REALISTIC MOBILE MANIPULATION TASKS FOR EVALUATING HOME-ASSISTANT ROBOTS

A Dissertation
Presented to
The Academic Faculty

By

Sriram Venkata Yenamandra

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in
Computer Science
College of Computing

Georgia Institute of Technology

December 2023

REALISTIC MOBILE MANIPULATION TASKS FOR EVALUATING HOME-ASSISTANT ROBOTS

Thesis committee:

Dr. Dhruv Batra
Department of Interactive Computing
Georgia Institute of Technology

Dr. Judy Hoffman
Department of Interactive Computing
Georgia Institute of Technology

Dr. Zsolt Kira
Department of Interactive Computing
Georgia Institute of Technology

Date approved: December 11, 2023

Our greatest weakness lies in giving up. The most certain way to succeed is always to try just one more time.

Thomas Edison

For Amma and Nanna

ACKNOWLEDGMENTS

I extend my heartfelt appreciation to the members of my thesis committee, Prof. Dhruv Batra, Prof. Judy Hoffman, and Prof. Zsolt Kira, for generously dedicating their time to provide valuable insights into my work.

I would like to express my thanks to the people who played a crucial role in the completion of this thesis. I express my gratitude to my advisor, Prof. Dhruv Batra, whose support and trust enabled me to pursue research in his lab. Dhruv's insightful feedback and advice during project and team presentations have significantly influenced the development of my work. I am particularly thankful for the trust he placed in me, allowing me to extend my stay at Georgia Tech.

Working in Dhruv's CVMLP lab connected me with highly diligent, competent and passionate individuals. My sincere appreciation goes to Harsh Agrawal and Yash Kant for their belief in my abilities, allowing me to join the lab mid-semester and kickstarting my research journey at Georgia Tech. I am also grateful to Arun Ramachandran for his contributions to Housekeep and OVMM. Thanks to Andrew Szot for assistance with Habitat and RL training, Mukul for his contributions to the OVMM and GOAT projects, and Karmesh for being a mentor and a constant source of technical and life-related advice.

I express my gratitude to Chris Paxton, the project lead of OVMM and my mentor, for his exemplary leadership and enthusiasm. I appreciate his trust, consistent support, and invaluable guidance throughout my research endeavors. Similarly, I appreciate the contributions of all other collaborators from Meta. Special thanks to Theophile Gervet and Devendra Chaplot for providing me the opportunity to contribute to the GOAT project.

Beyond the scope of this thesis, I appreciate the influence of individuals who enriched my journey at Georgia Tech. Prof. Judy Hoffman, whose course inspired my research pursuit, deserves thanks for allowing me to join her lab and for her continued support. I express my appreciation to Viraj for serving as a mentor. His assistance and guidance

greatly facilitated my work in all the projects within Judy's lab. Special thanks to Aaditya Singh, Prithvijit Chattopadhyay and Pratik Ramesh, close collaborators in my projects in Judy's lab, and to all lab members from both labs for transforming my desk at Coda building into a home away from home.

Finally, I express deep appreciation to my parents for their numerous sacrifices, allowing me to stay away from home for over two years. I am thankful for their understanding of my schedule and the limited time I could devote to them. My father's life lessons and my mother's advice have been invaluable pillars of support, and I am grateful for their unwavering concern for my well-being.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

By assisting in household chores, robotic home assistants hold the potential to significantly enhance the quality of human lives. Mobile manipulation tasks can serve as test beds for evaluating the capabilities essential to the development of robotic home assistants: perception, language understanding, navigation, manipulation and common-sense reasoning. However, it is imperative to use settings that closely resemble real-world deployment to ensure that progress made on these tasks is practically relevant.

The thesis introduces three tasks with the objective of realising the different dimensions of realism critical for evaluating embodied agents. These dimensions are: 1) autonomy, the ability to operate without very specific instructions (*e.g.* the precise locations of goal objects), 2) exposure to realistic novel multi-room environments, 3) working with previously unseen objects, and 4) extended durations of deployment.

The first task, HomeRobot Open Vocabulary Mobile Manipulation, involves moving a novel object from one receptacle to another based solely on its name. The HomeRobot stack allows a simulated agent to be benchmarked on a real robot operating in a real home. The second task, “GO To Any Thing”, entails navigating to a series of multimodal goals in unseen environments by leveraging past experience in the same environment. The last task, Housekeep, focuses on tidying up a household without any instructions. For solving this task, the agent must reason if an object is misplaced and identify correct locations for it. The thesis explores the extent to which these tasks fulfill the dimensions of realism.

The thesis proposes baselines for solving these tasks incorporating heuristic and learned components, and using large-scale pretrained models for detecting novel objects or reasoning about them. These baselines solve each task to a varying degree and their shortcomings underscore the open challenges of open-vocabulary object detection and common-sense reasoning. By using test scenarios closer to real-world deployment, this work attempts to advance research in the development of robotic assistants.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Picture this: you've just thrown a major party, and the aftermath is clear — plates and wine glasses scattered around the dining table, some on the floor near the couch, and a board game left on the coffee table. Now, imagine the ease of a robotic assistant autonomously cleaning up the mess, putting plates and glasses in the kitchen sink and neatly storing the board game in the bedroom cabinet. Imagine its capability to help you locate a misplaced medicine bottle given an image or verbal description amidst the chaos. And wouldn't it be ideal if these assistants benefit from their experience within the environment, reducing the need for extensive exploration the next time they need to reach previously visited destinations like kitchen sink?

Achieving this vision involves requires creating a capable mobile manipulator that can interpret multi-modal goal specifications, understand a wide variety of objects, interact with the environment, intelligently explore a world with limited sensing and retain memory of the environment as it explores. Rearrangement tasks [1], wherein the agent must transform an environment from one state to another by moving objects, can serve as effective test beds for assessing progress across a combination of these abilities.

Although previous studies have introduced various rearrangement tasks [2, 3, 4, 5, 6], their realism is compromised by their evaluation in simulated scenes, reliance on single-room environments, adoption of a closed object category set, limitation of episode length to pick-place interactions, or dependence on geometric coordinates for target specification. To tackle this challenge, four desiderata are established to enhance the realism in evaluations:

- **Autonomy:** Compared to Geometric Goal-based benchmarks [2, 4], it is crucial to assess the robotic assistants' ability to rearrange goals with minimal supervision, showcasing their independence and decision-making skills.

- **Real-robot benchmarking:** To bridge the gap between simulation and the real world, evaluation results should not only be limited to simulation environments but also demonstrate adaptability to real robots operating in real environments.
- **Novel object categories:** The agent should be capable of handling previously unseen instances and categories. The ability to handle objects extends beyond rearranging them, including tasks such as recognizing objects, understanding their properties, and effectively manipulating them to achieve the desired rearrangement.
- **Lifelong evaluation:** Assessing whether the agent improves over time with familiarity in the environment requires evaluations on longer-horizon tasks.

To address these challenges, three tasks have been proposed, aiming to realize the notions of realism in the evaluation process. We provide an overview of these tasks next.

1.2 OVMM: Reproducible sim2real benchmark

Let's delve into the scenario of cleaning up a post-party mess from above. Imagine the difficulty of specifying instructions to an assistant using geometric coordinates, such as saying "Pick the object 2m to the east and 3m north" to pick up a plate on the dining table. Instead, we explore a benchmark called HomeRobot Open Vocabulary Mobile Manipulation (OVMM) where the agent is tasked with rearranging an object, identified by its name from one receptacle to another (*e.g.* Move a plate from dining table to sink). This benchmark enables zero-shot sim2real benchmarking to ensure that progress made in simulation is practically meaningful. In simulation, we use a dataset of 60 human-authored interactive 3D scenes [7] instantiated in the AI Habitat simulator [8, 2] to create a large number of challenging, multi-room OVMM problems with a wide variety of

