Open-vocabulary Mobile Manipulation in Unseen Dynamic Environments with 3D Semantic Maps

Dicong Qiu[1◇], **Wenzong Ma**[1◇], **Zhenfu Pan**[2], **Hui Xiong**[1∗], **Junwei Liang**[1∗]
[1]Hong Kong University of Science and Technology (Guangzhou) [2]Jacobi.ai

Abstract: Open-Vocabulary Mobile Manipulation (OVMM) is a crucial capability for autonomous robots, especially when faced with the challenges posed by unknown and dynamic environments. This task requires robots to explore and build a semantic understanding of their surroundings, generate feasible plans to achieve manipulation goals, adapt to environmental changes, and comprehend natural language instructions from humans. To address these challenges, we propose a novel framework that leverages the zero-shot detection and grounded recognition capabilities of pretraining visual-language models (VLMs) combined with dense 3D entity reconstruction to build 3D semantic maps. Additionally, we utilize large language models (LLMs) for spatial region abstraction and online planning, incorporating human instructions and spatial semantic context. We have built a 10-DoF mobile manipulation robotic platform *JSR-1* and demonstrated in real-world robot experiments that our proposed framework can effectively capture spatial semantics and process natural language user instructions for zero-shot OVMM tasks under dynamic environment settings, with an overall navigation and task success rate of 80.95% and 73.33% over 105 episodes, and better SFT and SPL by 157.18% and 19.53% respectively compared to the baseline. Furthermore, the framework is capable of replanning towards the next most probable candidate location based on the spatial semantic context derived from the 3D semantic map when initial plans fail, keeping an average success rate of 76.67%.

Keywords: Open-vocabulary, Mobile Manipulation, Dynamic Environments, 3D Semantic Maps, Zero-shot, LLMs, VLMs

1 Introduction

Mobile manipulation is a vital and fundamental capability of autonomous robots. The recent surge of pretraining LLMs and VLMs, along with their integration with robotics, has drawn significant attention in research, particularly in the areas of open-vocabulary and zero-shot capabilities for autonomous robots in navigation and mobile manipulation tasks [1, 2, 3, 4, 5, 6]. Although recent studies have explored robot manipulation in both semantic navigation [2, 6, 7] and open-vocabulary [8, 1, 9, 10] settings, they often assume either a static environment [11, 6] or a non-mobile robot fixed on a tabletop [8, 12], and sometimes operate purely in simulation [6, 13]. These settings limit the capability of putting a moving robotic platform into real-world use. Additionally, the lack of prior knowledge about an unseen environment and the dynamic factors leading to changes in the setup further complicate the problem. However, addressing these two problems is crucial for developing robots to become generalists and be practically applicable to a wider spectrum of real-world tasks.

To address the above challenges, we propose a novel two-stage framework enabling robots to explore and build up semantic understanding of an unseen open environment, generate feasible efficient plans by taking environment semantic context into consideration, overcoming dynamic changes of

◇ Equal contribution.
∗ Corresponding authors.

the environment, and understand human instructions and hints in natural language. At the 3D semantic mapping stage, a robot explores the environment with heuristic algorithms [14, 15, 16] where sequential observations from the robot goes into a simultaneously location and mapping (SLAM) pipeline [17, 18, 19, 20] to reconstruct dense 3D structure of the environment for navigation, and a semantics extraction and abstraction pipeline leveraging the open-vocabulary detection and zero-shot abstract reasoning capabilities of LLMs [21] and VLMs [22, 23, 24] to build up semantic understanding of the environment captured in a 3-layer 3D semantic map (3DSMap) for open-vocabulary navigation and mobile manipulation. At the semantics-aware open-vocabulary mobile manipulation stage, the robot parses human instructions and hints given in natural language, and comes up with corresponding semantically optimal region search plans with LLMs. When it finds the target object to fetch using the open-vocabulary detection (OVD) capability of VLMs [22, 24], traditional search-based and probabilistic path and motion planners [25, 26, 27, 28] can take over to pick up the target object and return it to the user. Compared to traditional learning-based frameworks [12, 3, 6, 29] that usually require intensive training, inspired by [11, 9, 8, 30], the approach we propose does not requires training by taking the most advantages of pretrained foundation models to understand and reason about the environment semantics and open-vocabulary region and instance concepts in zero-shot.

