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†
*Indicates equal contribution, †Work done at Fair, Meta

***Abstract*—In deployment scenarios such as homes and warehouses, mobile robots are expected to autonomously navigate for extended periods, seamlessly executing tasks articulated in terms that are intuitively understandable by human operators. We present GO To Any Thing (GOAT), a universal navigation system capable of tackling these requirements with three key features: a) Multimodal: it can tackle goals specified via category labels, target images, and language descriptions, b) Lifelong: it benefits from its past experience in the same environment, and c) Platform Agnostic: it can be quickly deployed on robots with different embodiments. GOAT is made possible through a modular system design and a continually augmented instance-aware semantic memory that keeps track of the appearance of objects from different viewpoints in addition to category-level semantics. This enables GOAT to distinguish between different instances of the same category to enable navigation to targets specified by images and language descriptions.**

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 and ablations by 32% (absolute improvement). GOAT improves with experience in the environment, from a 60% success rate at the first goal to a 90% success after exploration. In addition, we demonstrate that GOAT can readily be applied to downstream tasks such as pick and place and social navigation.

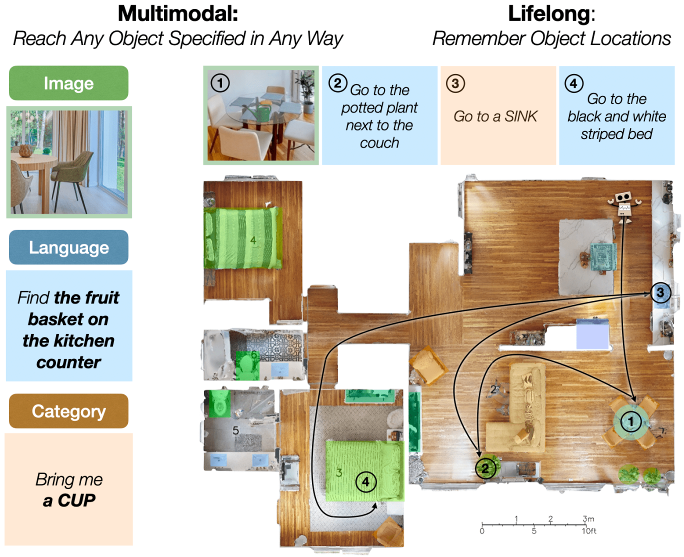
Multimodal:
Reach Any Object Specified in Any Way
Lifelong:
Remember Object Locations
Image
Language
Find the fruit basket on the kitchen counter
Category
Bring me a CUP
1
2 Go to the potted plant next to the couch
3 Go to a SINK
4 Go to the black and white striped bed

Fig. 1. **GOAT (GO to Any Thing) task.** The GOAT task requires lifelong learning, meaning taking advantage of past experience in the same environment, for multimodal navigation. The robot must be able to reach any object specified in any way and remember object locations to come back to them.

I. Introduction

Consider a robot starting in an unseen environment as shown in Figure 1, and suppose it is asked to find a dining table image (goal 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). Suppose the robot is then asked to *Go to the potted plant next to the couch* (goal 2). This requires visual grounding of the text instruction in the physical space. The next instruction is to *Go to a SINK* (goal 3), the capitalization emphasizing that any object of the category SINK is a valid goal. In this example, 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. In addition to these capabilities of multimodal perception, exploration, lifelong memory, and goal localization, the robot also needs effective planning and control to reach the goal while avoiding obstacles.

In this paper, we present *GO to Any Thing* (GOAT), a universal navigation system with three key features:
a) **Multimodal:** it can tackle goals specified via category labels, target images, and language descriptions, b) **Lifelong:** it benefits from its past experience in the same environment in the form of a map of objects instances (as opposed to stored implicitly within the parameters of a machine learning model) updated over time, and c) **Platform Agnostic:** it can be seamlessly deployed on robots with different embodiments — we deploy GOAT on a quadruped and a wheeled robot. GOAT is made possible through the design of an *Object Instance Memory* that keeps track of the appearance of objects from different viewpoints in addition to category-level semantics. This enables GOAT to distinguish between different instances of the same category to enable navigation to targets specified by images and fine-grained language descriptions. This memory is continually augmented as the agent spends more time in the environment, leading to improved efficiency in reaching goals over time.

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 and ablations by 32% (absolute improvement). GOAT performance improves with experience in the environment from a 60% success rate at the first goal to 90% success rate once the environment is fully explored. In addition, we demonstrate that GOAT, as a general navigation primitive, can readily be applied to downstream tasks like pick and place and social navigation. GOAT's performance can in part be attributed to the modular nature of the system: it leverages learning in the components in which it is required (*i.e.* object detection, image/language matching) while still leveraging strong classical methods (*i.e.* mapping and planning). Modularity is also responsible for the ease of deployment across different robot embodiments and downstream applications, as individual components can be easily adapted or new components introduced.

II. Related Work

While there is a large body of work on navigation [58], most only evaluate in simulation or develop specialized solutions to tackle a subset of these tasks. Classical robotics works [57] employed geometric reasoning to solve navigation to geometric goals. With advances in semantic understanding of images, researchers started using semantic reasoning to improve exploration efficiency in novel environments [8] and tackling semantic goals specified via categories [47, 21, 3, 7, 32, 53, 6], images [63, 9, 22, 33, 34] and language instructions [42, 56, 18, 40, 11, 26]. Most of these methods are a) specialized to a single task (*i.e.* are designed only for language goals or only for image goals), b) only tackle a single goal in each episode (*i.e.* are not lifelong), and c) evaluated only in simulation (or rudimentary real-world environments). GOAT advances upon these works on all three fronts and tackles multiple goal specifications in a lifelong manner in the real world. This supersedes past works that only innovate along one axis, *e.g.* past works [59, 9] tackle a sequence of goal but goals are limited to either be object goals [59] or image goals [9] in simulation, or rely on a pre-exploration phase [11]. [1] tackles flexible goal specifications but only shows simulated results for one goal per episode. This work builds on top of the system from [19], leveraging the same modular structure; using SLAM and pre-trained object detectors to maintain a semantic map for planning. While [19] can only reach one categorical goal, in this paper, we introduce a lifelong memory that enables this system to reach a sequence of goals including language and image goals.

