs a bounding box, from which we compute the center as the target point. With the target point, we compute the score for affordance grounding by $s = \Sigma_i \mathbb{1}_{valid}(i) \times (1 - norm_dist_i)$, where $\mathbb{1}_{valid}(i)$ is the indicator function of whether the model generates: an action matched with ground truth and a valid bounding box or point on the $i - th$ test case. 1 if both conditions are satisfied simultaneously, and 0 otherwise. $norm_dist_i \in [0, 1]$ is the distance between the predicted target point and the ground truth point, normalized by the diagonal of the image. In short, $s \in [0, 1]$ measures VLM's ability to generate accurate grounding with the correct format. Higher scores indicate better performance.

Regarding the baseline methods, we have evaluated both 1) multitasking foundation VLM models, including GPT-4o[11] and InternVL-2.5-8B that share the same unified input and output configuration as ours and 2) modularized agent with multiple models, including GPT-4o+PIVOT[24] and GPT-4o+Robopoint[39]. For Robopoint and PIVOT, which specialize in grounding, GPT-4o serves as the higher-level module for decision-making and image retrieval. If GPT-4o's actions need grounding, its outputs are combined with task details as input to Robopoint and PIVOT for grounding.

The results are reported in Table 2. Our model excels in decision-making, achieving state-of-the-art results in image retrieval and affordance grounding. GPT-4o and InternVL2.5, as generalist models, perform poorly at affordance grounding. In contrast, RoboPoint and Pivot that concentrated on affordance grounding, exhibit capabilities on par with our model in this task, indicating that existing specialized approaches already provide good effect on robot's action affordance.

Moreover, our model demonstrates a marked improvement over GPT-4o in decision-making tasks. This advantage directly translates into higher overall accuracy compared to methods that employ GPT-4o as the agent. In other words, using the data from our data synthesis pipeline to conduct a supervised fine-tuning yields a significant enhancement in robotic decision-making performance.

5.2 Episodic Evaluation

In episodic evaluation, we assess how well each model completes an OWMM task episode in the simulator. Task success is measured by placing objects in goal receptacles using distance thresholds of 0.85m or 1.7m. The 0.85m threshold relates to half the average diagonal length of goal receptacles' 3D bounding boxes in our test set.

Additionally, we introduce three metrics to assess subgoals: 1) Image retrieval: Success rate in locating object and goal receptacles from multiple posed images. 2) Object Picked: The success rate of the robot grasping an item when its end effector is either within 0.15m or 0.8m of the target, with the latter matching standard HomeRobot setups [37]. 3) Robot close to: The success rate of robot staying within 1.5m or 2.0m of the object or goal receptacle before picking or placing. Additionally, we propose the "dead loop" metric to quantify the number of cyclic stagnations occurring during test episodes. As mentioned in 5.1, GPT-4o may erroneously output image retrieval decisions when the expected action is navigation, thereby inducing cyclic stagnation. Detailed experimental results are presented in Table 3. See Appendix F for extra details about evaluation settings.

5.3 Real world Evaluation

In our real-robot experiments, we adopted the mobile manipulation system described in Robi Butler[33] within a real-world home environment. For safety reasons, we cannot allow the agent to fully operate the fetch robot in the real world. When OWMM-VLM generates a multi-modal action to execute, the agent prompts the visualization of the action and waits for human confirmation, and the fetch robot only executes the action with human consent.

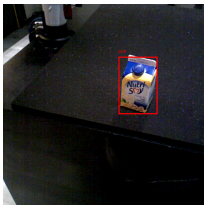
We first had the robot navigate through the scene with human control to perform SLAM process. We then select 10 test samples from the sequence of the robot's head view during its run. We used human operators to judge the model's output according to several criteria: whether the chosen action was correct, whether the predicted affordance was accurate, and whether the target was reachable, among other factors. The results of these experiments are presented in Table 4. The results show that the

model trained on synthetically generated data in the simulator also demonstrates strong zero-shot generalization capability in real-world scenarios. Table 5 presents the agent action prediction result on real-world data.

Table 4: **Real world single evaluation.** OWMM-VLM-38B model achieved the best performance, and OWMM-VLM-8B model also outperformed the baseline. While the baseline model demonstrated relatively strong affordance grounding capabilities for objects, its poor performance in action decision-making led to incorrect navigation.

