HomeRobot: Open-Vocabulary Mobile Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: HomeRobot (*noun*): An affordable compliant robot that navigates homes and manipulates a wide range of objects in order to complete everyday tasks.

Open-Vocabulary Mobile Manipulation (OVMM) is the problem of picking *any* object in *any* unseen environment, and placing it in a commanded location. This is a foundational challenge for robots to be useful assistants in human environments, because it involves tackling sub-problems from across robotics: perception, language understanding, navigation, and manipulation are all essential to OVMM. In addition, integration of the solutions to these sub-problems poses its own substantial challenges. To drive research in this area, we introduce the HomeRobot OVMM benchmark, where an agent navigates household environments to grasp novel objects and place them on target receptacles. HomeRobot has two components: a *simulation* component, which uses a large and diverse curated object set in new, high-quality multi-room home environments; and a *real-world* component, providing a software stack for the low-cost Hello Robot Stretch to encourage replication of real-world experiments across labs. We implement both reinforcement learning and heuristic (model-based) baselines and show evidence of sim-to-real transfer. Our baselines achieve a 20% success rate in the real world; our experiments identify ways future research work improve performance. See videos on our website: https://home-robot-ovmm.github.io/.

Keywords: Sim-to-real, benchmarking robot learning, mobile manipulation

1 Introduction

The aspiration to develop household robotic assistants has served as a north star for roboticists since the beginning of the field. The pursuit of this vision has spawned multiple areas of research within robotics from vision to manipulation, and has led to increasingly complex tasks and benchmarks. A useful household assistant requires creating a capable mobile manipulator that understands a wide variety of objects, how to interact with the environment, and how to intelligently explore a world with limited sensing. This has separately motivated research in diverse areas like navigation [1, 2], service robotics [3–5], language understanding [6, 7] and task and motion planning [8]. We refer to this guiding problem as *Open-Vocabulary Mobile Manipulation (OVMM):* a useful robot will be able to find and move arbitrary objects from place to place in an arbitrary home.

Prior work does not tackle mobile manipulation in large, continuous, real-world environments. Instead, it generally simplifies the setting significantly, e.g. by using discrete action spaces, limited object sets, or small, single-room environments that are easily explored. However, recent developments tying language and vision have enabled robots to generalize beyond specific categories [9–13], often through multi-modal models such as CLIP [14]. Further, comparison across methods has remained difficult and reproduction of results across labs impossible, since many aspects of the settings (environments, and robots) have not been standardized. This is especially important now, as a new wave of research projects have begun to show promising results in complex, open-vocabulary

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

Figure 1: Open-Vocabulary Mobile Manipulation requires agents to search for a previously unseen object at a particular location, and move it to the correct receptacle.

navigation [9, 15, 11, 12, 16] and manipulation [17, 10, 18] – again on a wide range of robots and settings, and still limited to single-room environments. Clearly, now is the time when we need a common platform and benchmarks to drive the field forward.

In this work, we define Open-Vocabulary Mobile Manipulation as a key task for in-home robotics and provide benchmarks and infrastructure, both in simulation and the real world, to build and evaluate full-stack integrated mobile manipulation systems, in a wide variety of human-centric environments, with open object sets. Our benchmark will further reproducible research in this setting, and the fact that we support arbitrary objects will enable the results to be deployed in a variety of real-world environments.

OVMM: We propose the first reproducible mobile-manipulation benchmark for the real world, with an associated simulation component. In simulation, we use a dataset of 200 human-authored interactive 3D scenes [19] instantiated in the AI Habitat simulator [20, 21] to create a large number of challenging, multi-room OVMM problems with a wide variety of objects curated from a variety of sources. Some of these objects' categories have been seen during training; others have not. In the real world, we create an equivalent benchmark, also with a mix of seen and unseen object categories, in a controlled apartment environment. We use the Hello Robot Stretch [22]: an affordable and compliant platform for household and social robotics that is already in use at over 40 universities and industry research labs. Fig. 1 shows instantiations of our OVMM task in both the real-world benchmark and in simulation. We have a controlled real-world test environment, and plan to run the real world benchmark yearly to assess progress on this challenging problem.

HomeRobot: We also propose HomeRobot,[1] a software framework to facilitate extensive benchmarking in both simulated and physical environments. It comprises identical APIs that are implemented across both settings, enabling researchers to conduct experiments that can be replicated in both simulated and real-world environments. Table 1 compares OVMM+HomeRobot to the literature.

In this paper, we use HomeRobot to compare two families of approaches: a *heuristic* solution, using a motion planner shown to work for real-world object search [2], and a *reinforcement learning* (RL) solution, which learns how to navigate to objects given depth and predicted object segmentation. We use the open-vocabulary object detector DETIC [23] to provide object segmentation for both the heuristic and RL policies. We observe that while the RL methods moved to the object more efficiently if an object was visible, the heuristic planner was better at long-horizon exploration. We also see

[1] Code will be released at the time of final submission.