GOAT maintains a map of the environment as well as visual landmarks - egocentric views of object instances - which are stored in our novel instance-aware object memory. This memory should be queryable with both images and natural language to satisfy GOAT's multimodality requirement. We enable this by storing raw images for visual landmarks, as opposed to features, allowing us to leverage recent advances in image-image matching and image-language matching independently. We use Contrastive Language-Image Pretraining (CLIP) [46] for image-language matching and SuperGlue [50] for image-image matching. CLIP follows a long history of associating text with images or regions in images [25, 16, 17, 15, 31, 35, 44] and has led to the development of language-conditioned open-vocabulary object detectors [62, 37, 43]. CLIP itself, or object detectors derived from CLIP have recently been used for robotic tasks, *e.g.* object search [18], mobile manipulation [61], and table-top manipulation [54]. Similarly, SuperGlue follows a long history of geometric image matching [27, 38] with recent learning-based methods [50] leading to better performance in certain situations. Recent work has started evaluating these in embodied settings where a robot must navigate either to an image in the world [33, 9] or to an image corresponding to a particular object instance [34].

GOAT's memory representation builds upon a rich set of scene representation in robotics over the last 40 years: occupancy maps (with geometry [14], explicit semantics [49, 10], or implicit semantics [21]), topological representations [9, 12, 36, 51], and neural feature fields [55, 52, 41, 5]. Many of these works have started using pre-trained vision-language features like CLIP [46] and either projecting them into 3D directly [29] or capturing them in an implicit neural field [52, 5]. Parametric representations summarize the environment into low-dimensional abstract features, while non-parametric representations view the collection of images itself as a representation. Our work leverages aspects of both. We build a semantic map for navigating to objects but also store raw images associated with discovered objects (landmarks). Dense representations storing CLIP features at every location [26] don't yet scale to entire homes without server-grade GPUs, whereas our sparse landmark representation does.

III. GOAT 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 $g_k,\ k \in \{1, 2, .., N\}$, 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}.

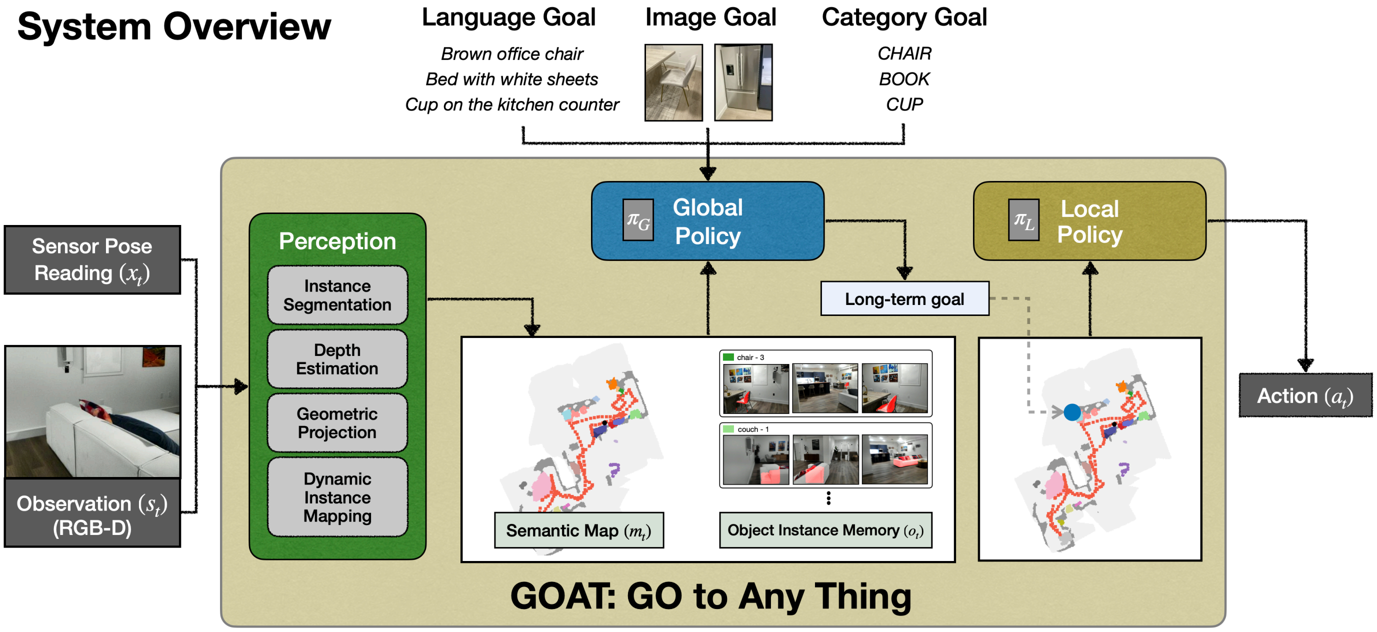

Fig. 2. **GOAT system overview.** The perception system detects and localizes object instances, the global policy outputs high-level navigation commands depending on whether the robot should explore or reach a goal already in memory, and the local policy executes these commands.

IV. GOAT Method

a) GOAT Agent: Figure 2 shows an overview of the GOAT system. 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. When searching for image or language goals, it is insufficient to simply find objects of the correct class (*i.e.* if the goal is an image of a specific black chair, finding a brown chair is incorrect). Previous works that only maintain a semantic map ([19]) are unable to solve this problem.