Method	**Image Retrieval**	**Affordance Grounding(object&receptacle)**	**Affordance Grounding(navigation)**	**Total Acc**
OVMM-VLM-38B(ours)	7/10	**10/10**	**10/10**	**90.00%**
OVMM-VLM-8B (ours)	5/10	**10/10**	9/10	80.00%
GPT-4o+PIVOT	**8/10**	6/10	0/10	46.67%
GPT-4o+Robopoint	**8/10**	6/10	0/10	46.67%

Table 5: **Demonstration of single step evaluation in real world.** These demos showcase OWMM-VLM-38B's outputs, illustrating that even though its training data are drawn entirely from our data-synthesis approach in the simulator, the model delivers outstanding decision-making and affordance-grounding performance in real-world settings.

Model's Output Action	Pick	Place	Nav to point
			
Task Description	Move the NutriSoy Bean Milk Box from the Minimalist Black Workstation Desk to the White Rectangular Office Meeting Table.	Move the banana from the black desk to the White Rectangular Office Meeting Table.	Move the chip box from Genuine Leather Sofa to the white table.
Context Description	The task has started and I am getting closer to the Minimalist Black Workstation Desk where the NutriSoy Bean Milk Box is located.	The task has started and I have navigated to the black desk and picked up the banana, I am getting closer to the White Rectangular Office Meeting Table where I should place the banana.	The task has started and I am getting closer to Genuine Leather Sofa where the chip box is located.
Action Information	[576, 263, 769, 548]]	[[0, 445, 1000, 999]]	[[539, 978, 578, 999]]

6 Conclusion

In this paper, we introduced OWMM-Agent, a novel agent architecture featuring the OWMM-VLM, a vision-language model fine-tuned via a simulation-based agentic data synthesis pipeline for Open-World Mobile Manipulation (OWMM) tasks. This approach enables the VLM to learn state tracking, multi-view reasoning, and multi-modal action generation grounded in global scene understanding and agent embodiment. Extensive experiments demonstrated that our OWMM-VLM, particularly the 38B variant, achieves state-of-the-art performance in single-step multi-modal capabilities like egocentric decision-making and affordance grounding, outperforming generalist VLMs and specialized robotics models. Episodic evaluations in simulated environments further confirmed the OWMM-Agent's superior success rates and robustness against common failure modes like dead loops, while real-world tests on a Fetch robot indicated strong zero-shot generalization. Ablation studies underscored the importance of our design choices, such as bounding box prediction and integrated reasoning, and revealed that while data scaling is crucial, egocentric spatial intelligence can be learned effectively even with limited object and scene diversity if data volume is sufficient. Future work will focus on addressing limitations like pre-mapping reliance and enhancing cross-embodiment adaptability for more complex manipulation tasks. Please also refer to the appendix for discussions about the potential impact of this research in Appendix A and extended discussions on limitations in Appendix B.

References

[1] Jiaqi Bai, Hongcheng Guo, Zhongyuan Peng, Jian Yang, Zhoujun Li, Mohan Li, and Zhihong Tian. Mitigating hallucinations in large vision-language models by adaptively constraining information flow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23442–23450, 2025.

[2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $pi0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[3] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024.

[4] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015.

[5] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[6] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022.

[7] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[8] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous systems*, 61(12):1258–1276, 2013.

[9] Difei Gao, Lei Ji, Zechen Bai, Mingyu Ouyang, Peiran Li, Dongxing Mao, Qinchen Wu, Weichen Zhang, Peiyi Wang, Xiangwu Guo, et al. Assistgui: Task-oriented desktop graphical user interface automation. *arXiv preprint arXiv:2312.13108*, 2023.

[10] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024.

[11] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[12] Charles C Kemp, Aaron Edsinger, Henry M Clever, and Blaine Matulevich. The design of stretch: A compact, lightweight mobile manipulator for indoor human environments. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3150–3157. IEEE, 2022.

[13] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16384–16393, 2024.

[14] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[15] Yeseung Kim, Dohyun Kim, Jieun Choi, Jisang Park, Nayoung Oh, and Daehyung Park. A survey on integration of large language models with intelligent robots. *Intelligent Service Robotics*, 17(5):1091–1107, 2024.

[16] Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, and Jie Tang. Autowebglm: A large language model-based web navigating agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5295—-5306, 2024.

[17] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[18] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

[19] Peiqi Liu, Yaswanth Orru, Jay Vakil, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024.

[20] Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*, 2024.

[21] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16488–16498, 2024.