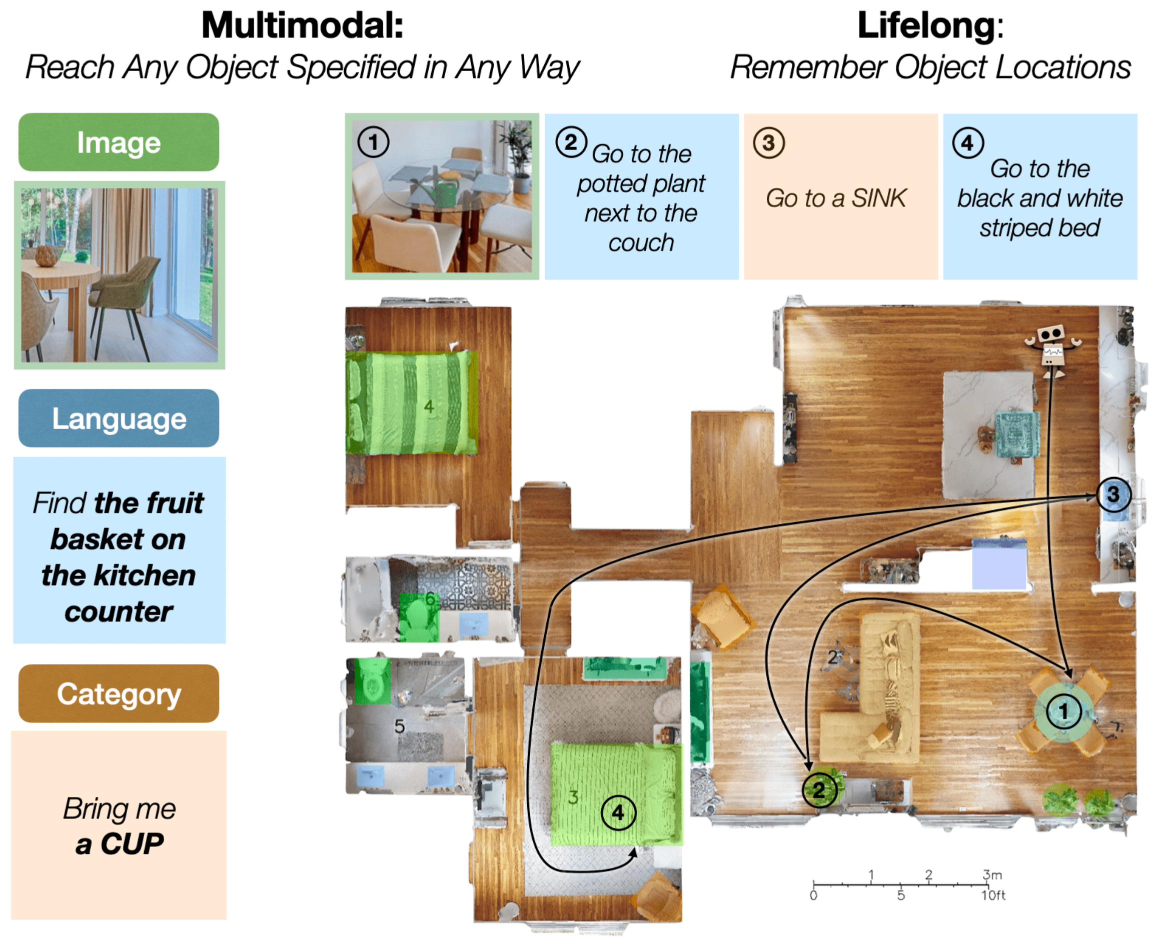
Multimodal:
Reach Any Object Specified in Any Way
Lifelong:
Remember Object Locations
Image
Language
Find the fruit basket on the kitchen counter
Category
Bring me a CUP
1
2 Go to the potted plant next to the couch
3 Go to a SINK
4 Go to the black and white striped bed
4
5
6
3
2
7
0 1 2 3m
5 10ft

Figure 1.1: **GOAT (GO to Any Thing) task.** The GOAT task requires lifelong learning, meaning taking advantage of past experience in the environment, for multimodal navigation. The robot must be able to reach an object specified in any way and remember object locations to come back to them.

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.

1.3 GOAT: Multimodal lifelong evaluation

Again consider the running example of cleaning up the post-party mess, imagine if you could just ask the a freshly deployed agent to reach a dining table for tidying just by providing an image (*e.g.* captured via an AR headset) of it (goal 1 in Figure 1.1).

Navigating to this goal requires recognizing that the picture shows a dining table and having the semantic understanding of indoor spaces to efficiently explore the home (e.g. dining tables are not found in the bathroom). Next, you ask the robot to *Go to the potted plant next to the couch* (goal 2), for picking up the wine glasses placed close by. This requires visual grounding of the text instruction in the physical space. The next instruction could be to *Go to a SINK* (goal 3), the capitalization emphasizing that any object of the category SINK is a valid goal. In this example (Figure 1.1), the robot has already seen a sink in the house during the first task, so it should remember its location and be able to plan a path to reach it efficiently. This requires the robot to build, maintain and update a lifelong memory of the objects in the environment, their visual and linguistic properties and their latest location. Given any new multimodal goal, the robot should also be able to query the memory to determine whether the goal object already exists in the memory or requires further exploration. We develop the GOAT: Go To Any Thing task, where the agent is given a sequence of goals specified via language, images or category names, to evaluate the multimodal perception, exploration and lifelong memory capabilities.

1.4 Housekeep: Complete autonomy but otherwise not realistic

In OVMM and GOAT tasks, the agent operates solely in response to the user's commands, relying on goals expressed in language, image, or as a category. However, consider the scenario where the agent could independently tidy up the post-party mess from the recurring example *without explicit instructions*: spotting dirty dishes on the dinner table and moving them to the sink, recognizing a displaced board game on the coffee table and relocating it to the cabinet. We introduce the Housekeep task to benchmark the ability of embodied AI agents to use physical commonsense reasoning and infer rearrangement goals that mimic human-preferred placements of objects in indoor environments. A robot is randomly spawned in an unknown house that contains unseen objects. Without explicit instructions, the agent must then discover objects placed in the house, classify the misplaced ones, and finally rearrange them to one of many suitable receptacles. Given the challenging nature of task, to scope the problem and focus on planning and common-sense reasoning, we take a step back from strict realism and mitigating certain aspects of it. Specifically, we assume privileged access to instance and semantic sensor among other assumptions and restrict evaluations to simulated environments.

1.5 Solutions and results on mobile manipulation tasks

Our proposed solutions mainly involved frontier-based exploration and semantic mapping [9]. In the context of OVMM, we additionally explore training navigation and manipulation policies via reinforcement learning (RL). We use DETIC [10] to detect open-vocabulary of objects in OVMM and MaskRCNN [11] to detect the fixed set of COCO categories used in GOAT. In GOAT, we make use of a map-based instance memory to track all detected instances across different goals in an episode, while language goal matching uses CLIP [12], and image goal matching utilizes SuperGLUE [13]. In Housekeep, we attempt to leverage models pre-trained on large-scale internet data to extract commonsense reasoning on likely locations of objects.

In experimental comparisons spanning over 90 hours in 9 different homes consisting of 675 goals selected across 200+ different object instances, we find GOAT achieves an overall success rate of 83%, surpassing previous methods by 32% (absolute improvement). On OVMM, our RL-based baseline achieves a 20% success rate on real robots and our modular baseline for Housekeep achieves 23% object success rate for correctly rearranging misplaced unseen objects. However, the performance of our proposed approaches remains modest on OVMM and Housekeep. The baselines struggle with detecting open-vocabulary of objects and identifying human-preferred locations for misplaced objects respectively. These findings show that the baselines require further advancements, mainly in perception and commonsense reasoning skills for realizing autonomous household robotic assistants. The thesis makes the following key contributions:

- Outlines **dimensions of realism** critical to evaluating mobile manipulators, namely: autonomy, handling unseen object categories, real-robot and lifelong evaluations.
- Introduces the **first reproducible sim2real benchmark**, HomeRobot Open Vocabulary Mobile Manipulation, along with two other tasks GOAT and Housekeep, aimed at realizing the different dimensions of realism.
- Proposes baselines for solving these tasks, notably achieving **83% success rate** (32% absolute improvement over prior work) on the real-world GOAT task benefiting from a semantic-aware instance memory for navigating to a series of multimodal goals.

1.6 Thesis Outline

The rest of thesis is aimed towards introducing the tasks, discussing the baselines and key results. This is followed by a discussion around the key learnings from these works and the remaining challenges that need to be addressed for building fully autonomous agents. The thesis is structured as below:

- Chapter 2: Related Work explores existing research in embodied AI tasks, real-world benchmarks, and commonsense reasoning.
- Chapter 3 delves into HomeRobot: Open Vocabulary Mobile Manipulation, detailing the task, baselines, reinforcement learning, and results.
- Chapter 4 presents the "GOAT: GO To Any Thing" task, methodology, and results, emphasizing impressive performance in the real world.
- Chapter 5 discusses the Housekeep task, human preferences dataset, baselines, and results, showcasing the application of LLMs for commonsense reasoning.
- Chapter 6: Discussion critically examines different tasks along the dimensions of realism.
- Chapter 7: Conclusion summarizes key findings and contributions of the thesis.

CHAPTER 2
RELATED WORK

2.1 Embodied AI Tasks

In recent times, we have seen a proliferation of Embodied AI tasks. Benchmarks on indoor navigation use point-goal specification [8, 14], object-goal [15, 16], room navigation [17], and language-guided navigation [18, 19]. Some interactive tasks study the agent's ability to follow natural language instruction such as ALFRED [20] and TEACh [6] while others focus on rearranging objects following a geometric goal or predicate based specification [3, 4, 2, 21]. [1] provides a summary of rearrangement tasks. All these tasks require an explicit goal specification lifting the burden of learning semantic compatibility of objects and their locations in the house from the agent, like in GOAT and OVMM. In contrast, Housekeep deals with tidying up the house without requiring an explicit goal specification.

2.2 Real World Benchmarks

RoboTHOR [22] provides a common set of scenes and objects for benchmarking navigation. RB2 [23] ranks different manipulation algorithms in a local setting. TOTO [24] takes a step further by providing a training dataset and running the experiments for the users. However, training and testing happen in the same environments and are limited to tabletop manipulation. Finally, the NIST Task Board [25] is a successful challenge for fine-grained manipulation skills [26], also limited to a tabletop context. Kadian et al. [27] 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 planner or tooling.