		Scenes	Object Cats	Object Inst.	Continuous Actions	Sim2Real	Robotics Stack	Open Licensing	Manipulation
Room Rearrangement	[24]	120	118	118	✖	✖	✖	✔	✖
Habitat ObjectNav Challenge	[25]	216	6	7,599	✔	✖	✖	✔	✖
TDW-Transport	[26]	15	50	112	✖	✖	✖	✓	✓
VirtualHome	[27]	6	308	1,066	✖	✖	✖	✔	✓
ALFRED	[6]	120	84	84	✖	✖	✖	✔	✓
Habitat 2.0 HAB	[21]	105	20	20	✔	✖	✖	✔	✔
ProcTHOR	[28]	10,000	108	1,633	✖	✖	✖	✔	✔
RoboTHOR	[29]	75	43	731	✖	✔	✖	✔	✖
Behavior-1K	[30]	50	1,265	5,215	✔	✔	✖	✖	✓
ManiSkill-2	[31]	1	2,000	2,000	✔	✓	✖	✓	✔
OVMM + HomeRobot		200	150	7,892	✔	✔	✔	✔	✔

Table 1: Comparisons of our proposed benchmark with prior work. We provide a large number of environments with a continuous action space, and uniquely provide a real-world robotics stack with demonstrated sim-to-real capabilities, allowing others to reproduce and deploy their own solutions. Additional nuances in footnote[3]. ✓Partial availability ✖Not available ✔Capability available

a substantial drop in performance in switching from from ground-truth segmentation to DETIC segmentation. This highlights the importance of the OVMM challenge, as only through viewing the problem holistically - integrating perception, planning, and action - can we build general-purpose home assistants.

To summarize, in this paper, we define Open-Vocabulary Mobile Manipulation as a new, crucial task for the robotics community in Sec. 3. We provide a new simulation environment, with multiple, multi-room interactive environments and a wide range of objects. We implement a robotics library called HomeRobot which provides baseline policies implementing this in both the simulation and the real world. We describe a real-world benchmark in a controlled environment, and show how current baselines perform in simulation and in the real world under different conditions.

2 Related Work

We discuss work related to challenges and reproducibility of robotics research in more detail, but continue the discussion of datasets and simulators in Appendix A.

Challenges. There have been several challenges aiming to benchmark robotic systems at different tasks. These challenges provided a great testbed for ranking different systems. However, in most of the challenges (e.g., [32–35, 3]), the participants create their own robotic platform making a fair comparison of the algorithms difficult. There are also challenges where the organizers provide the robotic platform to the participants (e.g., [36]). However, changing the task during the periodic evaluations made it difficult to track progress over time. Our aim is to have a real world benchmark using a standard hardware that is sustainable at least for a few years.

Reproducibility of robotics research. Standardized robotics benchmarks have been pursued for a long time, often by open-sourcing robot designs or introducing low-cost robots [37–45]. However, the environments in which these robots are used vary dramatically, leading to evaluation of components (e.g., object navigation, SLAM) in isolation, instead of as components of a larger system that may not benefit from those changes. The HomeRobot stack enables end-to-end benchmarking of individual components by providing a full robotics stack, with multiple implementations of different sub-modules. The simplicity helps move beyond standardized sets of objects (e.g., [46–48]) to a common set of robots, objects, and environments.

Real-World Benchmarks. RoboTHOR [29] provides a common set of scenes and objects for benchmarking navigation. RB2 [49] ranks different manipulation algorithms in a local setting. TOTO [50] takes a step further by providing a training dataset and running the experiments for

[3]ALFRED uses object masks for interaction. ObjectNav uses scans, not full object meshes. ProcThor scenes are procedurally generated, this has the benefit that the potential number of environments is unbounded.

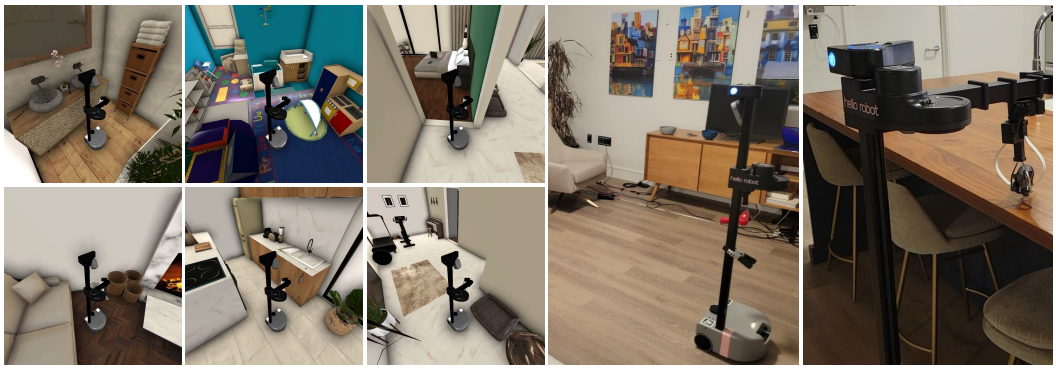

Figure 2: A low-cost home robot performing tasks in both a simulated and a real-world environment. We provide both (1) challenging simulated tasks, wherein a mobile manipulator robot must find and grasp multiple seen and unseen objects, and (2) a corresponding real-world robotics stack to allow others to reproduce this research and evaluation to produce useful home robot assistants.

the users. However, training and testing happen in the same environments and are limited to tabletop manipulation. Finally, the NIST Task Board [51] is a successful challenge for fine-grained manipulation skills [52], also limited to a tabletop context. Kadian et al. [53] propose the Habitat-PyRobot bridge (HaPy) to allow real-world testing on the locobot robot; their framework is limited to navigation, and doesn't provide a generally-useful robotics stack with visualizations, debugging, motion planners, tooling, etc.