To this end, in addition to the semantic map, GOAT maintains an *Object Instance Memory* that localizes individual instances of objects 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.

b) Perception: Figure 3 shows the perception system. It takes as input the current depth image D_t, RGB image I_t, and pose reading x_t from onboard sensors. It uses an off-the-shelf model to segment instances in the RGB image. We use MaskRCNN [24] with a ResNet50 [23] backbone pretrained on MS-COCO for object detection and instance segmentation. We chose MaskRCNN as current open-set models, such as Detic [62], were less reliable for common categories in early experiments. We also estimate depth to fill in holes for reflective objects in raw sensor readings using MiDaS [48] monocular depth estimation (additional details in supplementary). We project the first-person semantic segmentation into a point cloud, bin the point cloud into a 3D semantic voxel map, and finally sum over the height to compute a 2D instance map m_t. For each detected object instance, we also store the image in which the object was detected as part of the object instance memory.

c) Semantic Map Representation: The semantic map (m_t at timestep t) is a spatial representation of the environment that keeps track of object locations, obstacles, and explored areas. Concretely, it is a $K \times M \times M$ matrix of integers where $M \times M$ is the map size, and K is the number of map channels. Each cell of this spatial map corresponds to 25cm^2 (5cm × 5cm) in the physical world. Map channels $K = C + 4$ where C is the number of semantic object categories ($C = 15$ in our experiments), and the remaining 4 channels represent the obstacles, the explored area, and the agent's current and past locations. An entry in the map is non-zero if the cell contains an object of a particular semantic category, an obstacle, or is explored, depending on the channel, and zero otherwise. In this semantic map representation, the first C channels store the unique instance ids of the projected objects. The map is initialized with all zeros at the beginning of an episode, and the agent starts at the center of the map facing east. While this map representation has been used for previous methods for object goal navigation, this is insufficient for reaching specific objects specified by language or image goals (*i.e.* find the specific book depicted in an image instead of just finding any book). To solve this problem we must track and store

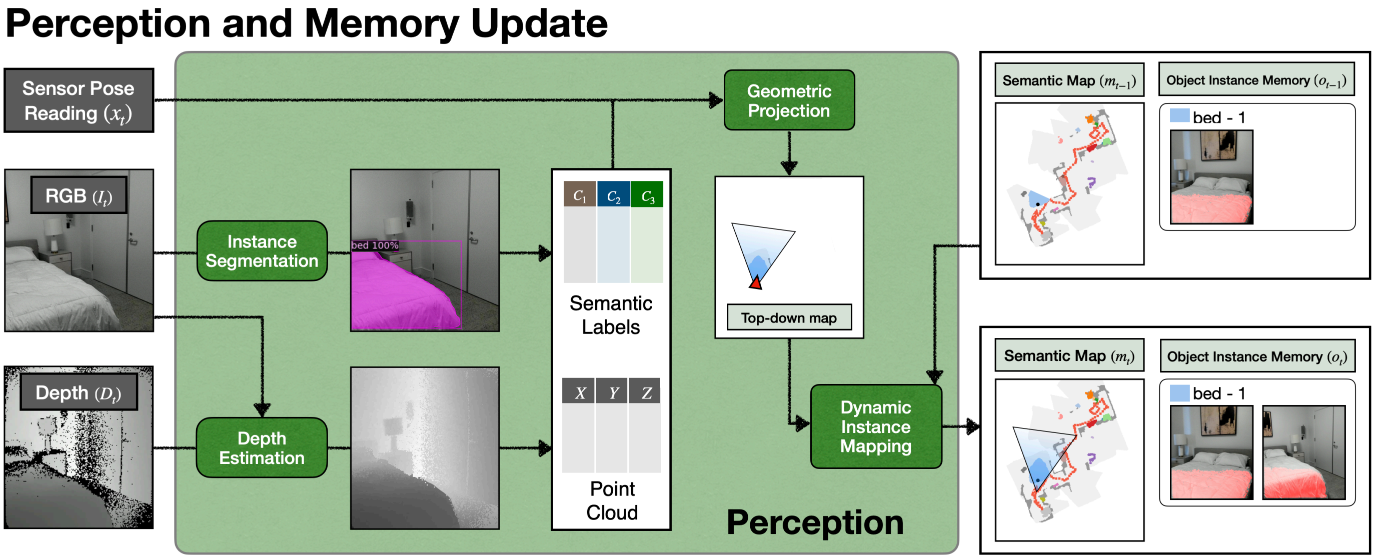

Fig. 3. **Perception and memory update.** The perception system processes RGB-D input to infill depth, segment object instances, project them into a top-down semantic map, and store views in the Object Instance Memory.

information about individual object instances. To achieve this we maintain an Object Instance Memory (Figure 4 A).

d) Object Instance Memory: Our Object Instance Memory tracks individual instances of objects and stores every image (along with the detection bounding box) in which that object has been detected. Every time an object o_i is detected in the incoming RGB observation I, the depth image is used to project the location of the detected object into the map. If the projection of o_i spatially overlaps an existing object o_j, they are considered to be the same object, and I is added to the list of images associated with o_j. If o_i does not overlap an existing object, it is considered a new instance.

e) Dynamic Memory: In real-world settings, the environment may be dynamic, with humans creating or removing obstacles, and relocating objects. To handle these cases we allow our scene representation to be dynamic as well. We take a simple approach: when new observations are received from the sensors, we overwrite the relevant cells in the semantic map based on the updated occupancy information. If a region was previously observed to contain an object and a new observation shows that it is vacant, it is marked as vacant. If a region was previously unoccupied and is observed to be traversable, it is marked as traversable. With this approach, the agent will still be able to navigate through regions that were previously marked as occluded and remove objects from the map that have been moved.

f) Global Policy: shows the global policy. When only searching for category goals (the ObjectNav task) goal localization is trivial. Simply find the closest cell in the semantic map that matches the desired class. However, when seeking image or language goals, finding the correct object instance is not so simple. In the GOAT system, the global policy (Figure 4 (B)) selects the correct object instance for each goal.

When a new goal is specified to the agent, the global policy π_G first searches the Object Instance Memory to see if the goal has already been observed. The method for matching goals to object instances is tailored to the modality of the goal specification. For category goals, we simply find the nearest object of the goal category in the semantic map. For language goals, we first extract an object category from the language description (by prompting with Mistral 7B [30] in our experiments), then match CLIP features of the language description with CLIP [46] features of each object instance of the inferred category in our Object Instance Memory. Similarly, for image goals, we first extract an object category from the image with MaskRCNN, then match keypoints of the goal image with keypoints of each object instance of the inferred category with an off-the-shelf SuperGlue [50] model (additional details and ablations about image and language matching in supplementary).