2.3 Commonsense Reasoning

Prior work in Natural Language Processing has studied the problem of imbuing commonsense knowledge in AI systems, from social common-sense knowledge [28, 29, 30, 31, 32, 33] to understand the likely intents, goals, and social dynamics of people, abductive commonsense reasoning [34], next event prediction [35, 36], to temporal common sense knowledge about temporal order, duration, and frequency of events [37, 38, 39, 40]. Most similar to our work is the study of physical commonsense knowledge [41] about object affordances, interaction, and properties (such as flexibility, curvature, porousness). However, these benchmarks are static in nature (as a dataset of textual or visual prompts). On the other hand, in this thesis, we consider tasks that are instantiated in an embodied partially-observed environment, and the agent has to explore unseen regions, discover misplaced objects and use common-sense reasoning to infer compatibility between objects and receptacles.

CHAPTER 3

HOMEROBOT: OPEN VOCABULARY MOBILE MANIPULATION

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.

3.1 Task

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. Figure 3.1 shows instantiations of our OVMM task in both the real-world benchmark and in simulation.

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 simulated version of the Open-Vocabulary Mobile Manipulation problem, for reproducibility, training, and fast iteration, and (2) a real-robot stack with a corresponding real-world benchmark. Our simulated environments

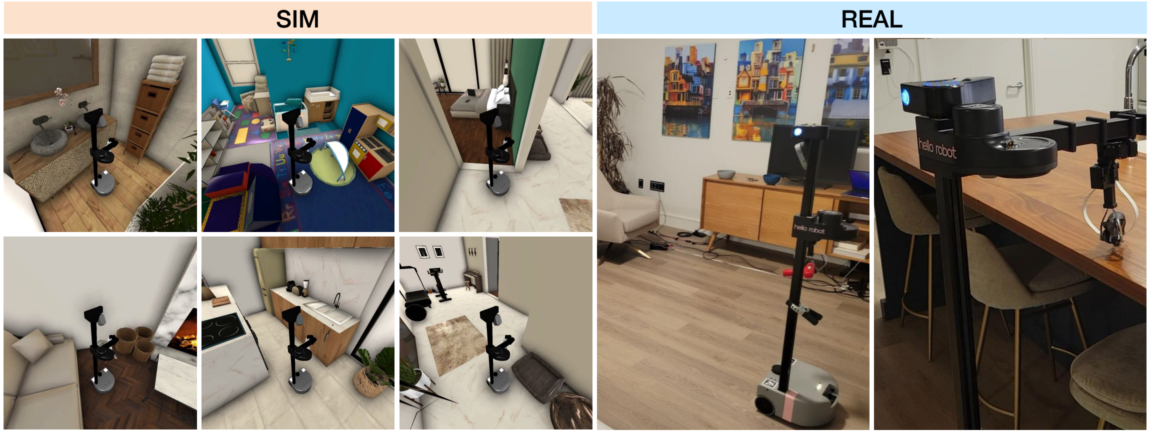

Figure 3.1: A low-cost home robot performing tasks in both a simulated and a real-world environment. We provide both (1) challenging simulated tasks, wherein a mobile manipulator robot must find and grasp multiple seen and unseen objects, and (2) a corresponding real-world robotics stack to allow others to reproduce this research and evaluation to produce useful home robot assistants.

allow for varied, long-horizon task experimentation; our real-world HomeRobot stack allows for experimenting with real data.

The Robot. We use Hello Robot Stretch [42] with DexWrist as the mobile manipulation platform, because it (1) is *relatively* affordable at $25k USD, (2) offers 6 DoF manipulation, and (3) is human safe and human-sized, making it safe to test in labs [43, 44] and homes [9], and can reach most places a human would expect a robot to go.

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.

Receptacles. We include common household receptacles (*e.g.* tables, chairs) 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

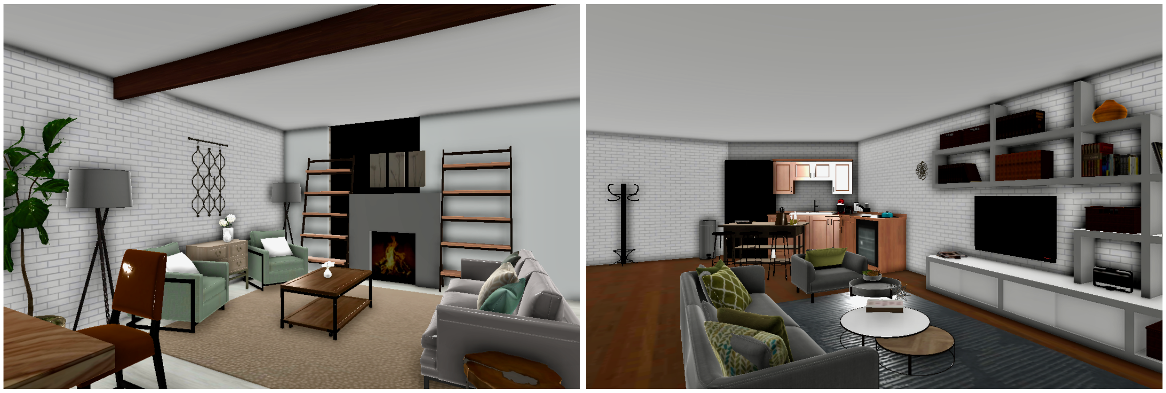

Figure 3.2: HSSD scenes.

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 compute *partial success* as a tie-breaker, in which agents receive 1 point for each successive stage accomplished, normalized by the number of stages.

3.1.1 Simulation Dataset

The Habitat Synthetic Scenes Dataset (HSSD) [7] consists of 200+ human-authored 3D home scenes containing over 18k 3D 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 [7] consisting of 60 scenes for which additional metadata and simulation structures were authored to support rearrangement 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 [7].

Objects and Receptacles. We aggregate objects from AI2-Thor [45], Amazon-Berkeley Objects [46], Google Scanned Objects [47] and the HSSD [7] dataset to create a large and diverse dataset of real-world robot problems.

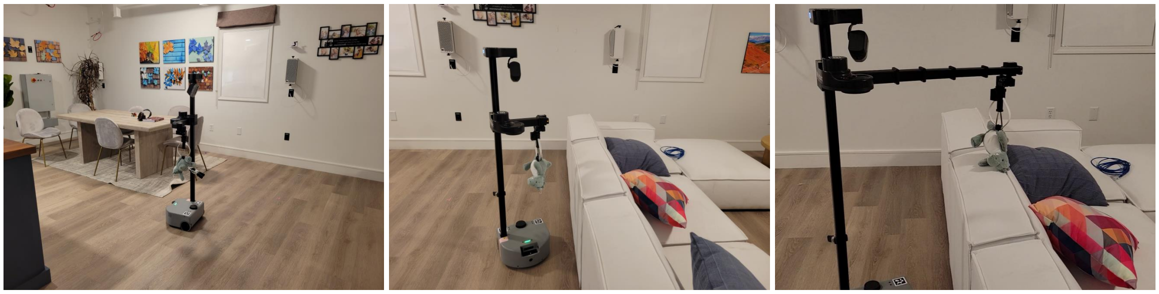

Figure 3.3: An example of the robot navigating to a `goal_receptacle` (sofa) and using the heuristic place policy to put down the `object` (stuffed animal). Heuristic policies provide an interpretable and easily extended baseline.

In total, we annotated 2,535 objects from 129 total categories.We identified 21 different categories of receptacles which appear in the HSSD dataset [7]. 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.

3.1.2 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 Figure 3.1, and task execution is shown in Figure 3.3.

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

3.2 Baselines

3.2.1 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, using motion planning [9] and simple rules for manipulation; and a reinforcement learning baseline. We implement a high-level policy called `OVMMAgent` which calls a sequence of skills one after the other. These skills are:

- **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, to improve the success rate of grasping.
- **Pick:** 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`.

Specifically, `OVMMAgent` is a state-machine that calls **FindObj**, **Gaze**, **Pick**, **FindRec**, and **Place** in that order, where **Pick** is a grasping policy provided by the robot library in the real world. The other skills are created using the approaches given below:

3.2.2 Heuristic Baseline

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. [48]). Here, DETIC [49] 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. Figure 3.3 shows an example of the heuristic navigation and place policy being executed in the real world.

3.2.3 Reinforcement Learning Baseline

We train the four skills: `FindObject`, `FindReceptacle`, `GazeAtObject`, and `PlaceObject` in our modified version of Habitat [2].

Action Space

- **Navigation Skills** `FindObject` and `FindReceptacle` are, collectively, navigation skills. For these two skills, we use discrete action space. We found that discrete action space was better at exploration and easier to train.

- **Manipulation Skills** For our manipulation skills, we using a continuous action space to give the skills fine grained control. In the real world, low-level controllers have limits on the distance the robot can move in any particular step. Thus, in simulation, we limit our base action space by only allowing forward motions between 10-25 cm, or turning by 5-30 degrees in a single step. The head tilt, pan and gripper's yaw, roll and pitch can be changed by at most 0.02-0.1 radians in a single step. The arm's extension and lift can be changed by at most 2-10cm in a single step. We learn by *teleporting* the base and arm to the target locations.

Observation Space

Policies have access to depth from the robot head camera, and semantic segmentation, as well as the robot's pose relative to the starting pose (from SLAM in the real world), camera pose, and the robot's joint states, including the gripper. RGB image is available to the agent but not used during training.

Training Setup

All skills are trained using a slack reward (-0.005 per step), incentivizing completion of task using minimum number of steps. For faster training, we learn our policies using images with a reduced resolution of 160x120 (compared to Stretch's original resolution of 640x480).