3 Open-Vocabulary Mobile Manipulation

Formally, our task is set up as instructions of the form: "Move (`object`) from the (`start_receptacle`) to the (`goal_receptacle`)." The `object` is a small and manipulable household object (e.g., a cup, stuffed toy, or box). By contrast, `start_receptacle` and `goal_receptacle` are large pieces of furniture, which have surfaces upon which objects can be placed. The robot is placed in an unknown single-floor home environment - such as an apartment - and must, given the language names of `start_receptacle`, `object`, and `goal_receptacle`, pick up an `object` that is known to be on a `start_receptacle` and move it to any valid `goal_receptacle`. `start_receptacle` is always available, to help agents know where to look for the `object`.

The agent is successful if the specified `object` is indeed moved from a `start_receptacle` on which it began the episode, to any valid `goal_receptacle`. We give partial credit for each step the robot accomplishes: finding the `start_receptacle` with the `object`, picking up the `object`, finding the `goal_receptacle`, and placing the `object` on the `goal_receptacle`. There can be multiple valid objects that satisfy each query.

Crucially, we need and develop both (1) a simulation version of this OVMM problem, for reproducibility, training, and fast iteration, and (2) a real-robot stack with a corresponding real-world benchmark. We compare the two in Fig. 2. Our simulated environments allow for varied, long-horizon task experimentation; our real-world HomeRobot stack allows for experimenting with real data, and we design a set of real-world tests to evaluate the performance of our learned and heuristic baselines.

The Robot. We use the Hello Robot Stretch [22] with DexWrist as the mobile manipulation platform, because it (1) is *relatively* affordable at $\$25,000$ USD, (2) offers 6 DoF manipulation, and (3) is human safe and human-sized, making it safe to test in labs [54, 11] and homes [2], and can reach most places a human would expect a robot to go. For a breakdown on hardware choices, see Sec. G.1.

Objects. These are split into *seen* vs. *unseen categories* and *instances*. In particular, at test time we look at unseen instances of seen or unseen categories; i.e. no seen manipulable object from training appears during evaluation. Agents must pick and place any requested object.

Receptacles. We include common household receptables (e.g. tables, chairs, sofas) in our dataset; unlike with manipulable objects, all possible receptacle categories are seen during training.

Scenes. We have both a simulated scene dataset, and a fixed set of real-world scenes with specific furniture arrangements and objects. In both simulated and real scenes, we use a mixture of objects from *previously-seen* categories, and objects from *unseen* categories as the goal `object` for our Open-Vocabulary Mobile Manipulation task. We hold out *validation* and *test* scenes, which do not appear in the training data; while some receptacles may re-appear, they will be at previously-unseen locations, and target object instances will be unseen.

Scoring. We compute success for each stage: finding `object` on `start_receptacle`, successfully picking up `object`, finding `goal_receptacle`, and placing `object` on the goal. Overall success is true if all four stages were accomplished. We also compute a single *partial success* metric as a tie-breaker, in which agents receive 1 point for each successive stage accomplished per episode, normalized by the number of stages. More details in Appendix B.

3.1 Simulation Dataset

The Habitat Synthetic Scenes Dataset (HSSD) [19] consists of 200+ human authored 3D home scenes containing over 18k individual models of real-world objects. Like most real houses, these scenes are cluttered with furniture and other objects placed into realistic architectural layouts, making navigation and manipulation similarly difficult to the real world. We used a subset of HSSD [19] consisting of 60 scenes (e.g. Fig. 3) for which additional metadata and simulation structures were authored to support rearrangement [4]. For our experiments these are divided into train, validation, and test splits of 38, 12, and 10 scenes each, following the splits in the original HSSD paper [19].

Objects and Receptacles. We aggregate objects from AI2-Thor [55], Amazon-Berkeley Objects [56], Google Scanned Objects [57] and the HSSD [19] dataset to create a large and diverse dataset of real-world robot problems. In total, we annotated 2,535 objects from 129 total categories.We identified 21 different categories of receptacle which appear in the HSSD dataset [19].

Figure 3: Example (object free) top-down view from HSSD [19]. See App. Figs. 7 & 9 for navigation and viewpoints.

We construct our final set of furniture receptacle objects by first automatically labeling stable areas on top of receptacles, then manually refining and processing these in order to remove invalid or inaccessible receptacles. In addition, collision proxy meshes were automatically generated and in many cases manually corrected to support physically accurate procedural placement of object arrangements.