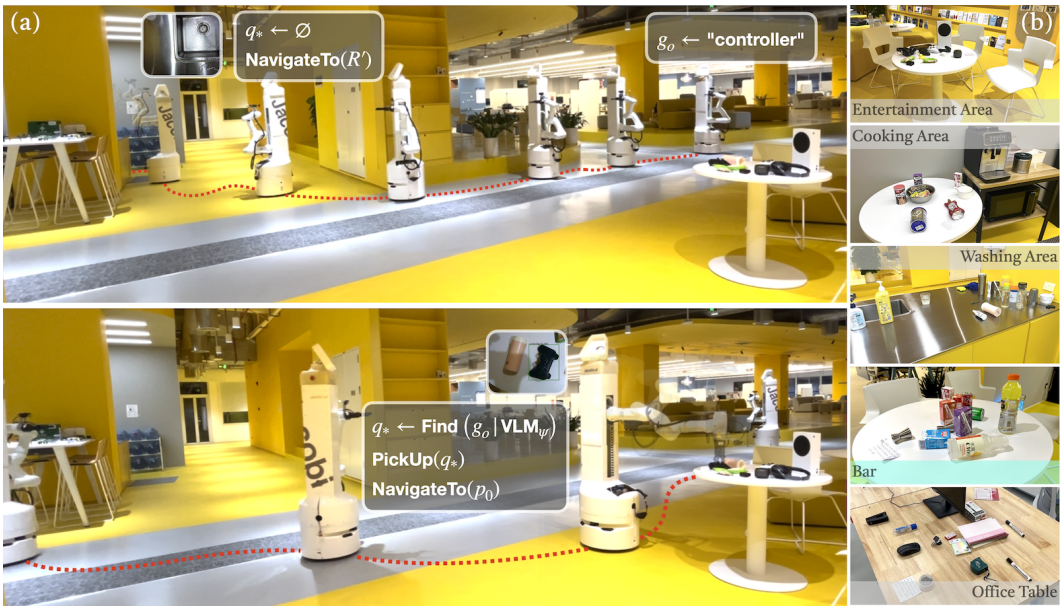

Figure 1: An illustration of our real-world experiment on OVMM with the *JSR-1* robot platform we built. (a) A sample run in our experiment. The robot receives a misleading instruction "`fetch the controller from the washing area`" from the user. Obeying the instruction given by the user, it prioritizes the `washing area` as the first region to search, where it fails to to find the target object g_o = `"controller"`. R' = `entertainment area` follows as the next most relevant semantic region, and *JSR-1* successfully finds g_o, picks up the correct corresponding instance q^*, and return to p_0. (b) The 5 regions in our experiment scene setup with 20 different categories of objects scattered within them conforming commonsense in daily life.

We validate the effectiveness and robustness of the proposed framework on the mobile manipulation robotic platform *JSR-1* we built. Our experiment (Sec. 5) shows the capability of the approach we proposed in taking in natural language human instructions for the zero-shot open-vocabulary mobile manipulation task in dynamic environments, and replanning towards the next most possible location in accord to the spatial semantic context from the 3D semantic map, if the target object does not present at the location given by the prior knowledge from the user.

The main contributions of this work includes (i) a novel two-stage approach for a robot to explore and build up semantic understanding of an unseen open environment a zero-shot and efficiently tackle mobile manipulation tasks in the real world in open-vocabulary and dynamic environment

settings, (ii) a 3-layer structure of 3D semantic map representation to capture not only the structural information but also the instance and abstract region semantics of the environment, (iii) a proposal-approval work flow with generic-proposed VLMs to effectively reduce the false positive rate of OVD models, and (iv) demonstrating the effectiveness of pretrained LLMs and VLMs on complex real-world robotic tasks with dynamic factors and semantic commonsense of an environment being considered, dispensing with the need of intensive training or fine-tuning neural network parameters.

2 Related Work

Pretraining foundation models for robotics. Pretraining foundation models [31, 32, 33], especially pretraining LLMs [34, 35, 21, 36, 37] and VLMs [24, 38, 39, 40, 41, 42, 43], have demonstrated remarkable zero-shot capabilities in a wide range of tasks [44, 45, 46, 47, 48, 49]. Leveraging such advantages in robotic and embodied applications has been under active research in recent years. [50] and [51] combine multiple pretrained models as submodules for visual-language manipulation and navigation. [11] directly fuses pretrained CLIP features [43, 39] for language conditioned navigation. As for planning and decision making, [5, 52, 53] decompose high-level tasks with LLMs into feasible plans consisting of pretrained or predefined executable actions. [30, 54, 55] solve long-horizon robot planning problems with LLMs by incorporating classical planners. [3, 12, 6] adapt the weights of pretraining vision-language models to train end-to-end models that directly map robot observations to low-level control actions, while [8, 56] propose training-free approaches with pretraining foundation models to achieve similar purpose. In this work, we utilize the zero-shot and abstract reasoning capabilities of pretraining VLMs and LLMs to address semantic-aware OVMM tasks, allowing real-world robots to perform complex tasks in unseen and dynamic environment settings without specialized training or fine-tuning, which has significant potential to generalize across various robotic platforms and tasks.

Scene reconstruction and semantic mapping. Reconstruction of environment facilitates robot navigation and manipulation by providing structural context of its surroundings. Popular feature-based SLAM methods such as ORB-SLAM [57, 19, 18, 17] and VINS [58, 59] support monocular and visual-inertial RGB-D SLAM with loop closure for online applications, requiring less computation than traditional offline multi-view geometric dense 3D reconstruction approaches [60, 61, 60]. Recent advances in neural radian fields [62, 63] and Gaussian splatting [64] have been further developed to reconstruct continuous and dense 3D scenes and instance representations for robotic applications in mapping and localization [65, 66, 67, 68], navigation [69], object pose estimation [70] and manipulation [71, 72, 73, 74]. Moreover, [75, 9, 10] attach entity and spatial semantics captured in 3D scene graph [76] or implicit representations [77], on top of structural reconstruction, enabling spatial semantic awareness of robots in task planning, entity localization, navigation, etc. In this work, we introduce 3DSMaps, a novel 3-layer structural representation to capture both spatial structure and semantics in one, and demonstrate the effectiveness of it in our experiment.

Mobile manipulation and navigation in the open world requires open-vocabulary and strong adaptive capabilities of a robot. Although recent progress on pretraining foundation models have inspired more research focus, it remains an open problem [1]. [3, 12, 6, 7] propose to address this problem in end-to-end pipelines. On the other hand, [30, 8, 56] introduce training-free approaches with pretraining foundation models to tackle this challenge. Prior studies often assume either a static environment [11, 6] or a non-mobile robot setting [8, 12], and sometimes operate purely in simulation [6, 13]. We propose a novel framework in this study to holistically address the aforementioned challenges of robotic mobile manipulation in the open world with open-vocabulary, dynamic and unseen environment settings, demonstrating its potential to bring robots into real-world use.

3 Problem Settings and Formulation

We follow the open-vocabulary object navigation and mobile manipulation problem settings proposed by [1, 2, 7] and consider a more interactive problem setting which can be found more com-

mon in the real world. A robot is ask by a human user to find and fetch an object for the user in open-vocabulary settings. Aiming for generality and practicability of applying our framework to real-world scenarios, we do not assume any prior knowledge and further consider dynamic factors of the environment. The robot needs to explore and learn about the structural and semantic information about the environment. And at online task execution stage, objects may not remain where they were during exploration. The user may optionally provide suggestions to the robot about where the object may be located, but such suggestions can be sometimes misleading as well in reality, which is taken into consideration in our problem definition.