[22] Drew McDermott, Malik Ghallab, Adele E. Howe, Craig A. Knoblock, Ashwin Ram, Manuela M. Veloso, Daniel S. Weld, and David E. Wilkins. Pddl-the planning domain definition language. 1998. URL `https://api.semanticscholar.org/CorpusID:59656859`.

[23] Andrew Melnik, Michael Büttner, Leon Harz, Lyon Brown, Gora Chand Nandi, Arjun PS, Gaurav Kumar Yadav, Rahul Kala, and Robert Haschke. Uniteam: Open vocabulary mobile manipulation challenge. *arXiv preprint arXiv:2312.08611*, 2023.

[24] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. *arXiv preprint arXiv:2402.07872*, 2024.

[25] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

[26] Dicong Qiu, Wenzong Ma, Zhenfu Pan, Hui Xiong, and Junwei Liang. Open-vocabulary mobile manipulation in unseen dynamic environments with 3d semantic maps. *arXiv preprint arXiv:2406.18115*, 2024.

[27] Ri-Zhao Qiu, Yafei Hu, Ge Yang, Yuchen Song, Yang Fu, Jianglong Ye, Jiteng Mu, Ruihan Yang, Nikolay Atanasov, Sebastian Scherer, et al. Learning generalizable feature fields for mobile manipulation. *arXiv preprint arXiv:2403.07563*, 2024.

[28] Fabian Schmalstieg, Daniel Honerkamp, Tim Welschehold, and Abhinav Valada. Learning hierarchical interactive multi-object search for mobile manipulation. *IEEE Robotics and Automation Letters*, 2023.

[29] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[30] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*, 2023.

[31] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

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

[33] Anxing Xiao, Nuwan Janaka, Tianrun Hu, Anshul Gupta, Kaixin Li, Cunjun Yu, and David Hsu. Robi butler: Multimodal remote interaction with a household robot assistant, 2025. URL `https://arxiv.org/abs/2409.20548`.

[34] Haoyu Xiong, Russell Mendonca, Kenneth Shaw, and Deepak Pathak. Adaptive mobile manipulation for articulated objects in the open world. *arXiv preprint arXiv:2401.14403*, 2024.

[35] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[36] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. *arXiv preprint arXiv:2412.14171*, 2024.

[37] Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alex William Clegg, John Turner, Zsolt Kira, Manolis Savva, Angel Chang, Devendra Singh Chaplot, Dhruv Batra, Roozbeh Mottaghi, Yonatan Bisk, and Chris Paxton. Homerobot: Open vocab mobile manipulation. In *Conference on Robot Learning*, 2023. URL `https://arxiv.org/abs/2306.11565`.

[38] Sriram Yenamandra, Arun Ramachandran, Mukul Khanna, Karmesh Yadav, Jay Vakil, Andrew Melnik, Michael Büttner, Leon Harz, Lyon Brown, Gora Chand Nandi, et al. Towards open-world mobile manipulation in homes: Lessons from the neurips 2023 homerobot open vocabulary mobile manipulation challenge. *arXiv preprint arXiv:2407.06939*, 2024.

[39] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

[40] Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2025.

[41] Sipeng Zheng, Jiazheng Liu, Yicheng Feng, and Zongqing Lu. Steve-eye: Equipping llm-based embodied agents with visual perception in open worlds. *arXiv preprint arXiv:2310.13255*, 2023.

[42] Peiyuan Zhi, Zhiyuan Zhang, Muzhi Han, Zeyu Zhang, Zhitian Li, Ziyuan Jiao, Baoxiong Jia, and Siyuan Huang. Closed-loop open-vocabulary mobile manipulation with gpt-4v. *arXiv preprint arXiv:2404.10220*, 2024.

[43] Yuqi Zhu, Shuofei Qiao, Yixin Ou, Shumin Deng, Shiwei Lyu, Yue Shen, Lei Liang, Jinjie Gu, Huajun Chen, and Ningyu Zhang. Knowagent: Knowledge-augmented planning for llm-based agents. *arXiv preprint arXiv:2403.03101*, 2024.

[44] Filippo Ziliotto, Tommaso Campari, Luciano Serafini, and Lamberto Ballan. Tango: Training-free embodied ai agents for open-world tasks. *arXiv preprint arXiv:2412.10402*, 2024.