While the environment is being explored, we consider the object instance to match a given goal if the similarity score crosses a threshold (0.28 for CLIP, 6.0 for Superglue). When the environment is fully explored, we simply select the object instance with the highest similarity score. After an instance is selected, its stored location in the top-down map is used as a long-term point navigation goal. If no instance is localized, the global policy outputs an exploration goal. We use frontier-based exploration [60], which selects the closest unexplored region as the goal.

g) Local Policy: Given a long-term goal output by the global policy π_G, the local policy π_L uses the Fast Marching Method to plan a path towards it. On the Spot robot, we use the built-in point navigation controller to reach waypoints along this path. On the Stretch robot with no such built-in controller, we plan the first low-level action along this path deterministically as in [19].

V. Results

We evaluate the ability of the GOAT agent to tackle the GOAT task, i.e., reach a sequence of unseen multimodal object instances in unseen environments.

We deployed GOAT on and conducted qualitative experiments with Boston Dynamics Spot and Hello Robot Stretch robots. We conducted large-scale quantitative experiments with GOAT on Spot (due to its higher reliability) against 3 baselines in 9 real-world homes to reach a total of 200+ different object instances (see Figure 5). A demo video qualitatively illustrating our results can be found in the supplementary.

a) Experimental Setting: We evaluate the GOAT agent as well as three baselines in nine visually diverse homes (see Figure 5) with 10 episodes per home consisting of 5-10 object instances randomly selected out of objects available in the home, representing 200+ different object instances in total (more visualizations in supplementary). 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'). These categories were chosen to cover a wide range of object sizes (from cups to couches), classes with multiple instances (there may be many chairs), and objects that may be co-located in a 2D map (book resting on a dining table). We took a picture of each object for image goals following the protocol in Krantz *et al.* [34], 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 [2], 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.

b) Baselines: We compare GOAT to three baselines:
1. CLIP on Wheels [18] - the existing work that comes closest to being able to address the GOAT problem setting - which keeps track of all images the robot has seen and, when given a new goal object, decides whether the robot has already seen it by matching CLIP [46] 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 [19], 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.

c) Quantitative Results: Table I 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 [18] attains a 51% success rate, showing that using GOAT's Object Instance Memory for goal matching is more effective than CLIP feature matching against all previously viewed images. GOAT w/o Instances achieves a 49% success rate, with 29% and 28% success rates for image and language goals, respectively. This shows the need to keep track of enough information in memory to distinguish between different object instances, which [19] couldn't 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 shows the need to keep track of a lifelong memory. Figure 6 further emphasizes this point: GOAT performance improves with experience in the environment from a 60% success rate (0.20 SPL) at the first goal to 90% (0.80 SPL) for goals 5-10 after thorough exploration. Conversely, GOAT without memory shows no improvement from experience, while COW benefits but plateaus at much lower performance. Figure 7 shows example trajectories from GOAT and baselines.

VI. Applications

As a general navigation primitive, the GOAT policy can readily be applied to downstream tasks such as pick and place and social navigation.

Open Vocabulary Mobile Manipulation: The ability to perform rearrangement tasks is essential in any deployment scenarios for mobile robots (homes, warehouses, factories) [4, 61, 13, 28, 20]. 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. Demo video and visualizations can be found in supplementary.

Social Navigation: To operate in human environments, mobile robots need the ability to treat people as dynamic obstacles, plan around them, and search for and follow people [39, 45]. To give the GOAT policy such skills, we treat people as image object instances with the *PERSON* category. For each participant, we take a front-facing full-body image to be used as the image goal for that participant. This enables GOAT to deal with multiple people, just like it can deal with multiple instances of any object category. Using the dynamic memory protocol described in Section IV GOAT will remove someone's previous location from the map after they have moved, and continue mapping their new location. This allows GOAT to track a moving person.

To evaluate GOAT's ability to treat people as dynamic obstacles, we conducted a pilot study including moving people as obstacles. In one of the novel homes used for evaluation,