Episode Generation. We generate episodes consisting of varying object arrangements and particular values for `object`, `start_receptacle`, and `goal_receptacle`, which allow our agent to successfully move about and interact with the world. In the case of Open-Vocabulary Mobile Manipulation, this task is particularly challenging because we have to place objects in locations which are *navigable*, meaning that the robot can get to them, *reachable*, meaning its arm can make it to these locations, and from which we can navigate to a *navigable*, *reachable* goal receptacle. For full episode generation details see App. C.2.

	SC, SI	SC, UI	UC, UI	Total
Cats	85	64	44	129
Insts	1,363	748	424	2,535

Table 2: # of objects in each split: (S)een, (U)nseen, (I)nstance, and (C)ategory in simulation.

Training and Validation Split. Training episodes consist of objects from the large pool of seen instances of seen categories *(SC,SI)*. In contrast, we use unseen instances of seen object categories *(SC,UI)* and unseen instances of unseen categories *(UC,UI)* for validation and test episodes. Two-thirds of the categories were randomly designated as seen, and two-thirds of the objects in the seen category were randomly marked as seen instances. Split sizes are in Table 2 and the distribution of objects across categories is in App. Fig. 5.

[4] All 200+ scenes with rearrangement support will be released at the time of final submission.

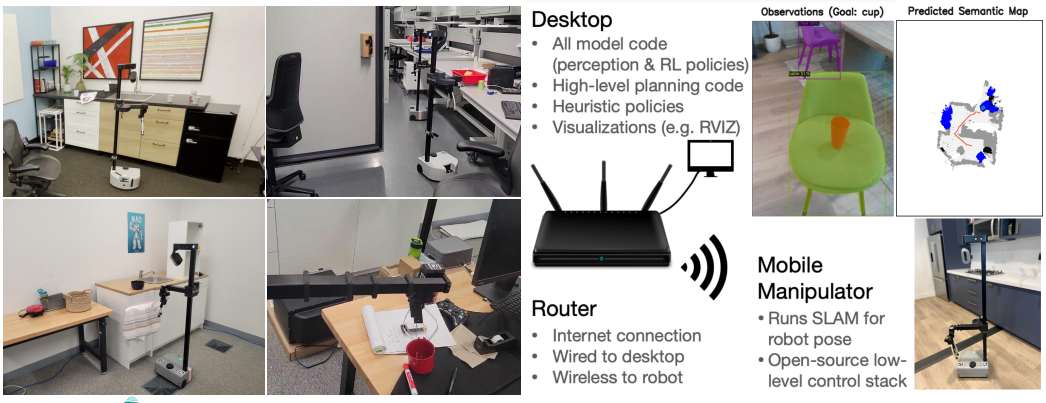

Figure 4: HomeRobot is a simple, easy-to-set-up library which works in multiple environments and requires only relatively affordable hardware. Computationally intensive operations are performed on a desktop PC with a GPU, and a dedicated consumer-grade router provides a network interface to a robot running low-level control and SLAM.

3.2 Real-World Benchmark

Real-world experiments are performed in a controlled 3-room apartment environment, with a sofa, kitchen table, counter with bar, and TV stand, among other features. We documented the positioning of various objects and the robot start position, in order to ensure reproducibility across trials. Images of various layouts of the test apartment are included in Fig. 2, and task execution is shown in Fig. 13.

During real-world testing, we selected a pool of object instances that did not appear during simulation training, but split between classes that did and did not appear in training. We used eight different categories, of which five were seen during training (*Cup*, *Bowl*, *Stuffed Toy*, *Medicine Bottle*, and *Toy Animal*), and three were not (*Rubik's cube*, *Toy Drill*, and *Lemon*). We performed 20 experiments on the Stretch robot for each of our two different baselines and with seven different receptacle classes: *Cabinet*, *Chair*, *Couch*, *Counter*, *Sink*, *Stool*, *Table*.

4 The HomeRobot Library

To facilitate research on these challenging problems, we open-source the HomeRobot library, which implements navigation and manipulation capabilities supporting Hello Robot's Stretch [22]. In our setup, it is assumed that users have access to a mobile manipulator and a NVIDIA GPU powered workstation. The mobile manipulator runs the low-level controller and the localization module, while the desktop runs the high-level perception and planning stack(Fig. 4). The robot and desktop are connected using an off-the-shelf router[5]. HomeRobot is designed as a user-friendly software stack, enabling quick setup of the robot for immediate testing. The key features of our stack include:

Transferability: Unified state and action spaces between simulation & real-world settings for each task, providing an easy way to control a robot with either high-level action spaces (e.g., pre-made grasping policies) or low-level continuous joint control.
Modularity: Perception and action components to support high-level states (e.g. semantic maps, segmented point clouds) and high-level actions (e.g. go to goal position, pick up target object).
Baseline Agents: Policies that use these capabilities to provide basic functionality for OVMM.