A Impact Statement

This work contributes to the long-term vision of creating generalist household robots capable of assisting with daily activities in homes and other human-centric spaces. Ethically, deploying such systems raises considerations regarding safety, privacy, and workforce displacement. Ensuring safe interactions with humans and securing data used for training are critical priorities. In addition, while automation may replace certain household jobs, it also creates opportunities for new roles in robot design, deployment, and maintenance. Future societal implications include increased accessibility to robotic assistance for individuals with disabilities or aging populations. By addressing current limitations through continued research into adaptability and real-world robustness, OWMM-VLM can pave the way toward more inclusive and effective robotic solutions for societal benefit.

B Limitations

In this work, we have proposed a novel embodied agent architecture with a foundational VLM model to address the open-world mobile manipulation problem. However, we also identify some limitations of our approach.

Pre-mapping: Although our method does not require 3D reconstruction of the environment, we still assume a pre-mapping phase with a camera pose graph and 2D occupancy map for path planning in navigation.

Complex manipulation: Following the grasping setup in [37], our agent and model can be directly applied robot with suction as end effector. However, our model fells short in the circumstances when the robot needs to control complex end effectors like dexhands.

Cross-embodiment: As demonstrated in the experiments, our model learns the object-scale prior for spatial understanding and reasoning. However, when deploying the model onto other robots with different mechanical compositions such as maximum arm stretch distance, our model could fail, i.e. the cross-embodiment issue.

C Implementation Details

Regarding the model's architecture, we have trained two variants consisting of 8 billion and 38 billion parameters, based on the pre-trained model from InternVL-2.5[5]. The 8B model is composed of InternViT-300M and InternLM-2.5-7B[3], and the 38B model is composed of InternViT-6B and Qwen2.5[35]. We directly finetune the base model on our OWMM dataset. The OWMM-VLM model is trained to autoregressively generate the response tokens consisting of the output action and its corresponding task context in JSON format. Specifically, we freeze the parameters in ViT and only adjust the parameters in MLP and LLM. As for the training time, *OWMM-VLM-8B* is trained on 8X NVIDIA A100 GPUs for about 7 hours, and *OWMM-VLM-38B* is trained on 24X NVIDIA A100 GPUs for about 18 hours. Both our models were trained for 1 epoch. For the testing, we deploy *OWMM-VLM* and RoboPoint[39] locally and use the openAI API to access GPT-4o and PIVOT[24].

D Details of Datasets

D.1 Extra Dataset Construction Details

Our evaluation pipeline is constructed using the HomeRobot[37] framework, which serves as a software structure designed to enable comprehensive benchmarking in both simulated and real-world settings. Specifically, we use the simulation part of HomeRobot project, built on Habitat platform[25], with 200 scenes, 150 categories, and 7892 object instances. The original episodic data in HomeRobot are generated with Stretch Robot[12], which has a special telescopic arm instead of a normal articulated arm with rotary joints. This adds additional difficulty in base control as it requires the mobile chassis to rotate accurately to align the arm with the target object for manipulation. However, the baseline VLMs and methods we are going to compare with are designed for robots with conventional articulated arms [39, 24], providing a broad range of chassis poses that allow for successful arm manipulation.

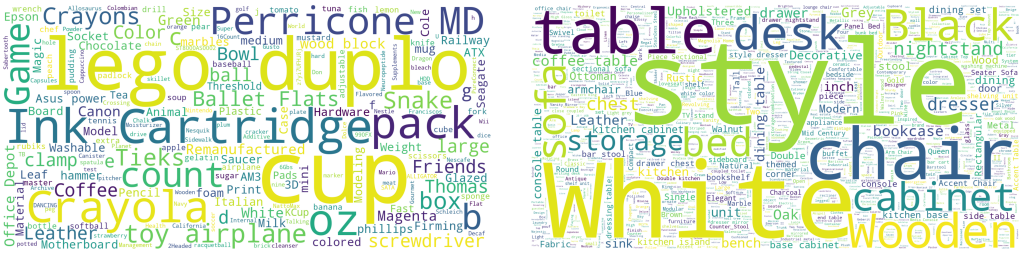
Therefore, we recreate the OWMM episodic training and testing datasets in the simulation using the Fetch Robot, which is a mobile robot equipped with a standard articulated arm and has also been integrated into the Habitat platform. We partitioned the scenes into training and testing sets using a ratio of 113:30. Besides, we allocated 157 objects between the training and validation sets with a ratio of 137:20, ensuring that the testing set contained entirely unseen objects. This division resulted in a total of 152k training data entries and 4k testing data entries, establishing a robust dataset for training and testing in our OWMM task.