- **Navigation Skills:** We train `FindObject` and `FindReceptacle` policies for the agent to reach a candidate object or a candidate target receptacle respectively. The training procedure is the same for both skills. We pass in the CLIP [12] embedding corresponding with the goal object, as well as segmentation masks corresponding with the detected target objects. The agent is spawned arbitrarily, but at least 3 meters from the target, and must move until within 0.1 meters of a goal "viewpoint," where the object is visible.

- **`GazeAtObject`:** The `GazeAtObject` skill starts near the object and provides some final refinement steps until the agent is close enough to call a grasp action, i.e. it is in arm's length of the object and the object is centered and visible. The agent needs to move closer to the object and then adjust its head tilt until the candidate object is close and centered. It makes predictions to move and rotate the head, as well as to center the object and make sure it's within arm's length so that the discrete grasping policy can execute. The `GazeAtObject` skill is supposed to start off from locations and help reach a location within arm's length of a candidate object. This is trained by first initializing the agents close to candidate start receptacles. The agent is then tasked to reach close to the agent and adjust its head tilt such that the candidate object is close and centered in the agent's camera view.

- **`PlaceObject`**: Finally, the robot must move its arm in order to place the object on a free spot in the world. In this case, it starts at a viewpoint near a `goal_receptacle`. It must move up to the object and open its gripper in order to

Table 3.1: Partial and overall success rate (SR) (in %) for different combinations of skills and perception systems. The partial SR for each skill is dependent on the previous skill's SR. The partial SR for the place skill is the same as the overall SR. The partial success metric is calculated by averaging the 4 partial SRs.

Simulation Results	Skill			Partial Success Rates			Overall	Partial
Perception	Navigation	Gaze	Place	FindObj	Pick	FindRec	Success Rate	Success Metric
Ground Truth	Heuristic	None	Heuristic	54.1	48.5	31.5	5.1	34.8
	Heuristic	RL	RL	56.5	51.5	42.3	13.2	40.9
	RL	None	Heuristic	65.4	54.8	43.7	7.3	42.8
	RL	RL	RL	66.6	61.1	50.9	14.8	48.3
DETIC [10]	Heuristic	None	Heuristic	28.7	15.2	5.3	0.4	12.4
	Heuristic	RL	RL	29.4	13.2	5.8	0.5	12.2
	RL	None	Heuristic	21.9	11.5	6.0	0.6	10.0
	RL	RL	RL	21.7	10.2	6.2	0.4	9.6

Table 3.2: Success Rate (in %) for heuristic and RL baselines for real world OVMM.

Real World	FindObj	Pick	FindRec	Overall Success
Heuristic Only	70	35	30	15
RL Only	70	45	30	20

place the object on this surface. The episode succeeds if the agent releases the object and the object stays on the receptacle for 50 timesteps.

3.3 Results and Discussion

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 3.1 and 3.2. The columns **FindObj**, **Pick** and **FindRec** refer to the first 3 phases of the task mentioned in the scoring section (Section 3.1), 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 Section 3.1.1. We show results with two different perception systems: **Ground Truth** segmentation, where we use the segmentation input directly from the simulator, and **DETIC** segmentation [10], where the RGB images from the simulator are passed through DETIC, an open-vocabulary object detector.

We report results on HomeRobot OVMM in Table 3.1. RL policies outperformed heuristic methods for both navigation and placement tasks. However, all policies declined in performance when using DETIC instead of ground truth segmentation. Heuristic policies exhibited less degradation in performance compared to RL policies: when using DETIC, the heuristic FindObj policy even outperforms RL. We attribute this to the heuristic policy's ability to incorporate noisy predictions by constructing a 2D semantic map, which helps handle small objects that are prone to misclassification. Furthermore, using the learned gaze policy led to improved pick performance, except when used in combination with the Heuristic nav with DETIC perception. Example simulation trajectories can be found in Figure 3.4.

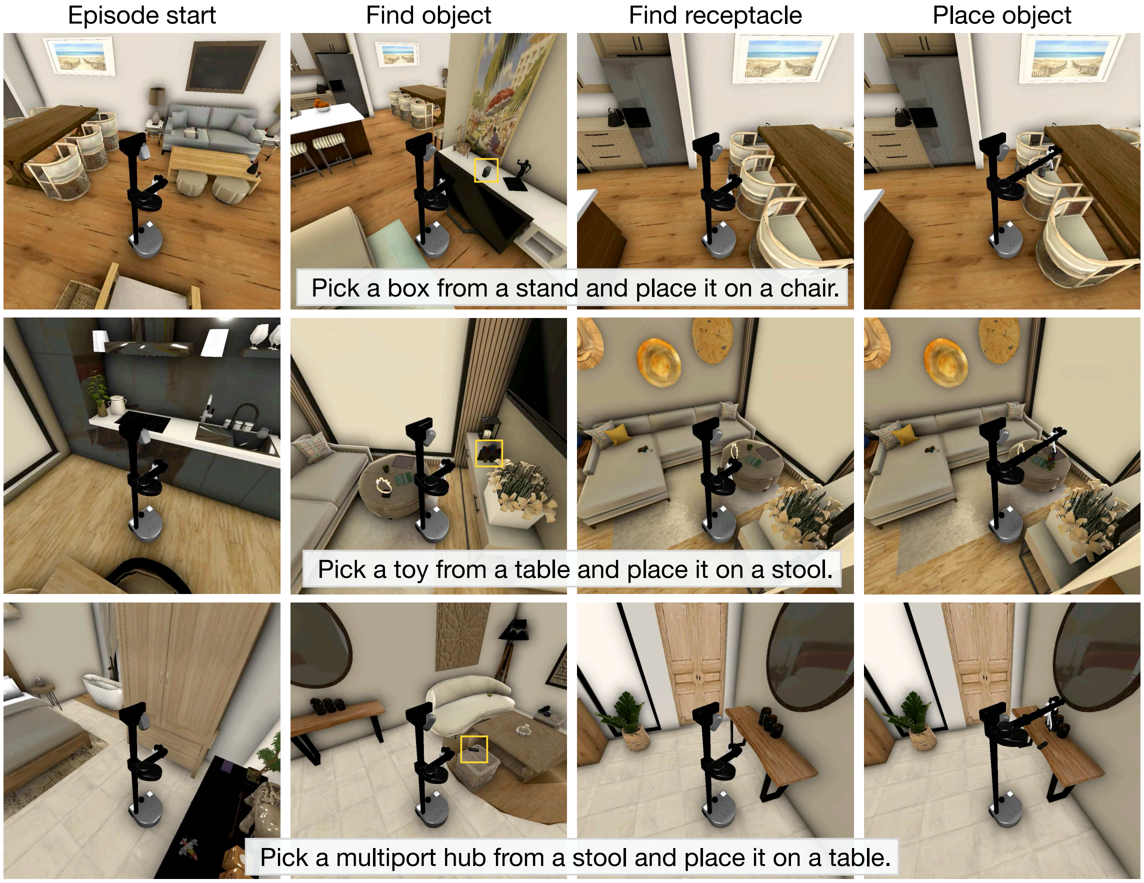
Episode start
Find object
Find receptacle
Place object
Pick a box from a stand and place it on a chair.
Pick a toy from a table and place it on a stool.
Pick a multiport hub from a stool and place it on a table.

Figure 3.4: We show multiple executions of the Open-Vocabulary Mobile Manipulation task in a variety of simulated environments.

CHAPTER 4
GOAT: GO TO ANY THING

In OVMM task, the goals were specified using object names and receptacle categories. However, general-purpose agents will need to handle **multimodal goal specifications**, such as language descriptions or video demonstrations. Likewise, agents deployed in real homes will be required to perform tasks that extend beyond a single OVMM pick-place episode. Towards building such lifelong multimodal agents, we propose the GOAT: Go To Any Thing task.

4.1 Task

We formalize the Go to Any Thing task T as follows. We construct navigation episodes consisting of a sequence of unseen goal objects to be reached in unseen environments. The robot is spawned at a random location. At every timestep t, the robot receives observations consisting of an RGB image I_t, depth image D_t, and pose reading x_t from onboard sensors, as well as the current object goal g_k, $k \in \{1, 2, .., 5 - 10\}$, which consists in an object category (*SINK*, *CHAIR*), an image or language description (*the potted plant next to the couch*, *the black and white striped bed*) uniquely identifying an object instance in the environment. The robot must reach the goal object g_k as efficiently as possible within a limited time budget. As soon as it reaches the current goal or when the time budget is exhausted, the robot receives the next goal to navigate to, g_{k+1}. In searching for this sequence of goals the agent is allowed to maintain a memory computed using incoming observations. In this way, if g_{k+1} has been observed during the process of reaching g_k the agent can often more efficiently navigate to g_{k+1}.

4.2 Method

GOAT Agent As the agent moves through the scene, the perception system processes RGB-D camera inputs to detect object instances and localize them into a top-down semantic map of the scene. In addition to the semantic map, GOAT maintains an Object Instance Memory (see Figure 4.1) that localizes individual instances of object categories in the map and stores images in which each instance has been viewed. This Object Instance Memory gives GOAT the ability to perform lifelong learning for multimodal navigation. When a new goal is specified to the agent, a global policy first searches the Object Instance Memory to see if the goal has already been observed. After an instance is selected, its stored location in the map is used as a long-term point navigation goal. If no instance is localized, the global policy outputs an exploration goal. A local policy finally computes actions towards the long-term goal.