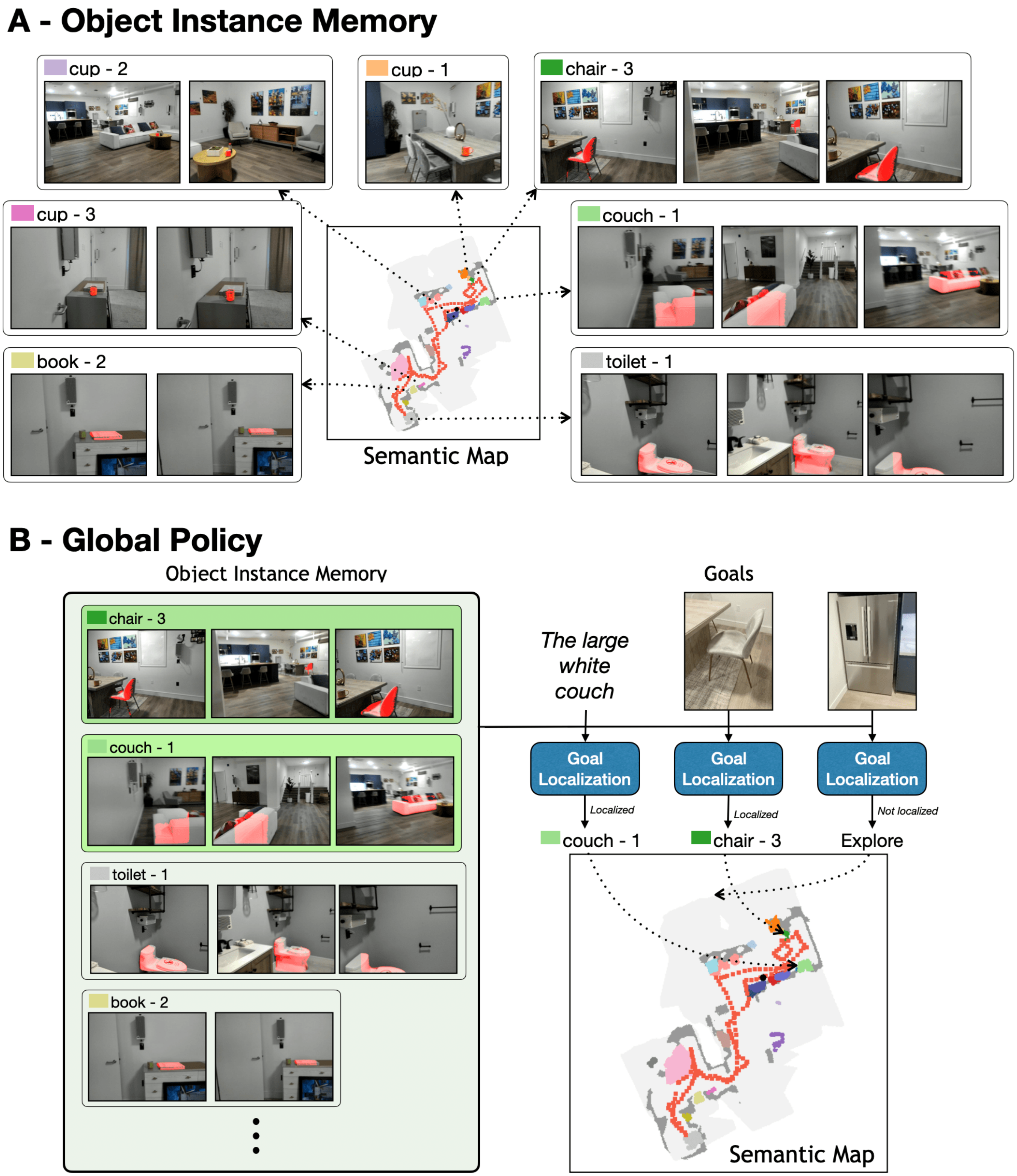

Fig. 4. **(A) Object Instance Memory.** We cluster object detections, along with image views in which they were observed, into instances using their location in the semantic map and their category. **(B) Global Policy.** When a new goal is specified, the global policy first tries to localize it within the Object Instance Memory. If no instance is localized, it outputs an exploration goal.

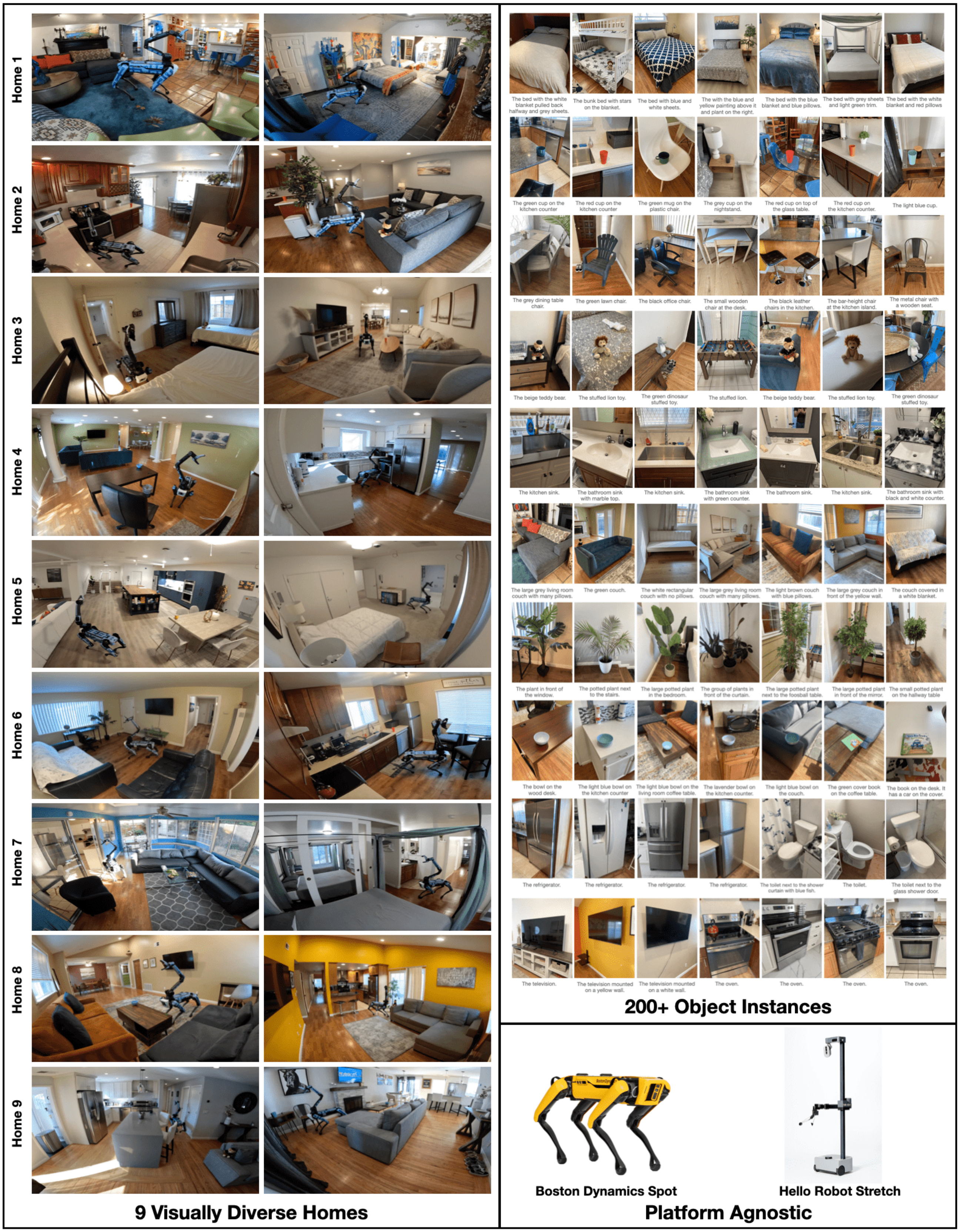
Home 1
Home 2
Home 3
Home 4
Home 5
Home 6
Home 7
Home 8
Home 9
9 Visually Diverse Homes
The bed with the white blanket pulled back halfway and grey sheets.
The bunk bed with stars on the blanket.
The bed with blue and white sheets.
The with the blue and yellow painting above it and plant on the right.
The bed with the blue blanket and blue pillows.
The bed with grey sheets and light green trim.
The bed with the white blanket and red pillows
The green cup on the kitchen counter
The red cup on the kitchen counter
The green mug on the plastic chair.
The grey cup on the nightstand.
The red cup on top of the glass table.
The red cup on the kitchen counter.
The light blue cup.
The grey dining table chair.
The green lawn chair.
The black office chair.
The small wooden chair at the desk.
The black leather chairs in the kitchen.
The bar-height chair at the kitchen island.
The metal chair with a wooden seat.
The beige teddy bear.
The stuffed lion toy.
The green dinosaur stuffed toy.
The stuffed lion.
The beige teddy bear.
The stuffed lion toy.
The green dinosaur stuffed toy.
The kitchen sink.
The bathroom sink with marble top.
The kitchen sink.
The bathroom sink with green counter.
The bathroom sink.
The kitchen sink.
The bathroom sink with black and white counter.
The large grey living room couch with many pillows.
The green couch.
The white rectangular couch with no pillows.
The large grey living room couch with many pillows.
The light brown couch with blue pillows.
The large grey couch in front of the yellow wall.
The couch covered in a white blanket.
The plant in front of the window.
The potted plant next to the stairs.
The large potted plant in the bedroom.
The group of plants in front of the curtain.
The large potted plant next to the foosball table.
The large potted plant in front of the mirror.
The small potted plant on the hallway table
The bowl on the wood desk.
The light blue bowl on the kitchen counter
The light blue bowl on the living room coffee table.
The lavender bowl on the kitchen counter.
The light blue bowl on the couch.
The green cover book on the coffee table.
The book on the desk. It has a car on the cover.
The refrigerator.
The refrigerator.
The refrigerator.
The refrigerator.
The toilet next to the shower curtain with blue fish.
The toilet.
The toilet next to the glass shower door.
The television.
The television mounted on a yellow wall.
The television mounted on a white wall.
The oven.
The oven.
The oven.
The oven.
200+ Object Instances
Boston Dynamics Spot
Hello Robot Stretch
Platform Agnostic