In dataset construction pipeline,we first sample key imformation at each step.This information included the robot's coordinates, current action, the positions of objects and receptacles, and the extrinsic parameters of the robot's head-view camera. In particular, at this stage, we did not collect the robot's head-view images to enhance the data collection efficiency.We recollected the robot's head-view images of these steps within the simulator after selection strategy.

In the key step data selection strategy, for navigation actions, among all steps that the robot is moving, we select the step that the receptacle is visible from the robot's head-view image as the start point of the navigation action. The point at which the robot stops moving is considered the end point of the navigation action. Within these steps, we sample the waypoint step data at specified intervals. For grasp and manipulation actions, we select the first three frames during which the robot executes the action as the pre-defined action data.

The data filtering pipeline ensures the following matters: for navigation actions, both the receptacle and the next waypoint are within the robot's head-view image. For grasp actions,the object to be grasped is reachable by the robotic arm, and the object is within the robot's head-view image. For manipulation actions,the receptacle intended for object placement is reachable by the robotic arm, and the receptacle is within the robot's head-view image.

To enhance the diversity of the dataset, we paraphrased reasoning and summarization parts of the answers using GPT-4o mini.



(a) Object Categories (b) Receptacle Categories

Figure 4: **Word Cloud Distribution of Objects and Receptacles in our dataset**

D.2 Analysis on the training data

This analysis tries to answer two questions: 1) **How does the diversity of objects and environments affect the model's performance on unseen objects and environments in the test set**? We examine dataset diversity using three 45k-sample sets: 100% scenes and objects, 100% scenes with 30% objects, and 30% scenes with 100% objects. We control the total number of training samples while changing the number of object instances or scenes appearing in the training data. 2) **How does the model's performance change as the training data scales up?** For data scaling, we use five data sizes: 0k (no fine-tuning), 15k (10%), 45k (30%), 76k (50%), and 152k (100%). At 0k, we give the Internvl-2.5-8B model limited input-output pairs, allowing it to generate structured outputs via in-context learning. We evaluate the performance in image retrieval, egocentric decision-making, and three affordance grounding subtasks. Results are shown in Table 6 and Figure 5.

The results for the first question show that object and scene diversity have negligible effects on multi-modal capabilities, as metric fluctuations remain within a 5% range. For the second quesion, data scaling is crucial for enhancing OWMM-VLM's performance. As seen in Figure 5, increasing the dataset from 0k to 152k samples shows a logarithmic improvement, especially at lower sizes (0k to 15k, and 15k to 45k). However, benefits diminish near 152k. While larger datasets aid generalization,

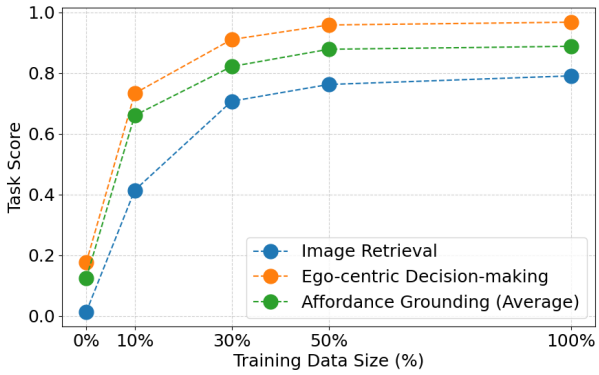


Figure 5: **OVMM-VLM-8B Sub-task Performance with the Increase of Training Data Size.** The task scores consistently improve as the training data size increases.

marginal gains decrease beyond a threshold. As performance gains plateau, egocentric decision making approaches a success rate of 1.0, whereas image retrieval lingers at approximately 0.8. This difference is likely due to the model's limited capacity with 8 billion parameters. We also draw two extra observations from the experiment:

1) The embodiment prior for deciding the current action based on the ego-centric RGB image, especially how close the robot should be to interact with the target objects, can be learned in a data-driven approach.

2) The ability to comprehend multiple images or the multimodal context length may present one of the bottlenecks for VLM models to function as the core cognitive model for intelligent robots, particularly when scene-level understanding is essential.

Table 6: **Results with different data diversity data scales.** The best performance across training sets with different scales is indicated with **bold font**. Besides, underline highlights the best performance across three 45k-sample training sets with different diversity.