Instance Matching Strategy The matching module of the global policy has to identify the goal object instance among previously seen object instances in the Object Instance Memory. We evaluated different design choices and settled on the following: match language goal descriptions with object views in memory using the cosine similarity score between their CLIP [12] features, match image goals with object views in memory using keypoint-based matching with SuperGLUE [13], represent object views in memory as bounding boxes with some padding to include additional context, match the goal only against instances of the same object category, match the goal with the instance with the maximum matching score across all views.

4.3 Results

Experimental Setting We evaluate the GOAT agent as well as three baselines in nine visually diverse homes with 10 episodes per home consisting of 5-10 object instances randomly selected out of objects available in the home, representing 200+ different object

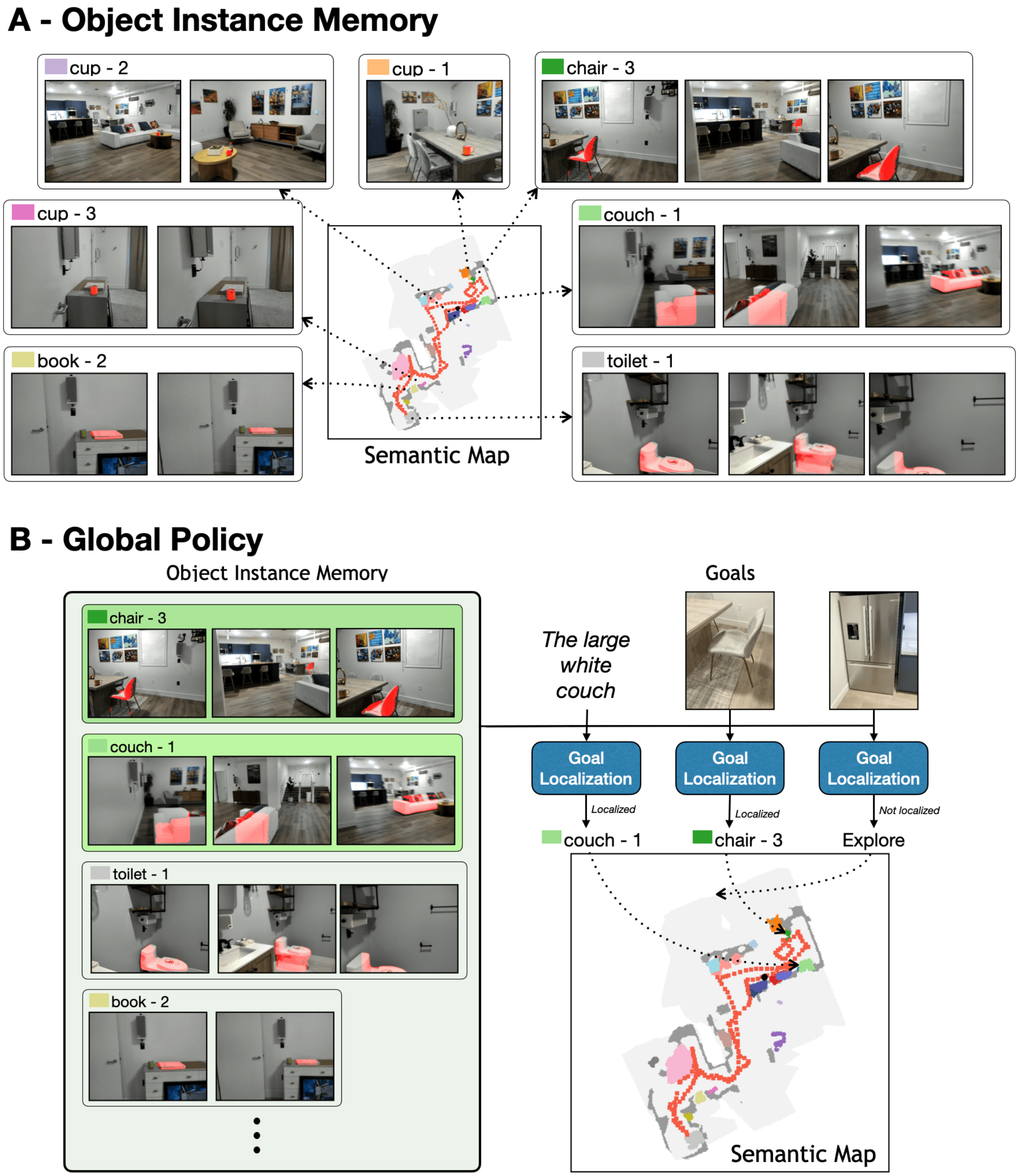

Figure 4.1: **(A) Object Instance Memory.** We cluster object detections, along with image views in which they were observed, into instances using their location in the semantic map and their category. **(B) Global Policy.** When a new goal is specified, the global policy first tries to localize it within the Object Instance Memory. If no instance is localized, it outputs an exploration goal.

instances in total. We selected goals across 15 different object categories (‘chair’, ‘couch’, ‘potted plant’, ‘bed’, ‘toilet’, ‘tv’, ‘dining table’, ‘oven’, ‘sink’, ‘refrigerator’, ‘book’, ‘vase’, ‘cup’, ‘bottle’, ‘teddy bear’), took a picture for image goals following the protocol in Krantz *et al.* [50], and annotated 3 different language descriptions uniquely identifying the object. To generate an episode within a home, we sampled a random sequence of 5-10 goals split equally among language, image, and category goals among all object instances available. We evaluate approaches in terms of success rate to reach the goal and SPL [51], which measures path efficiency as the ratio of the agent’s path length over the optimal path length. We report evaluation metrics per goal within an episode with two standard deviation error bars.

Baselines We compare GOAT to three baselines: **1. CLIP on Wheels** [52] - the existing work that comes closest to being able to address the GOAT problem setting - which keeps track of all images the robot has ever seen and, when given a new goal object, decides whether the robot has already seen it by matching CLIP [12] features of the goal image or language description with CLIP features of all images in memory, **2. GOAT w/o Instances**, an ablation that treats all goals as object categories, i.e., always navigating to the closest object of the correct category instead of distinguishing between different instances of the same category as in [53], allowing us to quantify the benefits of GOAT’s instance awareness, and **3. GOAT w/o Memory**, an ablation that resets the semantic map and Object Instance Memory after every goal, allowing us to quantify the benefits of GOAT’s lifelong memory.

Quantitative Results Table Table 4.1 reports metrics for each method aggregated over the 90 episodes. GOAT achieves 83% average success rate (94% for object categories, 86% for image goals, and 68% for language goals). We observed that localizing language goals is harder than image goals (detailed in the Discussions section). CLIP on Wheels [52] attains a 51% success rate, showing that using GOAT’s Object Instance

Table 4.1: Navigation Performance in Unseen Natural Home Environments. We compare GOAT to three baselines in 9 unseen homes with 10 episodes per home consisting of 5-10 image, language, or category goal object instances in terms of success rate and SPL [51], a measure of path efficiency, per goal instance.

	SR per Goal				SPL Per Goal			
	Image	Language	Category	Average	Image	Language	Category	Average
GOAT	**86.4 ± 1.1**	**68.2 ± 1.5**	**94.3 ± 0.8**	**83.0 ± 0.7**	**0.679 ± 0.013**	**0.511 ± 0.014**	**0.737 ± 0.010**	**0.642 ± 0.007**
CLIP on Wheels	46.1 ± 1.8	40.8 ± 1.9	65.3 ± 1.5	50.7 ± 1.0	0.368 ± 0.014	0.317 ± 0.013	0.569 ± 0.015	0.418 ± 0.008
GOAT w/o Instances	28.6 ± 1.7	27.6 ± 1.6	**94.1 ± 0.8**	49.4 ± 0.8	0.219 ± 0.013	0.222 ± 0.012	**0.739 ± 0.011**	0.398 ± 0.007
GOAT w/o Memory	59.4 ± 1.5	45.3 ± 1.6	76.4 ± 1.3	60.3 ± 0.8	0.193 ± 0.020	0.134 ± 0.022	0.239 ± 0.021	0.188 ± 0.012

Memory for goal matching is more effective than CLIP feature matching against all previously viewed images. GOAT w/o Instances achieves 49% success rate, with 29% and 28% success rates for image and language goals respectively. This demonstrates the need to keep track of enough information in memory to be able to distinguish between different object instances, which [53] wasn't able to do. GOAT w/o memory achieves 61% success rate with an SPL of only 0.19 compared to the 0.64 of GOAT. It has to re-explore the environment with every goal, explaining the low SPL and low success rate due to many time-outs. This demonstrates the need to keep track of a lifelong memory.

4.4 GOAT for Mobile Manipulation

The ability to perform rearrangement tasks is essential in any deployment scenarios for mobile robots (homes, warehouses, factories) [1, 54, 55, 56]. These are commands such as "pick up my coffee mug from the coffee table and bring it to the sink," requiring the agent to search for and navigate to an object, pick it up, search for and navigate to a receptacle, and place the object on the receptacle. The GOAT navigation policy can easily be combined with pick and place skills (we use built-in skills from Boston Dynamics) to fulfill such requests. We evaluate this ability on 30 such queries with image/language/category objects and receptacles across 3 different homes. GOAT can find objects and receptacles with 79% and 87% success rates, respectively.