The goal of the above task can be formally described as a (g_o, g_R)-tuple. The robot receives a natural language instruction $\mathcal{L}$ from the user about the goal, a typical example is "`fetch something [from somewhere]`", where "`something`" suggests the target object g_o to fetch and "`somewhere`" is a hint from the user prior knowledge about the possible target region $\hat{g}_R$ where g_o may be located. Note that the hint $\hat{g}_R$ given by the user is optional and it can sometimes be misleading due to false memory of the user or dynamic changes of the environment. In a successful run, the robot shall reach the target region g_R, pick up the target object g_o and return it to the user.

Real-world scenarios are rather complicated, to avoid ambiguity and alleviate excessive engineering, there are two reasonable assumptions. (i) In any scene, there can be multiple types of objects and sometimes multiple objects in the same type presenting in the scene or even the same location. We may assume either the target object is the only one object with the same type presenting in the scene, or if there are multiple objects with the same type as the target object, fetching any of them is considered successful. (ii) View angle planning itself, especially in 3D settings, is an active standalone research topic. To focus on the core problem to tackle, we may assume one or several predefined view angles (camera poses) of the robot at each location in a scene. A robot can navigate and pose its camera to these predefined angles in search for the target object.

4 Method

Prior knowledge about the environment and exact region to fetch the target object from are inaccessible to a robot. Hints from users are optional and not assumed to be reliable. Beyond the open-vocabulary settings in mobile manipulation, it poses challenges on both building up structural and semantic knowledge about the environment, and effectively and efficiently leverage such knowledge to complete the aforementioned task. To that end, we propose a two-stage framework to holistically tackle the above problem in a training-free manner. At the 3D semantic mapping stage (Sec. 4.1), heuristic exploration and reconstruction from feature-based SLAM are followed by instance semantics extraction and region semantics abstraction, taking the zero-shot and abstract reasoning advantages of VLMs and LLMs for the robot to learn spatial structure and environment semantics in a training-free manner. At the semantics-aware open-vocabulary mobile manipulation stage (Sec. 4.2), the robot prioritizes the regions to search with LLMs in accord to the user instruction and the environment semantics, picks up the target object detected by open-vocabulary detection models and returns it to the user.

4.1 3D Semantic Mapping in Unseen Environments

In a completely unseen environment with no prior knowledge, it is essential for a robot to become aware of the environment structure and further build up semantic understanding of the environment. In this section, we propose a 3-layer (structural layer, instance semantics layer and abstraction region layer) structure of 3D semantic maps as illustrated in fig.2(a) to capture the both structure and semantics of the environment.

Heuristic exploration and structural reconstruction. Off-the-shelf heuristic frontier exploration algorithms [14, 15, 16, 78] are used for the robot to explore the environment. Sequential vision-inertial sensing input from the RGB-D camera and the IMU of the robot is recorded, consisting of RGB image frames $\{\mathcal{I}^t_{\text{RGB}}\}$ and depth frames $\{\mathcal{I}^t_{\text{D}}\}$ along with the IMU readings $\{I^t\}$, each of which is associated with a timestamp t. We adopt ORB feature based RGB-D-I simultaneous

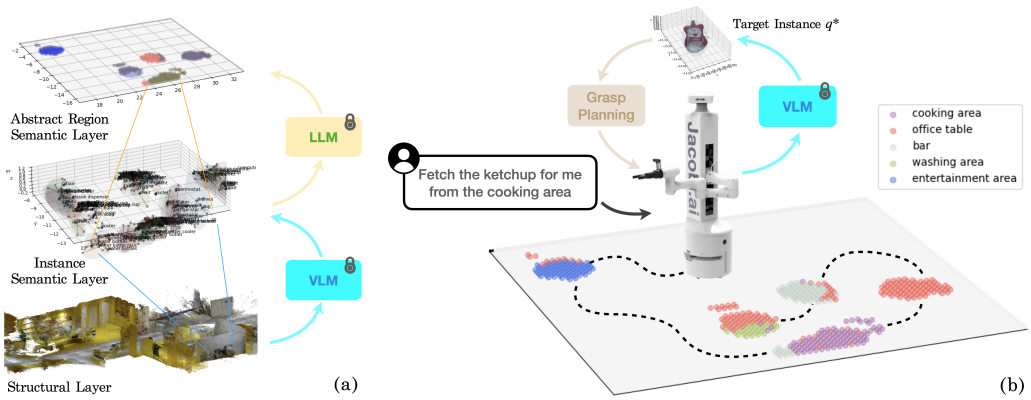

Figure 2: An overview of our proposed two-stage framework. (a) A 3-layer structured *3DSMap* is built at the 3D semantic mapping stage, by leveraging heuristic exploration and feature-based SLAM to build the *structural layer*, using pre-trained VLMs and LLMs to extract instances from and propose region divisions for an environment to build the *instance semantic layer* and the *abstract region semantic layer*, respectively. (b) At the semantics-aware OVMM stage, a robot takes both region semantics from *3DSMap* and user instructions as context to prioritize regions to search with LLMs and fetch the target object for the user.

localization and mapping (SLAM) algorithms [17, 18, 19, 20] to fuse the RGB-D camera data and the IMU readings, build 3D feature maps and determine key frames from loop-closure corrected camera poses trajectory $\{T^{\text{cam},k}\}_{k=1}^{K}$ with K being the total number of key frames.

The camera poses combined with the RGB and depth data associated to key frames in the sensor data sequence generates a key frame trajectory as a sequence of $(\mathcal{I}^k_{\text{RGB}}, \mathcal{I}^k_{\text{D}}, T^k_{\text{cam}})$-tuples. Each depth key frame $\mathcal{I}^k_{\text{D}}$ is by nature associated with a list of homogeneous 3D points $\mathbf{P}^k_{\text{cam}} = \{p^{(k,j)}_{\text{cam}}\}_j$ in the camera coordinate frame by the intrinsic parameters of the camera [79]. Dense reconstruction of the 3D environment structure in point cloud representation can be achieved by re-projecting the dense points from the depth frames onto the global 3D space $\mathbf{P}^k = T^{\text{cam},k}(\mathbf{P}^k_{\text{cam}})^{\mathsf{T}}$ and accumulating them [19] for localization and navigation. We can further color the point cloud by the re-projection and association of the 3D points to corresponding RGB image frames for visualization. The dense reconstruction of the environment structure composes the *structural layer* of a 3DSMap.