4.1 Baseline Agent Implementation

Crucially, we provide baselines and tools that enable researchers to effectively explore the Open-Vocabulary Mobile Manipulation task. We include two types of baselines in HomeRobot: a heuristic baseline, in which we use a well known motion planning technique [2] and simple rules to execute grasping and manipulation actions; and a reinforcement learning baseline, where we learn exploration

[5] Our experiments used a NetGear Nighthawk router.

Simulation Results	Skill			Partial Success Rates			Overall Success Rate	Partial Success Metric
Perception	Navigation	Gaze	Place	FindObj	Pick	FindRec		
Ground Truth	Heuristic	None	Heuristic	46.2	39.5	18.6	6.9	27.3
	Heuristic	RL	RL	47.2	41.7	27.1	19.7	32.5
	RL	None	Heuristic	55.1	41.9	26.4	6.5	32.2
	RL	RL	RL	55.7	50.2	35.2	21.0	39.8
DETIC [23]	Heuristic	None	Heuristic	23.3	11.5	3.0	0.3	9.5
	Heuristic	RL	RL	24.8	9.5	5.0	0.7	10.0
	RL	None	Heuristic	19.9	10.2	4.4	0.8	8.8
	RL	RL	RL	19.8	11.8	6.3	1.5	9.8

Table 3: We observe that one of the main causes of failures for our baseline systems was perception failures; ground-truth performance is notably higher. We also see that both RL and heuristic skills struggled navigating tightly constrained multi-room environments and successfully placing objects.

and manipulation skills using an off-the-self policy learning algorithm, DDPPO [58]. Due to the challenging, long-horizon nature of the task, we implement a high-level policy called `OVMMAgent` which calls a sequence of skills to accomplish a task. We breakdown our agents into four skills:

FindObj/FindRec: Locate an `object` on a `start_receptacle`; or find a `goal_receptacle`.
Gaze: Move close enough to an `object` to grasp it, and orient head to get a good view of the object. The goal of the gaze action is to improve the success rate of grasping.
Grasp: Pick up the `object`. We provide a high-level action for this, since we do not simulate the gripper interaction in Habitat. However, our library is compatible with a range of learned grasping skills and supports learning policies for grasping.
Place: Move to a location in the environment and place the `object` on top of the `goal_receptacle`.

Heuristic. We implement a version using only off-the-shelf learned models and heuristics, noting that previous work in mobile manipulation has used these models to great effect (e.g. [59]). Here, DETIC [60] provides masks for an open-vocabulary set of objects as appropriate for each skill. The `start_receptacle`, `object`,`goal_receptacle` for each episode is given. Fig. 13 shows an example of the heuristic navigation and place policy being executed in the real world (App. D).

RL. We train the four skills in our modified version of Habitat [21] as policies which predict actions given depth, ground truth semantic segmentation and priopreceptive sensors (i.e. joints, gripper state), using DDPPO [58]. While RGB is available in our simulation, our baseline policies do not directly utilize it; instead, they rely on predicted segmentation from Detic [23] at test time.

5 Results

We first evaluate the two baselines in our simulated benchmark, followed by evaluation in a real-world, held-out test apartment. These results highlight the significance of OVMM as a challenging new benchmark, encompassing numerous essential challenges that arise when deploying robots in real-world environments.

We break down the results by sub-task in addition to reporting the overall performance in Tables 3 and 4. The columns **FindObj**, **Pick** and **FindRec** refer to the first 3 phases of the task mentioned in the scoring section (Sec. 3), and succeeding in the final Place phase leads to a successful episode.

Simulation. We evaluate the baselines on held-out scenes, with objects from unseen instances of seen classes, and unseen instances of *unseen* classes, as described in Sec. 3.1. We show results with two different perception systems: **Ground Truth** segmentation, where we use the segmentation input directly from the simulator, and **DETIC** segmentation [23], where the RGB images from the simulator are passed through DETIC, an open-vocabulary object detector.

We report results in Table 3 broken down by skill. The results show that RL policies outperformed heuristic methods for both navigation and placement tasks. However, all policies experienced a decline in performance when the perception is changed from ground truth to DETIC. Notably, heuristic policies exhibited less degradation in performance compared to RL policies under DETIC

Real World	FindObj	Pick	FindRec	Overall Success
Heuristic Only	0.70	0.35	0.30	0.15
RL Only	0.70	0.45	0.30	0.20

Table 4: Results for heuristic and RL baseline in the real world on the OVMM task. In both cases, the grasping action is executed as described in Sec. 4; but initial conditions of the robot such as its position relative to the object or to other obstacles may cause various failures.

perception. With DETIC perception, the heuristic FindObj policy outperforms RL. We attribute this to the heuristic policy's ability to incorporate noisy predictions by constructing a 2D semantic map, which proves advantageous in handling small objects that are prone to misclassification. Furthermore, we observed that the learned gaze policy generally led to improved pick performance, except when used in combination with the Heuristic nav with DETIC perception. For additional information, example simulation trajectories can be found in Appendix Figure 15, and results comparing seen versus unseen categories are discussed in Appendix F.2.