CHAPTER 5

HOUSEKEEP: TIDYING VIRTUAL HOUSEHOLDS USING COMMONSENSE REASONING

We introduce the Housekeep task to benchmark the ability of embodied AI agents to use physical commonsense reasoning and infer rearrangement goals that mimic human-preferred placements of objects in indoor environments. Figure 5.1 illustrates our task, where the Fetch robot is randomly spawned in an unknown house that contains unseen objects. Without explicit instructions, the agent must then discover objects placed in the house, classify the misplaced ones (LEGO set and lunch bag in Figure 5.1), and finally rearrange them to one of many suitable receptacles (matching color-coded square).

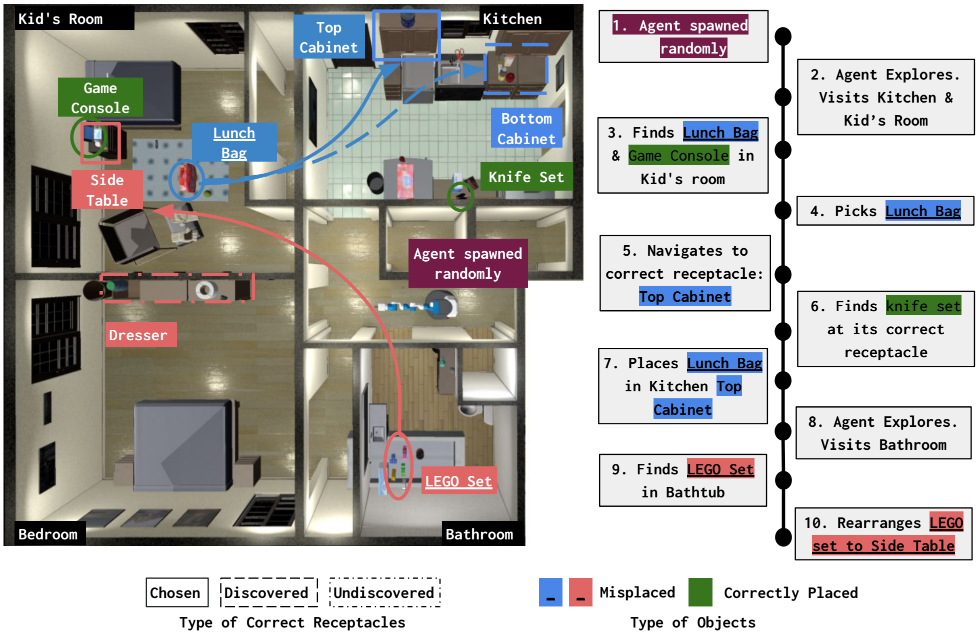
Kid's Room
Top Cabinet
Kitchen
Game Console
Lunch Bag
Bottom Cabinet
Side Table
Knife Set
Agent spawned randomly
Dresser
LEGO Set
Bedroom
Bathroom
1. Agent spawned randomly
2. Agent Explores. Visits Kitchen & Kid's Room
3. Finds Lunch Bag & Game Console in Kid's room
4. Picks Lunch Bag
5. Navigates to correct receptacle: Top Cabinet
6. Finds knife set at its correct receptacle
7. Places Lunch Bag in Kitchen Top Cabinet
8. Agent Explores. Visits Bathroom
9. Finds LEGO Set in Bathtub
10. Rearranges LEGO set to Side Table
Chosen
Discovered
Undiscovered
Type of Correct Receptacles
Misplaced
Correctly Placed
Type of Objects

Figure 5.1: In Housekeep, an agent is spawned in an untidy environment and tasked with rearranging objects to suitable locations without explicit instructions. The agent explores the scene and discovers misplaced objects, correctly placed objects, and receptacles where objects belong. The agent rearranges a misplaced object (like a lunch box on the floor in the kid's room) to a better receptacle like the top cabinet in the kitchen.

5.1 Task Specification

Definition: Recall, in Housekeepan embodied agent is required to clean up the house by rearranging misplaced objects to their correct location within a limited number of time steps. The agent is spawned randomly in an unseen environment and has to explore the environment to find misplaced objects and put them in their correct locations (receptacles).

Scenes and Rooms: We use 14 interactive and realistic iGibson scenes [57]. These scenes span 17 room types (*e.g.* living room, garage) and contain multiple rooms with an average of 7.5 rooms per scene. We remove one scene from the original iGibson dataset (`benevolence_0_int`) because it's unfurnished.

Receptacles: We define *receptacles* as flat horizontal surfaces in a household (furniture, appliances) where objects can be found – misplaced or correctly placed. We remove assets that are neither objects nor receptacles (*e.g.* windows, paintings, etc) and end up with 395 unique receptacles spread over 32 categories. An iGibson scene can contain between 19-78 receptacles. Overall, there are 128 distinct room-receptacles in the iGibson scenes.

Objects: We collect 1799 unique objects spread across 268 categories from four popular asset repositories – Amazon Berkeley Objects [46], Google Scanned Objects [58], ReplicaCAD [2], and YCB Objects [59]. We further categorize these objects into 19 high-level semantic categories such as stationery, food, electronics, toys, etc.

Agent: We simulate a Fetch robot [60], which has an RGBD camera (90° FoV, 128×128 pixels) on the robot's head. The robot moves its base and head through five discrete actions – move forward by 0.25m, rotate base right or left by 10°, rotate head camera up or down (pitch) by 10°. The robot interacts with objects through a "magic pointer abstraction" [1] where at any step the robot can select a discrete "interact" action. When invoked, this action casts a ray 1.5m in front of the agent. If the agent is not currently holding an object and this ray intersects with a graspable object, then the object is now "held" by the agent. If the agent is already holding an object and the ray intersects with a receptacle, then the object is placed on that receptacle. Rather than place the object at the point selected on the receptacle, the object is automatically placed on the receptacle.

Access to privileged information: The task assumes access to egocentric semantic and instance sensors, in addition to information on whether an object is on top of a receptacle and the room a given receptacle is located in.

5.2 Human Preferences Dataset: Where Do Objects Belong?

The central challenge of Housekeepis understanding how humans prefer to put everyday household objects in an organized and disorganized house. We want to capture where objects are typically found in an unorganized house (before tidying the house), and in a tidy house where objects are kept in their correct position (after the person has tidied the house). To this end, we run a study on Amazon MTurk [61, 62] with 372 participants. Each participant is shown an object (*e.g.* salt-shaker), a room (*e.g.* kitchen) for context, and asked to classify all the receptacles present in the room into the following categories:

- `misplaced`: subset of receptacles where object is found *before* housekeeping.
- `correct`: subset of receptacles where object is found *after* housekeeping.
- `implausible`: subset of receptacles where object is unlikely to be found either in a clean or an untidy house.

We also ask each participant to rank receptacles classified under `misplaced` and `correct`. For example, given a can of food, someone may prefer placing it in kitchen cabinets while others will rank pantry over the kitchen cabinet.

For each object-room pair (268 × 17), we collect 10 human annotations. We collect human annotations through multiple batches of smaller annotation tasks. In a single annotation task, we ask participants to classify-then-rank receptacles for 10 randomly sampled object-room pairs. On average a participant took 21 minutes to complete one annotation task. Overall, participants spent 1633 hours doing our study.

5.3 Baselines

5.3.1 Extracting Embodied Commonsense from LLMs

One of the main goals of Housekeep is to equip the agent with commonsense knowledge to reason about the compatibility of an object with different receptacles present across different rooms. Large Language Models (LLMs) trained on unstructured web-corpora have been shown to work well for several embodied AI tasks like navigation [63, 64, 65, 66, 67]. We study whether we can use LLMs to extract physical (embodied) common sense about how humans prefer to rearrange objects to tidy a house. For this, we build a ranking module (`L`) which takes as input a list of objects and a list of receptacles in rooms and then outputs a sequence of desired rearrangements based on which object receptacle pairings are most likely. We select the rearrangements that maximize $\mathbb{P}(\text{receptacle}, \text{room}|\text{object})$. We decompose computing this probability into a product of two probabilities:

- Object Room `[OR] --` $\mathbb{P}(\texttt{room}|\texttt{object})$: Generate compatibility scores for rooms for a given object.
- Object Room Receptacle `[ORR] --` $\mathbb{P}(\texttt{receptacle}|\texttt{object}, \texttt{room})$: Generate compatibility scores for receptacles within a given room and for a given object.

Both of these are learned from the human rearrangement preferences dataset. From the compatibility scores in the `ORR` task, we first determine which objects in our list of objects are misplaced and which are correctly placed. To do this, we compute a hyperparameter s_L — the score threshold — from our `val` episodes using a grid search. Receptacles whose scores are above s_L for a given object-room pair are marked as correct, while those whose scores are below s_L are marked as incorrect. We then treat this as a classification task and pick s_L that maximizes the F1 score on the `val` episodes.

Next, to determine the ranking of receptacles for a given misplaced object, we use the probabilities from both the `OR` and `ORR` tasks. For a given object, we first rank the rooms in descending order of $\mathbb{P}(\text{room}|\text{object})$. Then, for each object-room pair in the ranked room list, we rank the *correct* receptacles in the room in descending order of $\mathbb{P}(\text{receptacle}|\text{object}, \text{room})$. Finally, we place *incorrect* receptacles at the end of our list.