Geometric and semantic extraction of 3D instances. To build semantic awareness of the environment in a training-free manner inspired by [11] and [9], we adopt the Grounded SAM pipeline [22] for 2D open-vocabulary detection and pixel-wise segmentation. For each RGB image frame at key frame k, it detects object instances $\left\{b^{(k,i)}\right\}_{i=1}^{N_k}$ with open-vocabulary detection models, such as Grounding DINO [49] or Detic [40], and segments the detected instances with SAM [47] into 2D pixel masks $\left\{m^{(k,i)}\right\}_{i=1}^{N_k}$. Text prompt inputs required by OVD models consisting of instance proposals can be automatically generated by open-set image tagging models [80], captioning models [81, 82] or generic-purposed VLMs [24] (Appx. A.1). Re-projection of the 3D points $\mathbf{P}^k$ from depth frames onto each of pixel mask $m^{(k,i)}$ further segments these 3D points into point-based geometric representations $\left\{\mathbf{P}^{(k,i)}\right\}_{i=1}^{N_k}$ of the instances presenting in the scene, each of which is decomposed into its geometric center $\bar{p}^{(k,i)} = \frac{1}{N_k}\sum_{p\in\mathbf{P}^{(k,i)}} p$ and relative geometry $\bar{\mathbf{P}}^{(k,i)} = \mathbf{P}^{(k,i)} - \bar{p}^{(k,i)}$. Associated with object instance semantics $b^{(k,i)}$, especially the label of the instance, an instance geometric representation becomes a semantic geometric instance $q^{(k,i)} = \left(b^{(k,i)}, \bar{p}^{(k,i)}, \bar{\mathbf{P}}^{(k,i)}\right)$ and is then registered by its spatial coordinate $\bar{p}^{(k,i)}$ onto the *instance semantics layer* of a 3DSMap. The geometry $\mathbf{P}^{(k,i)}$ of an instance from a key frame by the nature of field of view is a partial geometric representation it, and multiple $\mathbf{P}^{(k,i)}$'s from different key frames ks may point to the same physical instance. Instance fusion [9] across key frames can effectively resolve this problem, however, it is not strictly required by our proposed framework.

Region semantics abstraction. We consider robots operating in large open scenarios. In real world, a scene in general consists of multiple functional areas with different semantic context.

Efficient utilization of such abstract semantics for planning is beneficial for a robot to overcome dynamic factors in an environment, i.e. objects may not always persist where they were when first observed. To this end, instances $\{q^{(k,i)}\}$ extracted and registered on the instance semantic layer are projected onto the 2D floor plane by simply removing the height dimension from their geometric center $\bar{p}^{(k,i)}$, forming a 2D bird-eye-view (BEV) instance semantics map. We place a circular sliding window with radius r on the BEV map, starting from the top-left corner and swiping through the entire map by certain step size Δd. At step (s_x, s_y), the area within the sliding window $\left\{(x,y)|(x - s_x\Delta d)^2 + (y - s_y\Delta d)^2 \leq r^2\right\}$ selects a set of object instances falling inside, whose labels after repeated terms removed effectively describe the objects presenting within this area. We then leverage the zero-shot abstract reasoning capability of LLMs [21] to come up with a list of region proposals, depicting the abstract region semantics of the area (Appx. A.2). After a full sweep, a zero-shot dense prediction of region semantics $R(x, y)$ over the entire BEV space is therefore generated, constituting the *abstract region layer* of a 3DSMap, providing abstract semantic information about different regions in an environment.

4.2 Semantics-aware Open-vocabulary Mobile Manipulation

With structural and semantic knowledge built up about the unseen environment and captured in 3DSMaps, a robot can efficiently leverage such knowledge to complete OVMM tasks and withstand dynamic changes in the environment conforming semantic commonsense. Alg.1 depicts the overall procedure of open-vocabulary mobile manipulation considering environment semantic context.

Algorithm 1 Semantics-aware open-vocabulary mobile manipulation

Require: natural language user instruction $\mathcal{L}$, random sequence of regions in the scene $\mathcal{S}_R$
Ensure: robot starts nearby the user at p_0

```
(g_o, ĝ_R) ← LLM_θ(L | T_Parsing)                          ▷ parse user instruction
S*_R ← LLM_θ(S_R, g_o | T_Prioritization)                  ▷ prioritize regions by semantic relevance with LLM
if ĝ_R ≠ ∅ then
    S*_R ← Reprioritize(S*_R, ĝ_R)                         ▷ assign highest priority to ĝ_R
end if
for R ∈ S*_R do
    for p ∈ Searchable(R) do
        NavigateTo(p)                                      ▷ navigate to a searchable location p in region R
        q ← Find(g_o | VLM_ψ)                              ▷ attempt to find g_o leveraging VLM
        if q ≠ ∅ then
            if PickUp(q) then                              ▷ attempt to pick up instance q
                NavigateTo(p_0)                            ▷ bring it back to the user
            end if
        end if
    end for
end for
```

Open-vocabulary semantic prioritization for search regions. The robot receives a natural language instruction $\mathcal{L}$ from the user about the goal of the mobile manipulation task, asking the robot the fetch a target object g_o. The user may optionally provide hints about at which region $\hat{g}_R$ the target object may be. We parse $\mathcal{L}$ and extract g_o and $\hat{g}_R$ from it, using pretrained LLMs [21] with prompt template for instruction parsing $\mathcal{T}_{\text{Parsing}}$ (Appx. A.3). The target object g_o along with a list of regions $\mathcal{S}_R$ presenting in the scene obtained from region semantics abstraction (Sec. 4.1) are input into the pretrained LLMs with template $\mathcal{T}_{\text{Prioritization}}$ to prioritize the regions by semantic relevance between the target object g_o and each of the region in $\mathcal{S}_R$ (Appx. A.4), which is followed by an optional re-prioritization step to assign highest priority to the region $\hat{g}_R$ suggested by the user in $\mathcal{L}$. The final outcome from the prioritization step (and re-prioritization step) is an ordered sequence of regions S^*_R indicating the search priority of different regions for g_o.