Real-World. Finally, we conducted a series of experiments in a real-world held-out apartment setting. We performed a total of 20 episodes, utilizing a combination of seen and unseen object classes as our target objects. The results of these experiments are presented in Table 4. RL performed slightly better than the Heuristic baseline, successfully completing 1 extra episode and achieving a success rate of 20%. This difference primarily stemmed from the pick and place sub-tasks. In the pick task, the RL Gaze skill plays a crucial role in achieving better alignment between the agent and the target object, which led to more successful grasping. Similarly, the RL place skill demonstrated more precision, ensuring that the object stayed closer to the surface of the receptacle.

Both simulation and real-world results show the baselines are promising, but insufficient, for Open-Vocabulary Mobile Manipulation. DETIC [23] caused many failures due to misclassification, both in simulation and the real world. Further, RL navigation was on par or better than heuristic policies in both sim and real. Although our RL place policy performed better in sim than heuristic place, it needs further improvement in the real world. Gaining the advantages of webscale pretrained vision-language models like DETIC, but tuned to our agents may be crucial for improving performance.

6 Limitations

Our benchmark has a few key limitations: (1) Due to simulation limitations, we don't physically simulate grasping in the first version, which is why we provide a separate policy for this in the real world. Grasping is a well-studied problem [61–63], but simulations that train useful real-world grasp systems require special consideration. (2) We consider full natural language queries out-of-scope. Finally, (3) we do not implement many motion planners in HomeRobot (see Sec. D.2), or task-and-motion-planning with replanning, as would be ideal [64].

7 Conclusions and Future Work

We proposed a combined simulation and real-world benchmark to enable progress on the important problem of Open-Vocabulary Mobile Manipulation. We ran extensive experiments showing promising simulation and real-world results from two baselines: a heuristic baseline based on a state-of-the-art motion planner [2] and a reinforcement learning baseline trained with DDPPO [58]. In the future, we hope to improve the complexity of the problem space, adding more complex natural language and multi-step commands instead of pick-and-place, and provide end-to-end baselines instead of modular policies. Various proposed solutions for open-vocabulary navigation [9, 11, 12] and manipulation of unknown objects [10, 13, 18, 17] suggest possible ways of improving performance.

References

[1] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv*, 2020.

[2] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot. Navigating to objects in the real world. *arXiv*, 2022.

[3] T. Wisspeintner, T. Van Der Zant, L. Iocchi, and S. Schiffer. Robocup@ home: Scientific competition and benchmarking for domestic service robots. *Interaction Studies*, 2009.

[4] J. Bohren, R. B. Rusu, E. G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mösenlechner, W. Meeussen, and S. Holzer. Towards autonomous robotic butlers: Lessons learned with the pr2. In *ICRA*, 2011.

[5] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial intelligence*, 1999.

[6] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *CVPR*, 2020.

[7] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. Virtualhome: Simulating household activities via programs. In *CVPR*, 2018.

[8] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:265–293, 2021.

[9] 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. *arXiv*, 2022.

[10] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, B. Zitkovich, F. Xia, C. Finn, et al. Open-world object manipulation using pre-trained vision-language models. *arXiv*, 2023.

[11] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv*, 2022.

[12] B. Bolte, A. Wang, J. Yang, M. Mukadam, M. Kalakrishnan, and C. Paxton. Usa-net: Unified semantic and affordance representations for robot memory. *arXiv*, 2023.

[13] A. Curtis, X. Fang, L. P. Kaelbling, T. Lozano-Pérez, and C. R. Garrett. Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances. In *ICRA*, 2022.

[14] 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 *ICML*, 2021.

[15] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv*, 2023.

[16] J. Krantz, E. Wijmans, A. Majundar, D. Batra, and S. Lee. Beyond the nav-graph: Vision and language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, 2020.

[17] W. Liu, T. Hermans, S. Chernova, and C. Paxton. Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects. *arXiv*, 2022.

[18] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. *arXiv*, 2023.

[19] M. Khanna*, Y. Mao*, H. Jiang, S. Haresh, B. Schacklett, D. Batra, A. Clegg, E. Undersander, A. X. Chang, and M. Savva. Habitat Synthetic Scenes Dataset: An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation. *arXiv*, 2023.

[20] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. *ICCV*, 2019.

[21] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. In *NeurIPS*, 2021.

[22] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich. The design of stretch: A compact, lightweight mobile manipulator for indoor human environments. In *ICRA*, 2022.

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

[24] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi. Visual room rearrangement. In *CVPR*, 2021.

[25] K. Yadav, J. Krantz, R. Ramrakhya, S. K. Ramakrishnan, J. Yang, A. Wang, J. Turner, A. Gokaslan, V.-P. Berges, R. Mootaghi, O. Maksymets, A. X. Chang, M. Savva, A. Clegg, D. S. Chaplot, and D. Batra. Habitat challenge 2023. https://aihabitat.org/challenge/2023/, 2023.