Fig. 5. **"In-the-wild" evaluation.** We deploy the GOAT navigation policy in 9 visually diverse homes and evaluate in on reaching 200+ different object instances as category, image, or language goals. GOAT is platform-agnostic: we deploy it on both Boston Dynamics Spot and Hello Robot Stretch.

TABLE I

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 [2], 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

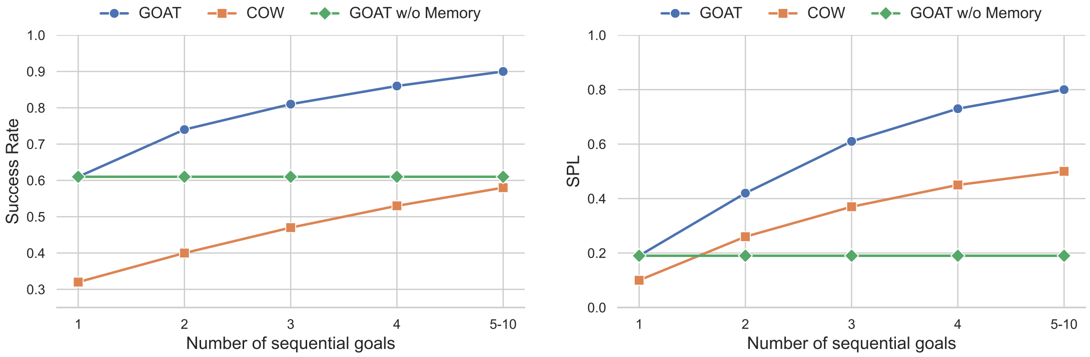

Fig. 6. **Navigation performance based on sequential goal count.** GOAT performance improves with experience in the environment: from a 60% success rate (0.2 SPL) at the first goal to 90% (0.8 SPL) for goals 5-10 after thorough exploration. Conversely, GOAT without memory shows no improvement from experience, while COW benefits but plateaus at much lower performance.

we collected an additional 5 trajectories (5-10 navigation goals each) during which people continuously moved throughout the scene. Either one or two people (chosen randomly) were instructed to walk to a randomly selected sequence of objects (waiting briefly at each object) in the scene while the robot was navigating. They were instructed to treat the robot as they would another human (*i.e.* not walking directly into the robot). In this setting, GOAT preserves an 81% success rate. We further evaluate the ability of GOAT to search for and follow people by introducing people as image goals in 5 additional trajectories following the same protocol. GOAT can localize and follow people with 83% success, close to the 86% success rate for static image instance goals. Demo video and visualizations can be found in supplementary.

VII. Discussion

a) Modularity allows GOAT to Achieve Robust General-Purpose Navigation in the Real World: The GOAT system as a whole is a robust navigation platform, achieving a success rate of 83% across image, language, and category goals in the wild and up to 90% once the environment is fully explored (Table I, Figure 6). This is possible in-part due to the modular nature of the system. A modular system allows learning to be applied in the components in which it is required (*i.e.* object detection, image/language matching), while still leveraging strong classical methods (*i.e.* mapping and planning). Furthermore, for learning-based components, we can use models trained on large datasets (*i.e.* CLIP, MaskRCNN), or specialized tasks (monocular depth estimation) to full effect, where a task-specific end-to-end learned approach would be limited by the available data for this specific task. GOAT is able to tie all of these components together using our Object Instance memory to achieve state-of-the-art performance for lifelong real-world navigation.

Furthermore, the modular design of GOAT allows it to be easily adapted to different robot embodiments and a variety of downstream applications. GOAT can be deployed on any robot with an RGB-D camera, a pose sensor (onboard SLAM), and the ability to execute low-level locomotion commands (move forward, turn left, turn right). GOAT's modularity eliminates the need for new data collection or training when deployed on a new robot platform. This stands in contrast to end-to-end methods, which would require new data collection and retraining for every different embodiment.