Prioritized navigation and in-region exploration. Following the prioritized list S^*_R of regions to search, the robot attempts to reach these regions one after another and find the target object g_o. Since

we consider mobile manipulation of grounded robots, though in general our framework is applicable to any 3D spatial robots, to reduce computational complexity, the structural layers of 3DSMaps is flattened into 2D cost maps [83, 84] for navigation. Heuristic reachability analysis [85, 86] is adapted to pre-compute a list 2D searchable locations within a region with overlapped and infeasible locations filtered out. Global navigation trajectories are planned using a search-based global planner [27, 25] and followed by the DWA local planner [87, 28], towards each of the searchable locations $p \in \text{Searchable}(R)$ within a region R then another following the prioritized list S^*_R. At each searchable location p, the robot attempts to find and pick up the target object g_o, whose details will be further discussed below. At failure of finding g_o at p, the robot will head to the next searchable location and repeat the above procedure as illustrated in alg. 1, allowing the robot to complete the task efficiently following the optimal semantic search path.

Open-vocabulary instance detection and manipulation. Reaching at a searchable location p in region R, the robot will then adopt [25, 26] to conduct end-effector planning for a camera pose T^{cam} looking towards (e.g. downwards on) the operation area at p. Similar pipeline [22, 49, 47] from Sec. 4.1 is reused for OVD and pixel-wise segmentation, and the prompt instructing the OVD models is simply tha target object g_o. However, OVD models suffer from high false-positive rate [88], significantly reducing the overall reliability. Therefore, we propose a proposal-approval work flow, with an OVD model [22] coming up with detection proposal and another VLM model [24] double checking the result to either approve or reject. The robot then plans the end-effector trajectory of its gripper for grasping [26, 25], towards the semantic geometric instance $q^* = \left(b^*, \bar{p}^*, \tilde{\mathbf{P}}^*\right)$ with the highest confidence from the approved list of detected instances. We allows at most $n_e = 3$ trials on grasping at each location p. After success in picking up q^*, the robot will return to the user.

5 Experiment

We analyze the effectiveness and performance of our proposed method in the a large real-world open space with the *JSR-1* mobile manipulation robotic platform we built. Our experiment covers 135 independent episodes (eps.) in total with real robot for quantitative evaluation, which are split into 5 experiment groups. Experiment details are presented and analyzed in below.

Experiment settings. Our robotic experiment is conducted in a large real-world indoor open space covering an area of over 200 m^2. Within it, we set up 5 regions as shown in fig.1(b) along with 20 different categories of objects placed within these regions conforming daily commonsense as shown in tab.1. Our quantitative experiment with real robot platform consists of 135 independent episodes in total, which are divided into 5 experiment groups (Appx.C.1). At the beginning of each run, the robot starts from the same position p_0 nearby the user and receives a natural language instruction.

Table 1: Division of Experiment Groups

Exp. Group	# of Eps.	Description
NoHint	45	User instructions do not contain region hints
Random	30	Control group, where region search orders are randomized
Hinting	30	User instructions contain region hints
ErrantSemantics	15	Target objects are placed in semantic irrelevant regions
Misleading	15	User instructions contain misleading region hints

Robot platform. We have built a 10-DoF mobile manipulation semi-humanoid robotic platform *JSR-1*, the robot appeared in fig.1(a), which consists of a 2-DoF wheeled chassis, a 6-DoF robotic arm with a 1-DoF gripper, and an RGBD camera on nearby its end-effector. Besides, a 1-DoF waist link assembles its chassis and arm, extending its operation range from 0 to 200cm in height.

Evaluation metrics. We introduce 5 metrics for quantitative evaluation. Success on first trial (**SFT**) is the ratio of episodes where the first semantic proposal for search region contains the target object. Success on navigation (**SN**) is the ratio of episodes where the robot has navigated to nearby the target object. Success on picking (**SP**) is the ratio of episodes where the robot has picked up the target

object successfully. Overall success rate (**Succ.**) is the ratio of episodes where the robot has successfully carried the target object to p_0. Success weighted by path length **SPL** $= \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{\max(p_i, l_i)}$ measures the efficiency of reaching the goal in addition to the success rate [89].

Table 2: The overall performance of proposed method evaluated through real-world robot experiment. We compare the result of each experiment groups with the control group (Random), and report the SFT increased (SFT Incr.) and SPL increased (SPL Incr.) for each group respectively.

Group. Metrics	SN	SP	Succ.	SFT	SPL	SFT Incr.	SPL Incr.
Random (R.)	26/30	23/30	23/30	7/30	0.5332	-	-
NoHint	34/45	33/45	33/45	33/45	0.6878	**214.31%**	**31.46%**
Hinting	25/30	21/30	21/30	30/30	0.6841	**328.63%**	**30.75%**
ErrantSemantics	13/15	13/15	13/15	0/15	0.4942	-100.00%	-5.54%
Misleading	13/15	10/15	10/15	0/15	0.4519	-100.00%	-13.63%
Total without R.	80.95%	73.33%	73.3%	60.00%	0.6254	**157.18%**	**19.53%**

Experiment Result Analysis. By the experiment result (tab.2), our proposed method demonstrates a decent performance and robustness in complex real-world OVMM tasks, achieving an overall success rate of 73.33% and a successful navigation rate of 80.95%, under various situations, including objects being randomly placed in semantic irrelevant regions and user giving misleading instructions. Compared to the control group (Random), it has better overall performance on SFT and SPL by 157.18% and 19.53% respectively. As for normal situations without misplacement of objects or misleading user instructions, our approach demonstrates a significant performance advantage in the NoHint and Hinting groups, with better SFT by 214.31% and 328.63%, and SPL by 31.46% and 30.75%. The result shows that our proposed method is able to efficiently incorporate spatial region semantics and user hints for semantic-aware OVMM tasks, and it can robustly recover from failure and complete the tasks even being exposed to dynamic factors and misleading instructions.