To learn the probability scores in the OR and ORR tasks, we start by extracting word embeddings from a pretrained RoBERTa LLM [68] of all objects, receptacles. We experiment with various contextual prompts [69, 70] for extracting embeddings of paired room-receptacle (*e.g.* "<receptacle> of <room>") and object-room (*e.g.* "<object> in <room>") combinations. Next, we implemented the following 2 methods of using these embeddings to get the final compatibility scores:

Finetuning by Contrastive Matching (CM). We train a 3-layered MLP on top of language embeddings and compute pairwise cosine similarity between any two embeddings. Embeddings are trained using objects from seen split. We train separate models for ORR and OR. For ORR, we match an object-room pair to the receptacle with the best average rank across annotators. We use contrastive loss [71] to promote similarity between an object-room pair and the matching receptacle. For OR, we match an object with all rooms that have at least one correct receptacle for it. In this case, we use the binary cross entropy (BCE) loss to handle multiple rooms per object.

Zero-Shot Ranking via MLM (ZS-MLM). Masked Language Modeling (MLM) is used extensively for pretraining LLMs [68, 72], which involves predicting a masked word (*i.e.* [mask]) given the surrounding context words. This objective can be extended for zero-shot ranking using various contextual prompts. For ORR, we use the prompt "in <room>, usually you put <object> <spatial-preposition> [mask]" to rank receptacles given an object, a room, and a spatial preposition (*e.g.* in or on). For OR, we use the prompt "in a household, it is likely that you can find <object> in the room called [mask]". We compare these ranking approaches with other baselines in Section 5.4.1.

Table 5.1: We report mAP scores on train, and unseen objects splits of val and test for both OR and ORR matching tasks.

	ORR			OR		
# **Method**	train	val-u	test-u	train	val-u	test-u
1 RoBERTa+CM	0.81	**0.79**	**0.81**	**1.0**	**0.65**	0.65
2 GloVe+CM	**0.88**	0.76	0.76	**1.0**	**0.65**	**0.66**
3 ZS-MLM	0.43	0.46	0.42	0.51	0.54	0.52
4 Random	0.47	0.47	0.46	0.58	0.52	0.59

5.3.2 High-level planner:

At each step, the planner invokes the LLM-based ranking module function (from Section 5.3.1) to determine potential rearrangements. It decides to explore (via frontier-based exploration) only if no rearrangements are pending. It continues to explore for n_ϵ (= 16) steps, before checking for rearrangements again.

5.4 Results

We first test whether LLMs can capture the embodied commonsense reasoning needed for planning in Housekeep. Then we deploy our modular agent equipped with this LLM-based planner to benchmark its ability to generalize to unseen environments cluttered with novel objects from seen (*i.e.* test-seen) and unseen (*i.e.* test-unseen) categories. Finally, we perform a thorough qualitative analysis of its failure modes and highlight directions for further improvements.

5.4.1 Language Models Capture Embodied Commonsense

Methods. We evaluate CM and ZS-MLM using RoBERTa [68] as our base LLM. We also compare these with GloVe-based [73] embeddings, and a baseline that randomly ranks rooms (for OR task) and receptacles (for ORR task).

Table 5.2: Results using our modular baseline on the `test-unseen` splits. `OR`: Oracle, `LM`: LLM-based ranking, `FTR`: Frontier exploration.

		Modules		Rearrange		Efficiency
	#	Rank	Explore	**ES** ↑	**OS** ↑	**PPE** ↑
t-unseen	1	OR	OR	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	2	OR	FTR	0.35 ± 0.02	0.65 ± 0.01	1.00 ± 0.00
	3	LM	OR	0.02 ± 0.00	0.32 ± 0.01	0.42 ± 0.01
	4	LM	FTR	0.01 ± 0.00	0.23 ± 0.01	0.35 ± 0.01

Evaluation. We evaluate mean average precision (mAP) across objects to compare the ranked list of rooms/receptacles obtained from our ranking module to the list of rooms/receptacles deemed `correct` by the human annotators. For a given object, a receptacle is considered `correct` when at least 6 annotators vote for it, and a room is considered `correct` if it has at least one `correct` receptacle within it. Higher AP score indicates `correct` items are likely to ranked higher than the `incorrect` items.

Results. Table 5.1 shows that `RoBERTa+CM` outperforms `ZS-MLM` by a large margin even when fintuned on a relatively small-sized training set (~40% of total data). We find good transfer of results from `val` to `test` splits by `RoBERTa+CM` method on both tasks demonstrating the better generalization capabilities of LLMs.

5.4.2 Main results for Housekeep

We utilize the best method from Section 5.4.1, `RoBERTa+CM` as scoring function within `Ranker` module to continuously rerank (thus replan) newly discovered rooms and receptacles while exploring Housekeep episodes. We use the following metrics: **Episode Success (ES)**: Strict binary (*all* or *none*) metric that is one if and only if all objects are correctly placed when episode ends. **Object Success (OS)**: Fraction of the objects placed correctly. **Pick-Place Efficiency (PPE)**: The minimum number of picks/places required to solve the episode divided by the number of picks/places made by agent in the episode.

We show oracle agent's performance, by swapping `Ranker` and `Explore` modules with their oracle (perfect) counterparts. Oracle ranker uses the ground truth human preferences to rank the objects and receptacles found, while Oracle explore knows complete map of the environment. Compared to oracle ranker (Row 1) language model (Row 3) impacts object success (`OS`) by -68%, and episode success (`ES`) by -98%. The dramatic drop in `ES` is expected as Housekeep is a multi-step problem with compounding errors. Nonetheless, the huge drop in (`OS`) observed when transitioning from the oracle ranking module to the LM-based ranking module highlights a substantial opportunity for improvement in reasoning about the correct locations of objects.

CHAPTER 6

DISCUSSION

6.1 Dimensions of realism

Table 6.1: Comparison of the tasks introduced in the thesis along the four dimensions of realism: Autonomy, real robot evaluations, handling unseen object categories, lifelong evaluations. ✓Partially satisfies ✖Doesn't satisfy ✔Completely satisfies

	Autonomy	Real robot	Unseen categories	Lifelong
OVMM	✓	✔	✔	✓
GOAT	✓	✔	✖	✔
Housekeep	✔	✖	✔	✔

In Table 6.1, we conduct a comparative analysis of the three tasks based on the realism dimensions introduced in Section 1.1.

6.1.1 Autonomy

All tasks eliminate the need for precise geometric locations as goal specifications. OVMM uses object names and receptacle categories as goal specifications, while GOAT extends this by allowing multi-modal goals through images, language, or object category specifications. However, these tasks aren't fully autonomous as they still depend on specification of pick and place locations in *in some form* from the user. In contrast, Housekeep evaluates higher autonomy, focusing on agents' commonsense reasoning to tidy up households without any explicit instructions.

6.1.2 Real Robots

OVMM evaluates on photorealistic scenes in simulation closely mimicking real-world settings and also performs evaluations on real robot. GOAT directly conducts evaluations on the real robot. However, Housekeep's benchmark assumes access to privileged information from simulators, limiting its direct applicability to real robots. Future work should aim at developing more realistic settings for evaluating tasks that require commonsense reasoning.

6.1.3 Unseen Categories

OVMM and Housekeep are evaluated using unseen instances of seen and unseen object categories. OVMM's challenge lies in detecting open-vocabulary of objects, while Housekeep requires reasoning about correct locations of objects. However, GOAT uses a closed set of categories and achieves an impressive 83% success in the real world. Future work should evaluate GOAT's performance at reaching an open-vocabulary of objects.

6.1.4 Lifelong Evaluations

OVMM evaluates the lifelong aspect to a limited extent through the agent's exploration in the FindObj phase having the potential to benefit the subsequent FindRec phase. GOAT explicitly assesses lifelong learning by requiring agents to reach a series of goals. Housekeep, requires rearranging 3-5 objects per episode, providing opportunities for the agent to improve at finding matching receptacles for misplaced objects as the episode progresses.

While there is room for improving performance on these benchmarks, especially in Housekeep and OVMM, future benchmarks could draw inspiration from Housekeep to create more realistic tasks requiring common-sense reasoning. Additionally, evaluating GOAT with an open vocabulary of objects could further enhance benchmarking comprehensiveness. Finally, future OVMM tasks could involve multiple pick-place interactions to better evaluate the lifelong aspect.

CHAPTER 7
CONCLUSION

This thesis delves into the potential of robotic home assistants to enhance human life, using mobile manipulation tasks as crucial benchmarks for evaluating capabilities of these assistants. These tasks focus on realistic deployment conditions, including autonomy, exposure to genuine environments, dealing with unseen objects, and extended deployment durations. While Open Vocabulary Mobile Manipulation involves moving objects based solely on their names, "GO To Any Thing" requires navigating to sequence of multimodal goals in unfamiliar environments, and Housekeep focuses on tidying up without specific instructions. Proposed baselines, incorporating heuristic and learned components, leverage large-scale pretrained models for open-vocabulary object detection and reasoning. The impressive performance on the GOAT task — an 83% success rate, underscores the research's contribution to the development of more effective robotic assistants, while also identifying areas for improvement in other tasks.