Data Composition/ Task Score	Ego-centric Decision-making↑	Image Retrieval↑	Affordance Grounding (object)↑	Affordance Grounding (receptacle)↑	Affordance Grounding (navigation)↑
0k(0%)	17.52%	1.27%	0.05(±0.19)	0.18(±0.31)	0.14(±0.26)
15k(10%)	73.27%	41.36%	0.69(±0.43)	0.84(±0.29)	0.45(±0.41)
45k(30%)	91.01%	70.68%	0.88(±0.24)	0.84(±0.31)	0.74(±0.31)
45k(100% scene + 30% object)	91.56%	71.95%	0.87(±0.26)	0.89(±0.23)	0.72(±0.33)
45k(30% scene + 100% object)	88.96%	69.12%	0.87(±0.26)	0.84(±0.31)	0.69(±0.36)
76k(50%)	95.79%	76.20%	0.91(±0.19)	0.88(±0.24)	**0.84(±0.20)**
152k(100%)	**96.72%**	**79.04%**	**0.93(±0.14)**	**0.91(±0.20)**	0.83(±0.21)

E Details of Baseline Setting

As Robopoint and PIVOT are designed for single-image QA task, we adjusted some settings to enable them fully utilizing their capabilities under the OWMM task.

E.1 Single Image Grounding

For the single-step evaluation, we first extracted robot's task instruction from the original prompt of the current step. Based on the ground truth action of the current step and whether the robot picks up an object, we designed new task instructions, as shown in Table 7. For Robopoint, we appended the following context: "Find a few spots for robot to execute the action. Your answer should be formatted as a list of tuples, i.e. [(x1, y1), (x2, y2), ...], where each tuple contains the x and y coordinates of a point satisfying the conditions above. The coordinates should be between 0 and 1, indicating the normalized pixel locations of the points in the image." This configuration aligns with Robopoint's original settings.

Ground Truth Action	New Task Instruction
Pick	The robot needs to pick {object item} on {target rec}
Place	The robot needs to place {object item} on {goal rec}
Nav to point(object picked)	The robot needs to navigate closer to the {goal rec} for placing {object item}
Nav to point(object not picked)	The robot needs to navigate closer to the {target rec} for picking {object item}

Table 7: **Redefined Task Instructions.** {object item}, {target rec} and {goal rec} are from robot's task instruction.

For PIVOT, we configured the following parameters: n_samples_init=10, n_samples_opt=6, n_iters=2. In our evaluation settings, as the input consists of a single RGB image and task instructions, we randomly sample initial points in the image from a 2D Gaussian distribution. The distribution is parameterized with a mean of (256, 256) and standard deviation of (100, 100).

E.2 Agent Setting

We employed GPT-4o for agent construction.GPT-4o first receives our instruction inputs and returns JSON-formatted responses. When gpt's output action is "search scene frame", we directly adopt GPT-4o's response as the agent's current-step output. For actions "nav to point", "pick", or "place", the system sends both the action name and robot's current-view RGB image (single frame) to Robopoint/PIVOT for action affordance. The reformulated task instruction sent to Robopoint/PIVOT follows this template:

"The robot needs to {task_instruction}. Now the robot needs to {gpt_output_action}. {robot_history}"

where {task_instruction} is the original task instruction,{gpt_output_action} is gpt's output action,{robot_history} is the summarization of previous step. In single-step evaluation, Robopoint and PIVOT process these new task instructions using the same methodology described in Appendix E.1.In episodic evaluation, we transmit depth information to PIVOT while maintaining consistency with its original configuration.

F Extra Details of Episodic Evaluation

As mentioned in section 5.2, we designed the following metrics for episodic evaluation. More detailed specifications of these metrics are outlined below:

Object to Goal Distance: We used the object to goal distance as the metric to determine whether objects are successfully placed in goal receptacles. To establish appropriate thresholds, we first calculated the 3D bounding box diagonal distances of all goal receptacles in the test set, filtering out those with distances less than 0.75m or greater than 3m.Table 8 show some examples of goal receptacles in our test set.Subsequently, we computed the average diagonal distance (1.7m) from the remaining valid receptacles. Based on this value, we selected half of the average (0.85m) as the strict threshold criterion and the full average (1.7m) as the relaxed threshold criterion. This threshold approach ensures successful placement recognition when robots position objects near goal receptacles,and reasonable constraint boundaries to prevent excessive leniency in evaluation.