[26] C. Gan, J. Schwartz, S. Alter, D. Mrowca, M. Schrimpf, J. Traer, J. De Freitas, J. Kubilius, A. Bhandwaldar, N. Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *NeurIPS Datasets and Benchmarks Track*, 2021.

[27] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. Virtualhome: Simulating household activities via programs. In *CVPR*, 2018.

[28] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, J. Salvador, K. Ehsani, W. Han, E. Kolve, A. Farhadi, A. Kembhavi, and R. Mottaghi. Procthor: Large-scale embodied ai using procedural generation. In *NeurIPS*, 2022.

[29] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, L. Weihs, M. Yatskar, and A. Farhadi. RoboTHOR: An Open Simulation-to-Real Embodied AI Platform. In *CVPR*, 2020.

[30] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *CoRL*, 2023.

[31] T. Mu, Z. Ling, F. Xiang, D. C. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. In *NeurIPS Datasets and Benchmarks Track*, 2021.

[32] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orlowski. The darpa robotics challenge finals: Results and perspectives. *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue*, 2018.

[33] G. Seetharaman, A. Lakhotia, and E. P. Blasch. Unmanned vehicles come of age: The darpa grand challenge. *Computer*, 2006.

[34] M. Buehler, K. Iagnemma, and S. Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*. Springer Berlin, Heidelberg, 2009.

[35] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman. Analysis and observations from the first amazon picking challenge. *IEEE Transactions on Automation Science and Engineering*, 2016.

[36] L. D. Jackel, E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan. The darpa lagr program: Goals, challenges, methodology, and phase i results. *Journal of Field Robotics*, 2006.

[37] M. Müller and V. Koltun. Openbot: Turning smartphones into robots. In *ICRA*, 2021.

[38] N. Kau, A. Schultz, N. Ferrante, and P. Slade. Stanford doggo: An open-source, quasi-direct-drive quadruped. In *ICRA*, 2019.

[39] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, J. Fiene, A. Badri-Spröwitz, and L. Righetti. An open torque-controlled modular robot architecture for legged locomotion research. *IEEE Robotics and Automation Letters*, 2019.

[40] B. Yang, J. Zhang, V. H. Pong, S. Levine, and D. Jayaraman. Replab: A reproducible low-cost arm benchmark platform for robotic learning. *arXiv*, 2019.

[41] D. V. Gealy, S. McKinley, B. Yi, P. Wu, P. R. Downey, G. Balke, A. Zhao, M. Guo, R. Thomasson, A. Sinclair, P. Cuellar, Z. McCarthy, and P. Abbeel. Quasi-direct drive for low-cost compliant robotic manipulation. In *ICRA*, 2019.

[42] M. Wüthrich, F. Widmaier, F. Grimminger, S. Joshi, V. Agrawal, B. Hammoud, M. Khadiv, M. Bogdanovic, V. Berenz, J. Viereck, M. Naveau, L. Righetti, B. Schölkopf, and S. Bauer. Trifinger: An open-source robot for learning dexterity. In *CoRL*, 2020.

[43] M. Ahn, H. Zhu, K. Hartikainen, H. Ponte, A. Gupta, S. Levine, and V. Kumar. ROBEL: RObotics BEnchmarks for Learning with low-cost robots. In *CoRL*, 2019.

[44] A. Murali, T. Chen, K. V. Alwala, D. Gandhi, L. Pinto, S. Gupta, and A. K. Gupta. Pyrobot: An open-source robotics framework for research and benchmarking. *arXiv*, 2019.

[45] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cáp, Y. F. Chen, C. Choi, J. Dusek, Y. Fang, D. Hoehener, S. Liu, M. M. Novitzky, I. F. Okuyama, J. Pazis, G. Rosman, V. Varricchio, H.-C. Wang, D. S. Yershov, H. Zhao, M. R. Benjamin, C. Carr, M. T. Zuber, S. Karaman, E. Frazzoli, D. D. Vecchio, D. Rus, J. P. How, J. J. Leonard, and A. Censi. Duckietown: An open, inexpensive and flexible platform for autonomy education and research. In *ICRA*, 2017.

[46] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *ICRA*, 2015.

[47] D. Morrison, P. Corke, and J. Leitner. Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation. *IEEE Robotics and Automation Letters*, 2020.

[48] B. Yang, P. E. Lancaster, S. S. Srinivasa, and J. R. Smith. Benchmarking robot manipulation with the rubik's cube. *IEEE Robotics and Automation Letters*, 2020.

[49] S. Dasari, J. Wang, J. Hong, S. Bahl, Y. Lin, A. S. Wang, A. Thankaraj, K. S. Chahal, B. Çalli, S. Gupta, D. Held, L. Pinto, D. Pathak, V. Kumar, and A. Gupta. Rb2: Robotic manipulation benchmarking with a twist. *arXiv*, 2022.

[50] G. Zhou, V. Dean, M. K. Srirama, A. Rajeswaran, J. Pari, K. B. Hatch, A. Jain, T. Yu, P. Abbeel, L. Pinto, C. Finn, and A. Gupta. Train offline, test online: A real robot learning benchmark. *arXiv*, 2022.