Appendices

Publications

- **HomeRobot: Open Vocabulary Mobile Manipulation**

  Sriram Yenamandra*[1], Arun Ramachandran*, Karmesh Yadav*, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John Turner, Zsolt Kira, Manolis Savva, Angel Chang, Devendra Singh Chaplot, Dhruv Batra, Roozbeh Mottaghi, Yonatan Bisk, Chris Paxton

  2023 Conference on Robot Learning (CoRL)

- **GOAT: Go To Any Thing**

  Matthew Chang*, Theophile Gervet*, Mukul Khanna*, Sriram Yenamandra*, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, Roozbeh Mottaghi, Jitendra Malik*, Devendra Singh Chaplot*

  2023 Preprint (Under submission)

- **Housekeep: Tidying Virtual Households using Commonsense Reasoning**

  Yash Kant, Arun Ramachandran, Sriram Yenamandra, Igor Gilitschenski, Dhruv Batra, Andrew Szot†[2], Harsh Agrawal†

  2022 European Conference on Computer Vision (ECCV)

[1] $*$ denotes equal technical contribution

[2] † indicates equal advising

REFERENCES

[1] D. Batra *et al.*, "Rearrangement: A challenge for embodied ai," *arXiv preprint arXiv:2011.01975*, 2020.

[2] A. Szot *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," in *NeurIPS*, 2021.

[3] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi, "Visual room rearrangement," in *CVPR*, 2021.

[4] C. Gan *et al.*, "Threedworld: A platform for interactive multi-modal physical simulation," *NeurIPS Datasets and Benchmarks Track*, 2021.

[5] C. Li *et al.*, "Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation," in *CoRL*, 2023.

[6] A. Padmakumar *et al.*, "TEACh: Task-driven embodied agents that chat," in *AAAI*, 2022.

[7] M. Khanna* *et al.*, "Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation," *arXiv preprint*, 2023. arXiv: 2306.11290 `[cs.CV]`.

[8] M. Savva *et al.*, "Habitat: A Platform for Embodied AI Research," *ICCV*, 2019.

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

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

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[12] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[13] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," *CVPR*, 2020.

[14] Habitat, *Habitat challenge*, https://aihabitat.org/challenge/2021/, 2021.

[15] D. Batra *et al.*, “Objectnav revisited: On evaluation of embodied agents navigating to objects,” *arXiv*, 2020.

[16] S. Wani, S. Patel, U. Jain, A. Chang, and M. Savva, “Multion: Benchmarking semantic map memory using multi-object navigation,” *NeurIPS*, 2020.

[17] M. Narasimhan *et al.*, “Seeing the un-scene: Learning amodal semantic maps for room navigation,” *CoRR*, vol. abs/2007.09841, 2020.

[18] P. Anderson *et al.*, “Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments,” in *CVPR*, 2018.

[19] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, “Vision-and-dialog navigation,” *CoRL*, 2019.

[20] M. Shridhar *et al.*, “Alfred: A benchmark for interpreting grounded instructions for everyday tasks,” in *CVPR*, 2020.

[21] S. Srivastava *et al.*, “Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments,” in *CoRL*, 2021.

[22] M. Deitke *et al.*, “RoboTHOR: An Open Simulation-to-Real Embodied AI Platform,” in *CVPR*, 2020.

[23] S. Dasari *et al.*, “Rb2: Robotic manipulation benchmarking with a twist,” *arXiv*, 2022.

[24] G. Zhou *et al.*, “Train offline, test online: A real robot learning benchmark,” *arXiv*, 2022.

[25] K. Kimble *et al.*, “Benchmarking protocols for evaluating small parts robotic assembly systems,” *IEEE Robotics and Automation Letters*, 2020.

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

[27] A. Kadian *et al.*, “Are we making real progress in simulated environments? measuring the sim2real gap in embodied visual navigation,” *arXiv*, 2019.

[28] H. J. Levesque, E. Davis, and L. Morgenstern, “The winograd schema challenge,” in *KR*, 2011.

[29] M. Sap, H. Rashkin, D. Chen, R. Le Bras, and Y. Choi, “Socialiqa: Commonsense reasoning about social interactions,” in *EMNLP*, Apr. 2019.

[30] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi, "COMET: Commonsense Transformers for Automatic Knowledge Graph Construction," in *ACL*, 2019.

[31] M. Sap *et al.*, "Atomic: An atlas of machine commonsense for if-then reasoning," *ArXiv*, vol. abs/1811.00146, 2019.

[32] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, "From recognition to cognition: Visual commonsense reasoning," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6713–6724, 2019.

[33] K. Sakaguchi, R. Le Bras, C. Bhagavatula, and Y. Choi, "Winogrande: An adversarial winograd schema challenge at scale," in *AAAI*, 2020.

[34] C. Bhagavatula *et al.*, "Abductive commonsense reasoning," *ArXiv*, vol. abs/1908.05739, 2020.

[35] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "Hellaswag: Can a machine really finish your sentence?" In *ACL*, 2019.

[36] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, "Swag: A large-scale adversarial dataset for grounded commonsense inference," in *EMNLP*, 2018.

[37] B. Zhou, D. Khashabi, Q. Ning, and D. Roth, ""going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding," *ArXiv*, vol. abs/1909.03065, 2019.

[38] H. Agrawal, A. Chandrasekaran, D. Batra, D. Parikh, and M. Bansal, "Sort story: Sorting jumbled images and captions into stories," in *EMNLP*, 2016.

[39] N. Mostafazadeh *et al.*, "A corpus and cloze evaluation for deeper understanding of commonsense stories," in *NAACL*, 2016.

[40] M. Granroth-Wilding and S. Clark, "What happens next? event prediction using a compositional neural network model," in *AAAI*, 2016.

[41] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi, *Piqa: Reasoning about physical commonsense in natural language*, 2019. arXiv: 1911.11641 `[cs.CL]`.

[42] 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.

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

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

[45] E. Kolve *et al.*, “AI2-THOR: an interactive 3d environment for visual AI,” *arXiv*, 2017.

[46] J. Collins *et al.*, “Abo: Dataset and benchmarks for real-world 3d object understanding,” in *CVPR*, 2022.

[47] L. Downs *et al.*, “Google scanned objects: A high-quality dataset of 3d scanned household items,” in *ICRA*, 2022.

[48] J. Wu *et al.*, “Tidybot: Personalized robot assistance with large language models,” *arXiv*, 2023.

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

[50] J. Krantz *et al.*, “Navigating to objects specified by images,” *arXiv*, 2023.

[51] P. Anderson *et al.*, “On evaluation of embodied navigation agents,” *arXiv preprint arXiv:1807.06757*, 2018.

[52] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, “Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 171–23 181.

[53] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, “Navigating to objects in the real world,” *Science Robotics*, vol. 8, no. 79, eadf6991, 2023.

[54] D. Driess *et al.*, “Palm-e: An embodied multimodal language model,” *arXiv*, 2023.

[55] b. ichter brian *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” in *Proceedings of The 6th Conference on Robot Learning*, K. Liu, D. Kulic, and J. Ichnowski, Eds., ser. Proceedings of Machine Learning Research, vol. 205, PMLR, Dec. 2023, pp. 287–318.

[56] J. Gu, D. S. Chaplot, H. Su, and J. Malik, “Multi-skill mobile manipulation for object rearrangement,” *arXiv preprint arXiv:2209.02778*, 2022.

[57] B. Shen *et al.*, “Igibson, a simulation environment for interactive tasks in large realistic scenes,” *arXiv preprint arXiv:2012.02924*, 2020.

[58] G. Research, *Google Scanned Objects*, https : / / app . ignitionrobotics . org / GoogleResearch / fuel / collections / Google % 20Scanned % 20Objects, [Online; accessed Feb-2022], 2020.

[59] 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.

[60] F. robotics, *Fetch*, http://fetchrobotics.com/, 2020.

[61] K. Crowston, "Amazon mechanical turk: A research tool for organizations and information systems scholars," in *Shaping the future of ict research. methods and approaches*, 2012.

[62] M. J. Salganik, *Bit by Bit: Social Research in the Digital Age*, Open Review Edition. 2017.

[63] A. Majumdar, A. Shrivastava, S. Lee, P. Anderson, D. Parikh, and D. Batra, "Improving vision-and-language navigation with image-text pairs from the web," *ArXiv*, vol. abs/2004.14973, 2020.

[64] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. Gould, "A recurrent vision-and-language BERT for navigation," in *ECCV*, 2021.

[65] F. Hill, S. Mokra, N. Wong, and T. Harley, "Human instruction-following with deep reinforcement learning via transfer-learning from text," *ArXiv*, vol. abs/2005.09382, 2020.

[66] S. Li *et al.*, "Pre-trained language models for interactive decision-making," *ArXiv*, vol. abs/2202.01771, 2022.

[67] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," *ArXiv*, vol. abs/2201.07207, 2022.

[68] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv:1907.11692*, 2019.

[69] F. Petroni *et al.*, "Language models as knowledge bases?" In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[70] F. Petroni *et al.*, "How context affects language models' factual predictions," in *Automated Knowledge Base Construction*, 2020.

[71] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.

[72] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[73] J. Pennington, R. Socher, and C. Manning, “GloVe: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.