Table 3: Comparison of the experiment results between NoHint and Hinting groups.

Exp. Group	SFT (%)	SN (%)	SP (%)	Succ. (%)	SPL (0.0-1.0)
NoHint	73.33	75.56	73.33	73.33	0.6878
Hinting	**100.00**	83.33	70.00	70.00	0.6841

Furthermore, as shown in tab.3, the Hinting group achieves 100.00% SFT for its leverage of region hints in the instruction from the user, indicating the effectiveness of our framework to incorporate prior knowledge and suggestions from humans. We also notice from tab.2 that in Misleading group SPL is below average and less than the control group 13.63%. It indicates that our framework is sensitive to human instruction, and misleading or wrong suggests can lead to lower efficiency. However, it keeps an **SN** $= 13/15 = 86.67\%$ above the average and a reasonable overall success rate at 66.67%, showing the failure recovering capability of our proposed framework.

6 Conclusion and Future Work

In this work, we propose a novel framework that tackles the problem of Open-Vocabulary Mobile Manipulation, which leverages the zero-shot detection and grounded recognition capabilities of pre-training visual-language models (VLMs) combined with dense 3D entity reconstruction to build 3D semantic maps. Additionally, we utilize large language models (LLMs) for spatial region abstraction and online planning, incorporating human instructions and spatial semantic context. We have built a 10-DoF mobile manipulation robotic platform *JSR-1* and conducted real-world experiments to demonstrate the effectiveness of our proposed training-free method. In future work, we will focus on incorporating autonomous exploration techniques to extend our system's capabilities to unknown environments. Furthermore, exploring the use of multiple agents or robots for collaborative exploration and scanning of environments will improve efficiency and coverage in unknown or large areas.

Acknowledgments

The authors would like to thank Lei Zhu from Jacobi.ai for his help during the experiment, and HKUST(GZ) MakerSpace for test site support.

References

[1] S. Yenamandra, A. Ramachandran, K. Yadav, A. S. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. Clegg, J. M. Turner, Z. Kira, M. Savva, A. X. Chang, D. S. Chaplot, D. Batra, and R. Mottaghi. Homerobot: Open-vocabulary mobile manipulation. In *Proceedings of (CoRL) Conference on Robot Learning*, October 2023.

[2] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot. Navigating to objects in the real world. *Science Robotics*, 8(79):eadf6991, 2023.

[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[4] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia, C. Finn, and K. Hausman. Open-world object manipulation using pre-trained vision-language models. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=9al6taqfTzr.

[5] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, pages 287–318. PMLR, 2023.

[6] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. In *Neural Information Processing Systems (NeurIPS)*, 2022.

[7] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *In Neural Information Processing Systems (NeurIPS)*, 2020.

[8] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

[9] S. Lu, H. Chang, E. P. Jing, A. Boularias, and K. Bekris. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *7th Annual Conference on Robot Learning*, 2023.

[10] H. Chang, K. Boyalakuntla, S. Lu, S. Cai, E. P. Jing, S. Keskar, S. Geng, A. Abbas, L. Zhou, K. Bekris, et al. Context-aware entity grounding with open-vocabulary 3d scene graphs. In *7th Annual Conference on Robot Learning*, 2023.

[11] C. Huang, O. Mees, A. Zeng, and W. Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023.

[12] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, brian ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=XMQgwiJ7KSX.

[13] Z. Al-Halah, S. K. Ramakrishnan, and K. Grauman. Zero experience required: Plug & play modular transfer learning for semantic visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17031–17041, 2022.

[14] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997.

[15] Z. Sun, B. Wu, C.-Z. Xu, S. E. Sarma, J. Yang, and H. Kong. Frontier detection and reachability analysis for efficient 2d graph-slam based active exploration. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2051–2058. IEEE, 2020.

[16] C. Cao, H. Zhu, F. Yang, Y. Xia, H. Choset, J. Oh, and J. Zhang. Autonomous exploration development environment and the planning algorithms. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8921–8928. IEEE, 2022.

[17] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[18] C. Campos, J. M. Montiel, and J. D. Tardós. Inertial-only optimization for visual-inertial initialization. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 51–57. IEEE, 2020.

[19] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[20] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[21] 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.

[22] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[23] Y. Zhang, X. Huang, J. Ma, Z. Li, Z. Luo, Y. Xie, Y. Qin, T. Luo, Y. Li, S. Liu, et al. Recognize anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514*, 2023.

[24] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, B. Li, P. Luo, T. Lu, Y. Qiao, and J. Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.

[25] I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. doi:10.1109/MRA.2012.2205651. https://ompl.kavrakilab.org.

[26] J. Kuffner and S. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 995–1001 vol.2, 2000. doi:10.1109/ROBOT.2000.844730.

[27] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[28] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997.

[29] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

[30] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=wMpOMO0Ss7a.

[31] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*, 2023.

[32] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136, 2023. doi:10.1109/JAS.2023.123618.

[33] C. Li, Z. Gan, Z. Yang, J. Yang, L. Li, L. Wang, J. Gao, et al. Multimodal foundation models: From specialists to general-purpose assistants. *Foundations and Trends® in Computer Graphics and Vision*, 16(1-2):1–214, 2024.

[34] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.

[35] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023.

[36] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[37] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

[38] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning, 2023.

[39] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RriDjddCLN.

[40] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022.

[41] L. H. Li*, P. Zhang*, H. Zhang*, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao. Grounded language-image pre-training. In *CVPR*, 2022.

[42] H. Zhang, P. Zhang, X. Hu, Y.-C. Chen, L. H. Li, X. Dai, L. Wang, L. Yuan, J.-N. Hwang, and J. Gao. Glipv2: Unifying localization and vision-language understanding. *arXiv preprint arXiv:2206.05836*, 2022.

[43] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

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

[45] J. Zhang, J. Huang, S. Jin, and S. Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[46] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[47] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[48] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023.