[51] K. Kimble, K. Van Wyk, J. Falco, E. Messina, Y. Sun, M. Shibata, W. Uemura, and Y. Yokokohji. Benchmarking protocols for evaluating small parts robotic assembly systems. *IEEE Robotics and Automation Letters*, 2020.

[52] W. Lian, T. Kelch, D. Holz, A. Norton, and S. Schaal. Benchmarking off-the-shelf solutions to robotic assembly tasks. In *IROS*, 2021.

[53] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra. Are we making real progress in simulated environments? measuring the sim2real gap in embodied visual navigation. *arXiv*, 2019.

[54] P. Parashar, J. Vakil, S. Powers, and C. Paxton. Spatial-language attention policies for efficient robot learning. *arXiv*, 2023.

[55] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. AI2-THOR: an interactive 3d environment for visual AI. *arXiv*, 2017.

[56] J. Collins, S. Goel, A. Luthra, L. Xu, K. Deng, X. Zhang, T. F. Y. Vicente, H. Arora, T. Dideriksen, M. Guillaumin, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *CVPR*, 2022.

[57] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, 2022.

[58] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *ICLR*, 2019.

[59] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser. Tidybot: Personalized robot assistance with large language models. *arXiv*, 2023.

[60] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: Learning view synthesis using multiplane images. *SIGGRAPH*, 2018.

[61] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *ICRA*, 2021.

[62] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox. 6-dof grasping for target-driven object manipulation in clutter. In *ICRA*, 2020.

[63] H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *CVPR*, 2020.

[64] C. R. Garrett, C. Paxton, T. Lozano-Pérez, L. P. Kaelbling, and D. Fox. Online replanning in belief space for partially observable task and motion problems. In *ICRA*, 2020.

[65] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[66] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019.

[67] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, 2020.

[68] M. Sap, H. Rashkin, D. Chen, R. LeBras, and Y. Choi. SocialIQA: Commonsense reasoning about social interactions. In *EMNLP*, 2019.

[69] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[70] S. Keisuke, L. B. Ronan, B. Chandra, and C. Yejin. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI*, 2019.

[71] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[72] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[73] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *ICRA*, 2016.

[74] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *IJRR*, 2018.

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

[76] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *CoRL*, 2018.

[77] P. Sharma, L. Mohan, L. Pinto, and A. K. Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. In *CoRL*, 2018.

[78] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv*, 2017.

[79] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning. *arXiv*, 2019.

[80] A. Gupta, A. Murali, D. P. Gandhi, and L. Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. In *NeurIPS*, 2018.

[81] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. D. Reid, S. Gould, and A. van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2017.

[82] H. Team. Habitat CVPR challenge, 2019. URL https://aihabitat.org/challenge/2019/.

[83] F. Xia, W. B. Shen, C. Li, P. Kasimbeg, M. Tchapmi, A. Toshev, R. Martín-Martín, and S. Savarese. Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 2020.

[84] C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *ECCV*, 2020.

[85] K. Ehsani, W. Han, A. Herrasti, E. VanderBilt, L. Weihs, E. Kolve, A. Kembhavi, and R. Mottaghi. Manipulathor: A framework for visual object manipulation. In *CVPR*, 2021.

[86] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2019.

[87] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020.

[88] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, 2020.

[89] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramithu, G. Tur, and D. Hakkani-Tur. TEACh: Task-driven embodied agents that chat. In *AAAI*, 2022.

[90] X. Gao, Q. Gao, R. Gong, K. Lin, G. Thattai, and G. S. Sukhatme. Dialfred: Dialogue-enabled agents for embodied instruction following. *IEEE Robotics and Automation Letters*, 2022.

[91] A. Szot, K. Yadav, A. Clegg, V.-P. Berges, A. Gokaslan, A. Chang, M. Savva, Z. Kira, and D. Batra. Habitat rearrangement challenge. https://aihabitat.org/challenge/2022_rearrange, 2022.

[92] C. M. Kim, M. Danielczuk, I. Huang, and K. Goldberg. Simulation of parallel-jaw grasping using incremental potential contact models. In *ICRA*, 2022.

[93] D. Hall, B. Talbot, and N. Sünderhauf. The robotic vision challenges. https://nikosuenderhauf.github.io/roboticvisionchallenges/cvpr2022, 2022.

[94] A. Mousavian, C. Eppner, and D. Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *ICCV*, 2019.

[95] C. Paxton, C. Xie, T. Hermans, and D. Fox. Predicting stable configurations for semantic placement of novel objects. In *CoRL*, 2022.

[96] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.

[97] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *ICRA*, 2000.

[98] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *NeurIPS*, 2020.

[99] B. Yamauchi. A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997.

[100] J. A. Sethian. Fast marching methods. *SIAM review*, 1999.

[101] D. S. Chaplot, S. Gupta, A. Gupta, and R. Salakhutdinov. Learning to explore using active neural mapping. *ICLR*, 2020.

[102] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.