F.1 Simulation

For one simulation step, the robot state delta is calculated by forward kinematics, as implemented by the Habitat 3.0 environment[25]. The robot state and observations updates can be expressed mathematically as:

$$\mathbf{x}_{t+1} = f_k(\mathbf{x}_t, \mathbf{a}_t, \Delta t)$$
$$I^c_{t+1}, D^c_{t+1} = f_{obs}(\mathbf{x}_{t+1})$$

where $\mathbf{x}_t$ stands for robot current state (e.g., joint angles, positions), $\mathbf{a}_t$ stands for velocities, and f_k represents the kinematic model function that computes the next robot state within the discretized

Circular table, Small oak 0.789m	Hisa Wooden Console 1.655m	Silver Picardy Bed 2.504m	tall sideboard 2.931m

Table 8: **Example Goal Receptacles in our Test Set.** The numbers in the figure represent the diagonal distances of the 3D bounding boxes of the receptacles. This indicates that there is significant variation in the sizes of the goal receptacles in the test set.

time step of duration Δt. f_{obs} is the observation model function, decided by the sensor link forward kinematics function and camera model.

G Ablation Study on OWMM-VLM

The ablation study evaluates the contributions of the components of the OWMM-VLM model. We focus on grounding output formats, comparing the bounding box and point coordinate, and we assess the inclusion of reasoning and summarization in the outputs. Furthermore, we examine the beam search option provided by the base model Internvl-2.5-8B[5]. The results are in Table 9.

From the table, we have these observations and indications:

1) **Beam Search.** Beam search is a decoding algorithm widely used in language generation, maintaining a beam number of top candidate sequences at each step. Beam search enhances Ego-centric Decision-making and Affordance Grounding tasks, with minimal impact on Image Retrieval, but increases temporal and spatial overhead in inference, especially on the 38B variant. Hence, its effect is briefly shown only in the ablation study.

2) **Grounding Format.** Replacing bounding box predictions with direct output coordinates reduces performance in Affordance Grounding, especially for objects $(0.9251 \rightarrow 0.6542)$ and receptacles $(0.9060 \rightarrow 0.6479)$. It is postulated that the large-scale visual grounding data in the pre-trained model allow our model to utilize this prior knowledge. The consistency in output format between the base model and the instruction fine-tuning dataset aids the training process.

3) **Reasoning and Summarization**. Removing reasoning and summarization capabilities leads to the worst performance across most metrics, with a decrease in Image Retrieval $(0.7904 \rightarrow 0.6586)$ and Ego-centric Decision-making $(0.9672 \rightarrow 0.9049)$. This highlights the critical role of reasoning and summarization in maintaining contextual coherence and task understanding.

Table 9: **Ablation Study on OWMM-VLM.** The best performance is indicated with **bold font**.

Model Ablation	Ego-centric Decision-making↑	Image Retrieval↑	Affordance Grounding (object)↑	Affordance Grounding (receptacle)↑	Affordance Grounding (navigation)↑
OWMM-VLM-8B	96.72%	**79.04%**	$0.93(\pm 0.14)$	$0.91(\pm 0.20)$	$0.83(\pm 0.21)$
+ beam search	**97.30%**	78.47%	$\mathbf{0.97(\pm 0.14)}$	$\mathbf{0.93(\pm 0.21)}$	$\mathbf{0.85(\pm 0.18)}$
+ output-coord	96.70%	78.19%	$0.65(\pm 0.15)$	$0.65(\pm 0.17)$	$0.63(\pm 0.14)$
- reasoning and summarization	90.49%	65.86%	$0.88(\pm 0.24)$	$0.83(\pm 0.33)$	$0.82(\pm 0.24)$

H Qualitative Evaluation

We provide the qualitative evaluation of our *OWMM-VLM* model compared to other baseline models.