[49] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[50] A. Zeng, M. Attarian, K. M. Choromanski, A. Wong, S. Welker, F. Tombari, A. Purohit, M. S. Ryoo, V. Sindhwani, J. Lee, et al. Socratic models: Composing zero-shot multimodal reasoning with language. In *The Eleventh International Conference on Learning Representations*, 2022.

[51] D. Shah, B. Osiński, S. Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pages 492–504. PMLR, 2023.

[52] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009, 2023.

[53] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022.

[54] 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. *arXiv preprint arXiv:2304.11477*, 2023.

[55] T. Silver, V. Hariprasad, R. S. Shuttleworth, N. Kumar, T. Lozano-Pérez, and L. P. Kaelbling. Pddl planning with pretrained large language models. In *NeurIPS 2022 foundation models for decision making workshop*, 2022.

[56] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[57] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. doi: 10.1109/TRO.2015.2463671.

[58] T. Qin and S. Shen. Online temporal calibration for monocular visual-inertial systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3662–3669. IEEE, 2018.

[59] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[60] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[61] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[62] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[63] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021.

[64] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[65] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*, pages 1347–1356. PMLR, 2022.

[66] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022.

[67] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022.

[68] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison. Gaussian Splatting SLAM. 2024.

[69] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.

[70] F. Li, S. R. Vutukur, H. Yu, I. Shugurov, B. Busam, S. Yang, and S. Ilic. Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2123–2133, 2023.

[71] G. Lu, S. Zhang, Z. Wang, C. Liu, J. Lu, and Y. Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. *arXiv preprint arXiv:2403.08321*, 2024.

[72] Q. Dai, Y. Zhu, Y. Geng, C. Ruan, J. Zhang, and H. Wang. Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1757–1763. IEEE, 2023.

[73] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[74] Y. Li and D. Pathak. Object-aware gaussian splatting for robotic manipulation. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024. URL https://openreview.net/forum?id=gdRI43hDgo.

[75] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler. Open-vocabulary queryable scene representations for real world planning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11509–11522. IEEE, 2023.

[76] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5664–5673, 2019.

[77] J. Hsu, J. Mao, and J. Wu. Ns3d: Neuro-symbolic grounding of 3d objects and relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2023.

[78] C. Cao, H. Zhu, Z. Ren, H. Choset, and J. Zhang. Representation granularity enables time-efficient autonomous exploration in large, complex worlds. *Science Robotics*, 8(80):eadf0970, 2023.

[79] R. Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[80] X. Huang, Y.-J. Huang, Y. Zhang, W. Tian, R. Feng, Y. Zhang, Y. Xie, Y. Li, and L. Zhang. Open-set image tagging with multi-grained text supervision. *arXiv e-prints*, pages arXiv–2310, 2023.

[81] X. Huang, Y. Zhang, J. Ma, W. Tian, R. Feng, Y. Zhang, Y. Li, Y. Guo, and L. Zhang. Tag2text: Guiding vision-language model via image tagging. *arXiv preprint arXiv:2303.05657*, 2023.

[82] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

[83] L. Matthies and A. Elfes. Integration of sonar and stereo range data using a grid-based representation. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 727–733. IEEE, 1988.

[84] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI magazine*, 9(2):61–61, 1988.

[85] N. Vahrenkamp, T. Asfour, and R. Dillmann. Robot placement based on reachability inversion. In *2013 IEEE International Conference on Robotics and Automation*, pages 1970–1975. IEEE, 2013.

[86] A. Makhal and A. K. Goins. Reuleaux: Robot base placement by reachability analysis. In *2018 second IEEE international conference on robotic computing (IRC)*, pages 137–142. IEEE, 2018.

[87] B. P. Gerkey and K. Konolige. Planning and control in unstructured terrain. In *ICRA workshop on path planning on costmaps*. Citeseer, 2008.

[88] C. Pham, T. Vu, and K. Nguyen. Lp-ovod: Open-vocabulary object detection by linear probing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 779–788, 2024.

[89] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

A Experiment Details of LLMs and VLMs on Zero-shot Tasks

A.1 VLMs for Instance Proposals for OVD Prompts

A.1.1 InternVL 1.5

Below is an example of using InternVL 1.5 [24] for the zero-shot instance proposals for OVD prompts.

What objects are in the image? Return in JSON format `{"objects": object_name, ...}` .

```
{"objects": ["headphones", "controller", "phone", "rubber duck", "charger",
"speaker", "note", "plug"]}
```

A.2 LLMs for Zero-shot Region Abstraction Proposal

A.2.1 GPT-4o

Below is an example of using GPT-4o [21] for the zero-shot region abstraction proposal task.

SYSTEM: The user will give you a list of objects inside a region and a list of region candidates in JSON format `{"objects": [object_1, object_2, ...], "region_candidates": [region_candidate_1, region_candidate_2, ...]}` , please order these regions in decreasing order of likelihood and return just in JSON format `{"region_proposals": [region_1, region_2, ...]}` , do not reply markdown format.

USER: `{ "objects": [ "beverage", "bottle", "can", "soda", "table", "juice", "mustard", "ketchup" ], "region_candidates": [ "living room", "classroom", "coffee shop", "kitchen", "hallway" ] }`

ASSISTANT: `{'region_proposals': ['kitchen', 'coffee shop', 'living room', 'classroom', 'hallway']}`

A.2.2 InternVL 1.5

Below is an example of using InternVL 1.5 [24] for the zero-shot region abstraction proposal task.

I will give you a list of objects inside a region, and a list of region candidates in JSON format, please order these regions in decreasing order of likelihood and return just in JSON format only: `{ "region_proposals": [ region_1, region_2, ... ] }`, with no verbose information or justification. Now, let's begin. `{ "objects": [ "beverage", "bottle", "can", "soda", "table", "juice", "mustard", "ketchup" ], "region_candidates": [ "living room", "classroom", "coffee shop", "kitchen", "hallway" ] }`

```
{ "region_proposals": [ "coffee shop", "kitchen", "classroom", "living room",
"hallway" ] }
```

A.2.3 LLaMA 3

Below is an example of using LLaMA 3 [34] for the zero-shot region abstraction proposal task.