Consequently, new modalities of goals can easily be added to the system as long as a mechanism for matching exists and the robot is equipped with the correct sensors. For example, if goals are specified by 3D models, all that is required is a module for estimating the 3D shape of detected objects and for matching against the specified goals.

b) Matching Performance During Exploration Lags Behind Performance After Exploration: Using a predefined threshold for goal-to-object matching scores during exploration can be error-prone (visualization in supplementary). On the other hand, once the scene has been explored, the agent has the privilege of selecting the best matching instance across all observed objects. This is reflected in the improved performance of the agent post-exploration (Figure 6). When the

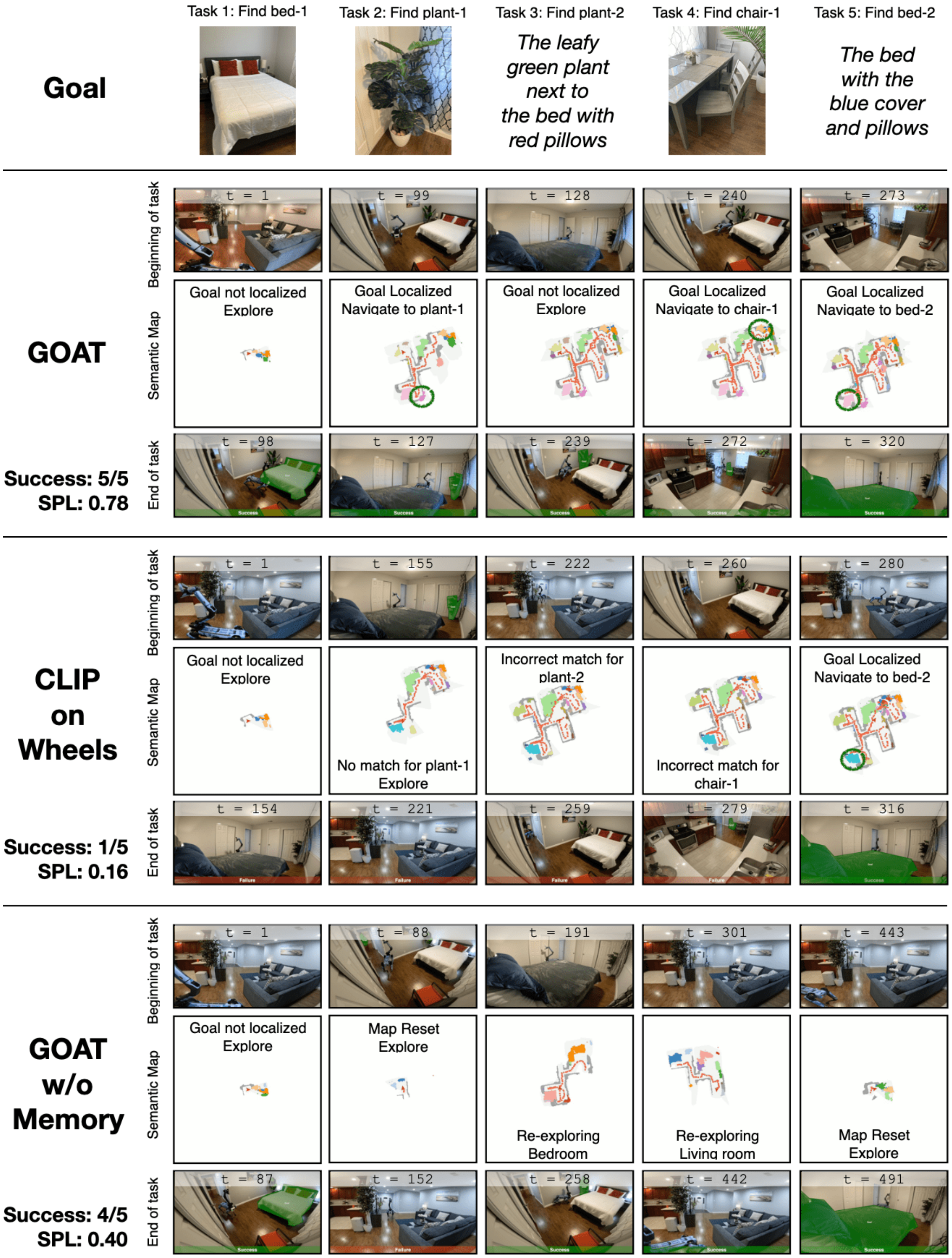

Fig. 7. **Online evaluation qualitative trajectories.** We compare methods on the same sequence of 5 goals (top) in the same environment. GOAT localizes all goals and navigates efficiently (with an SPL of 0.78). CLIP on Wheels localizes only 1 out of 5 goals, illustrating the superiority of GOAT's Object Instance Memory for matching. GOAT without memory is able to localize 4 our of 5 goals, but with an SPL of only 0.40 as it has to re-explore the environment with every goal. See Section V for details.

environment is fully explored, failures are almost exclusively due to failures in matching the correct goal. The most common failure is a language goal being matched against the an object of the correct class, but the wrong instance (i.e. The language specifies a bed, but the system matches against a different bed). Examples of these failures can be seen in supplementary figure S2, and additional details of matching matching performance can be found in supplementary.

c) Image Goal Matching is More Reliable than Language Goal Matching: We observe that image-to-image goal matching is more successful at identifying goal instances as compared to matching instance views with semantic features of language descriptions of the goal. This is expected because SuperGLUE-based image keypoint matching can leverage correspondences in geometric properties between predicted instances and goal objects. However, the semantic feature encodings from CLIP can be incapable of capturing fine-grained instance properties – that can often be crucial for goal matching. As a result, navigation with image goals is significantly more successful (Table I, SR 86.4 *vs.* 68.2).

d) Real-World Open-Vocabulary Detection: Limitations and Opportunities: An interesting and noteworthy observation is that despite the rapid advances in open (or large) vocabulary vision-and-language models (VLMs) [37, 43], we find their performance to be significantly worse than a Mask RCNN model from 2017. We attribute this observation to two possible hypotheses: (i) open-vocabulary models trade-off robustness for being more versatile, and supporting more queries, and (ii) the internet-scale weakly labeled data sources used to train modern VLMs under-represent the kind of embodied interaction data that would benefit robots occupying real-world environments with humans. The latter represents a challenging opportunity to develop such large-scale models that are simultaneously versatile and robust for embodied applications in real-world environments.