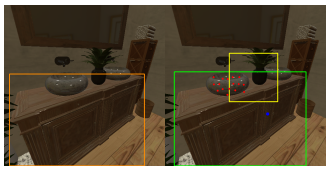
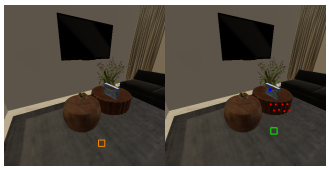
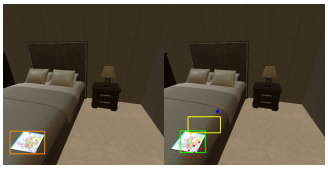
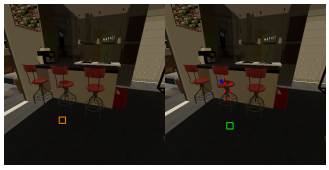
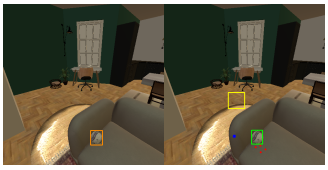
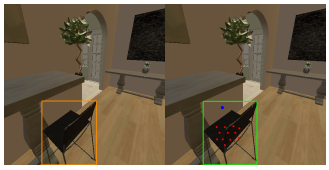
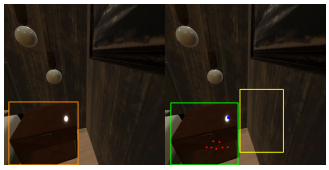
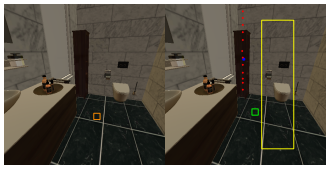
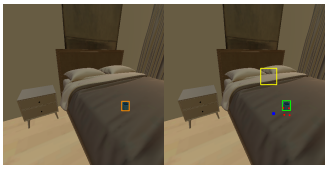
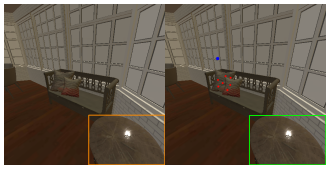
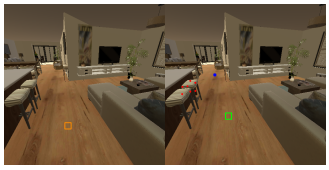
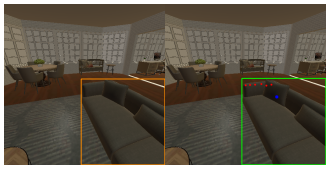
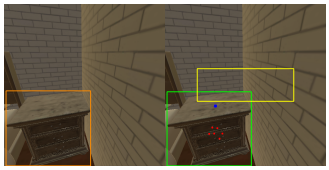
Pick	Place	Nav to point
		
		
		
		
		
		
		

Table 10: **Single step Qualitative Evaluation.** The table demonstrates the single step qualitative evaluation results: — represent the **ground truth**; — represent GPT-4o; • represents RoboPoint; • represents PIVOT; — represents InternVL base model; — represent **ours**

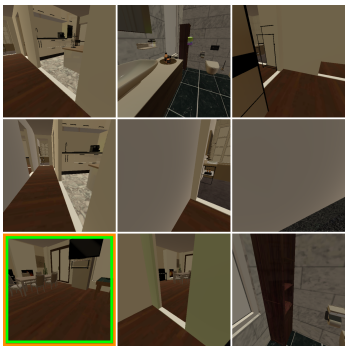
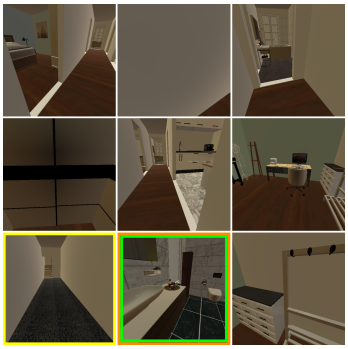
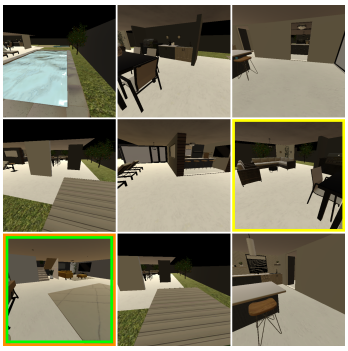
Task Description	Context Description	Response
Move Schleich Allosaurus from the Upholstered Sofa to the Brown and Gold Accent Cabinet.	Task just started.	
Move Tena Pads Heavy Long 42 pads from the Dark Wooden Tall Open Bathroom Cabinet to the multifunctional games table.	The task has started and I have navigated to Dark Wooden Tall Open Bathroom Cabinet and picked up the Tena Pads Heavy Long 42 pads.	
Move hammer from the Multiple Drawer Short Boy to the Dark Wooden Tall Open Bathroom Cabinet.	The task has started and I have navigated to Multiple Drawer Short Boy and picked up the hammer.	
Move 065-b cups from the Unch metal and wood bar stool to the Magnolia Home Foundry Console Table.	The task has started and I have navigated to Unch metal and wood bar stool and picked up the 065-b cups.	

Table 11: **Single step Qualitative Evaluation Search Scene Frame.** The table demonstrates the single step qualitative evaluation search scene frame results: — represent the **ground truth**; — represent GPT-4o; — represents InternVL base model; — represent **ours**