I will give you a list of objects inside a region, and a list of region candidates in JSON format, please order these regions in decreasing order of likelihood and return just in JSON format only: `{ "region_proposals": [ region_1, region_2, ... ] }`, with no verbose information or justification. Now, let's begin. `{ "objects": [ "beverage", "bottle", "can", "soda", "table", "juice", "mustard", "ketchup" ], "region_candidates": [ "living room", "classroom", "coffee shop", "kitchen", "hallway" ] }`

```
{ "region_proposals": [ "kitchen", "coffee shop", "living room", "hallway",
"classroom" ] }
```

A.3 LLMs for User Natural Language Instruction Parsing

A.3.1 GPT-4o

Below is an example of using GPT-4o [21] to parse user instruction in natural language, and convert into structural instruction.

SYSTEM: The user will give you an instruction in natural language about something ("target object") he/she wants to find, and the user may or may not give further guess about what region the target object may be located. Please turn the instruction into JSON format `{"target_object": target_object, "region": region}`, where `region` shall be set as `null` if the user does not give further guess about region, do not reply markdown format.

USER: Fetch the spray cleaner from the entertainment area.

ASSISTANT: `{'target_object': 'spray cleaner', 'region': 'entertainment area'}`

USER: Fetch the milk powder.

ASSISTANT: `{'target_object': 'milk powder', 'region': None}`

A.4 LLMs for Search Regions Prioritization

A.4.1 GPT-4o

Below is an example of using GPT-4o [21] to prioritize regions to search for the target object. In this part, we only consider the mapping from target object to a prioritized list of regions to search, without considering the region suggestion from user instruction.

SYSTEM: The user will give you a list of region names in JSON format `{"regions": [region_1, region_2, ...], "target_object": object_name}` , and the name of a target object he/she wants to find, please proposal a list containing the names of these regions in descending order of priority to search, and return in JSON format `{"ordered_regions": [ordered_region_1, ordered_region_2, ...]}` , do not reply markdown format.

USER: `{'regions': ['entertainment area', 'washing area', 'kitchen', 'bar', 'office table'], 'target_object': 'controller' }`

ASSISTANT: `{'ordered_regions': ['entertainment area', 'bar', 'office table', 'kitchen', 'washing area']}`

B List of Symbols

Symbol	Description
$\mathcal{L}$	User instruction in natural language
g_o	Target object to fetch
g_R	Target region where the target object is located
$\mathcal{I}_{\text{RGB}}$	Image data from RGB-D camera
$\mathcal{I}_{\text{D}}$	Depth data from RGB-D camera
T^{cam}	Homogeneous transformation indicating the global pose of the RGB-D camera
t	Time stamp of a timed sequence, $0 \leq t \leq T$
T	Maximum time stamp of a timed sequence
k	Key frame index, $k \in \{1, 2, \cdots, K\}$
K	Total number of key frames
$p_{\text{cam}}^{(k,j)}$	The j-th homogeneous 3D point in camera frame from the k-th key frame
$\mathbf{P}_{\text{cam}}^{k}$	Matrix containing all 3D points in camera frame as columns from the k-th key frame
$\mathbf{P}^{k}$	Matrix containing all 3D points in global frame as columns from the k-th key frame
$b^{(k,i)}$	The i-th object instance detected at the k-th key frame with label and bounding box
$m^{(k,i)}$	Pixel mask for the i-th object instance detected at the k-th key frame
N_k	Total number of object instances detected at the k-th key frame
$\mathbf{P}^{(k,i)}$	Matrix containing all 3D points associated with the i-th instance at the k-th key frame
$\bar{p}^{(k,i)}$	Geometric center of the i-th instance at the k-th key frame
$\bar{\mathbf{P}}^{(k,i)}$	Relative geometry of the i-th instance at the k-th key frame
$q^{(k,i)}$	The i-th semantic geometric instance extracted at the k-th key frame
r	Radius of the circular sliding window for region semantics abstraction
Δd	Step size distance by which the sliding window moves for region semantics abstraction
s_x	Sliding window swiping step along the x-axis
s_y	Sliding window swiping step along the y-axis
$R(x, y)$	Region semantics suggesting a label of the region containing coordinate (x, y)
Searchable(R)	A list of searchable locations in region R
q^*	Semantic geometric instance with the highest confidence to pick up
n_e	Maximum number of trails for grasping at a location

C Experiment Details

C.1 Experiment Setup

Table 4: Default object placement in different regions

Region	Default Objects
Entertainment Area	controller, toy, charger, snacks
Washing Area	sponge, cloth, cup, spray cleaner
Cooking Area	ketchup, milk powder
Bar	bottled water, cup, milk, soda
Office Table	marker, stapler, pen, tape, mouse, bottle glue

Table 5: Division of Experiment Groups

Exp. Group	# of Eps.	Description
NoHint	45	User instructions do not contain region hints
Random	30	Control group, where region search orders are randomized
Hinting	30	User instructions contain region hints
ErrantSemantics	15	Target objects are placed in semantic irrelevant regions
Misleading	15	User instructions contain misleading region hints

C.2 Experiment Result

.

Table 6: Detailed result of our robot experiment

Exp. Group	SFT	SN	SP	Succ.	SPL
NoHint	33/45 (73.33%)	34/45 (75.56%)	33/45 (73.33%)	33/45 (73.33%)	0.6878
Random	7/30 (23.34%)	26/30 (86.67%)	23/30 (76.67%)	23/30 (76.67%)	0.5332
Hinting	30/30 (100.00%)	25/30 (83.33%)	21/30 (70.00%)	21/30 (70.00%)	0.6841
ErrantSemantics	0/15 (0.00%)	13/15 (86.67%)	13/15 (86.67%)	13/15 (86.67%)	0.4942
Misleading	0/15 (0.00%)	13/15 (86.67%)	10/15 (66.67%)	10/15 (66.67%)	0.4519