e) Generalization to New Environments: While end-to-end learning-based solutions may suffer from overfitting on a few training scenes, the modular design of GOAT is able to avoid this issue. The generalization of GOAT is only limited by the robustness of its components, many of which have been trained on large internet-scale data. In real-world experiments, in 9 visually diverse homes we found no generalization issues in any of the components of GOAT. That being said, GOAT was designed for indoor navigation and consequently was not tested in outdoor settings where low-level locomotion is far more challenging. While utilizing large-scale models such as CLIP improves generalization, GOAT also inherits the limitations and biases of these models. For example, if the majority of the objects used for training the object detector originated from North America, the system's performance may be diminished when operating in other regions.

f) Computational Constraints: While the memory utilization of GOAT is consistently increasing throughout an episode, when the proper steps have been taken to optimize performance, this was not a hindrance in our experiments. Storing compressed images as 480×640 requires only 6 MB total for all images by the end of an episode on average. Similarly, only storing CLIP features for language matching requires minimal memory (only 257 KB on average for an entire trajectory), and allows for fast vectorized comparison for language matching (7ms on average and 29ms at max on a single GPU). The computational costs for image-to-image comparisons remain under control too as we continue to only match to the instances belonging to the category of interest. Matching a single image pair takes 45ms on a single GPU, and the matching takes 0.9s on average (and 2.6s at max) after the environment is fully explored — these matching times were more than fast enough for our experiments. However, for extremely long trajectories a mechanism to increase parallelism or cull duplicate images would be necessary to increase matching speeds.

g) Additional Limitations: To achieve robust image-matching results GOAT's memory system stores all images in which objects have been detected. For very small or compute-constrained robots this may be too memory inefficient and images should be sub-sampled. Like all systems that rely on 2D mapping, GOAT is designed to handle only a single story in a building. While this is remedied by detecting when a floor change has happened and maintaining a separate map per floor, we leave this to future work.

Acknowledgments

Saurabh Gupta's effort was supported by the NSF CAREER Award (IIS2143873).

References

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

[2] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

[4] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020.

[5] Benjamin Bolte, Austin Wang, Jimmy Yang, Mustafa Mukadam, Mrinal Kalakrishnan, and Chris Paxton. Usa-net: Unified semantic and affordance representations for robot memory. *arXiv preprint arXiv:2304.12164*, 2023.

[6] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. In *Advances in Neural Information Processing Systems*, 2020.

[7] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *In Neural Information Processing Systems (NeurIPS)*, 2020.

[8] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=HklXn1BKDH.

[9] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] Devendra Singh Chaplot, Murtaza Dalal, Saurabh Gupta, Jitendra Malik, and Russ R Salakhutdinov. Seal: Self-supervised embodied active learning using exploration and 3d consistency. *Advances in neural information processing systems*, 34:13086–13098, 2021.

[11] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11509–11522. IEEE, 2023.

[12] Howie Choset and Keiji Nagatani. Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization. *IEEE Transactions on robotics and automation*, 17(2):125–137, 2001.

[13] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[14] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[15] Hao Fang*, Saurabh Gupta*, Forrest Iandola*, Rupesh K Srivastava*, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, C Lawrence Zitnick, and Geoffrey Zweig. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015.

[16] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 15–29. Springer, 2010.

[17] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

[18] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran 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*, pages 23171–23181, 2023.

[19] Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. Navigating to objects in the real world. *Science Robotics*, 8(79): eadf6991, 2023.

[20] Jiayuan Gu, Devendra Singh Chaplot, Hao Su, and Jitendra Malik. Multi-skill mobile manipulation for object rearrangement. *arXiv preprint arXiv:2209.02778*, 2022.

[21] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[22] Meera Hahn, Devendra Singh Chaplot, Shubham Tulsiani, Mustafa Mukadam, James M Rehg, and Abhinav Gupta. No rl, no simulation: Learning to navigate without navigating. *Advances in Neural Information Processing Systems*, 34:26661–26673, 2021.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[25] Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2021.

[26] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023.

[27] Daniel P Huttenlocher and Shimon Ullman. Recognizing solid objects by alignment with an image. *International journal of computer vision*, 5(2):195–212, 1990.

[28] brian ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as i can, not as i say: Grounding language in robotic affordances. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 287–318. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/ichter23a.html.

[29] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.

[30] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[31] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016.

[32] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 104–120. Springer, 2020.

[33] Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-specific image goal navigation: Training embodied agents to find object instances. *arXiv preprint arXiv:2211.15876*, 2022.

[34] Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to objects specified by images. In *ICCV*, 2023.

[35] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[36] Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and autonomous systems*, 8(1-2):47–63, 1991.

[37] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RriDjddCLN.

[38] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[39] Matthias Luber, Luciano Spinello, Jens Silva, and Kai O Arras. Socially-aware robot navigation: A learning approach. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 902–907. IEEE, 2012.

[40] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35: 32340–32352, 2022.

[41] Pierre Marza, Laetitia Matignon, Olivier Simonin, Dhruv Batra, Christian Wolf, and Devendra Singh Chaplot. Autonerf: Training implicit scene representations with autonomous agents. *arXiv preprint arXiv:2304.11241*, 2023.

[42] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*, 2021.

[43] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022.

[44] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazeb-

nik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

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

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[47] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18890–18900, 2022.

[48] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.

[49] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013.

[50] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[51] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018.

[52] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022.

[53] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.

[54] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[55] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[56] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009, 2023.

[57] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B Cremers, Frank Dellaert, Dieter Fox, Dirk Hahnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, et al. Minerva: A second-generation museum tour-guide robot. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 3. IEEE, 1999.

[58] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

[59] Saim Wani, Shivansh Patel, Unnat Jain, Angel Chang, and Manolis Savva. Multion: Benchmarking semantic map memory using multi-object navigation. *Advances in Neural Information Processing Systems*, 33:9700–9712, 2020.

[60] Brian Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on Autonomous agents*, pages 47–53, 1998.

[61] Sriram Yenamandra, 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, and Chris Paxton. Homerobot: Open vocabulary mobile manipulation. 2023. URL https://github.com/facebookresearch/home-robot.

[62] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022.

[63] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017.