GOAT-Bench: A Benchmark for Multi-Modal Lifelong Navigation

Mukul Khanna[1]* Ram Ramrakhya[1]* Gunjan Chhablani[1] Sriram Yenamandra[1] Theophile Gervet[2]
Matthew Chang[3] Zsolt Kira[1] Devendra Singh Chaplot[4] Dhruv Batra[1] Roozbeh Mottaghi[5]
[1]Georgia Institute of Technology [2]Carnegie Mellon University
[3]University of Illinois Urbana-Champaign [4]Mistral AI [5]University of Washington
mukulkhanna.github.io/goat-bench

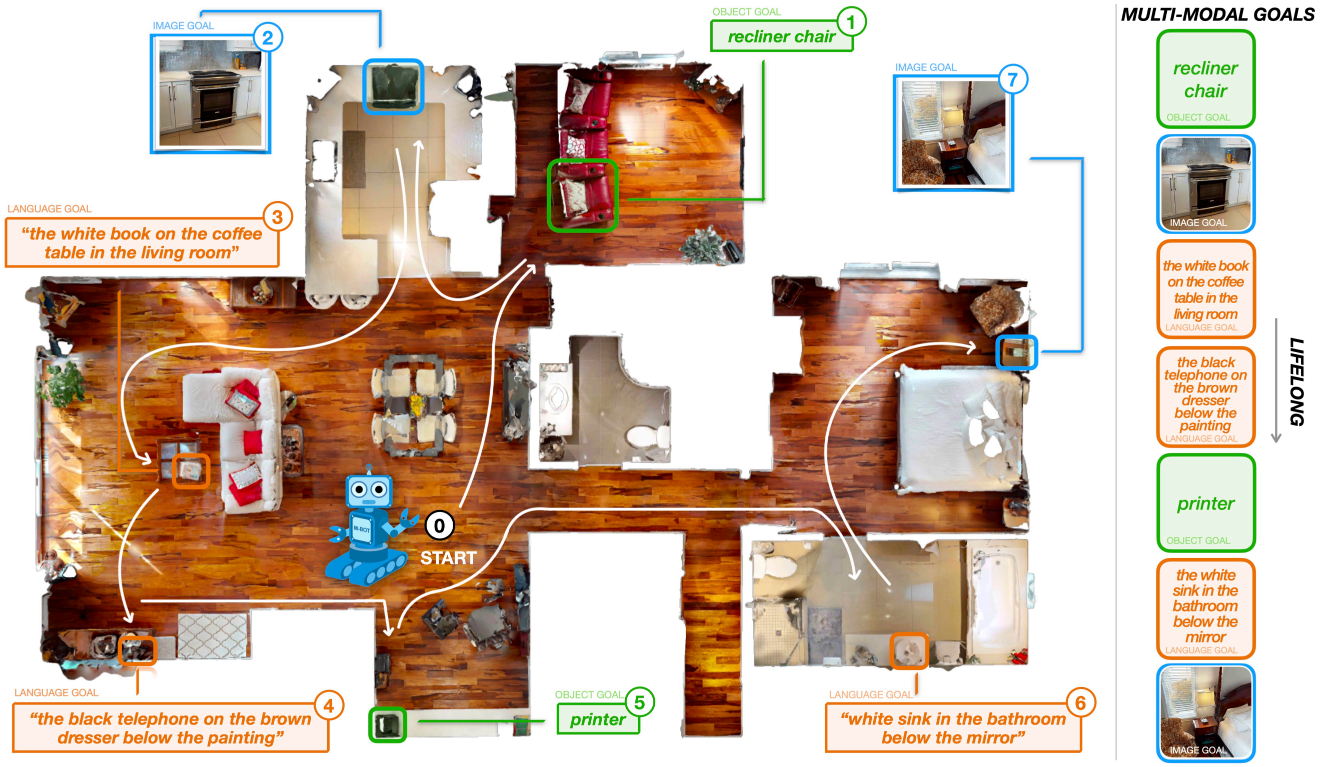
IMAGE GOAL
2
OBJECT GOAL
recliner chair
1
MULTI-MODAL GOALS
IMAGE GOAL
7
LANGUAGE GOAL
3
"the white book on the coffee table in the living room"
0
START
M-BOT
LANGUAGE GOAL
4
"the black telephone on the brown dresser below the painting"
OBJECT GOAL
5
printer
LANGUAGE GOAL
6
"white sink in the bathroom below the mirror"
recliner chair
OBJECT GOAL
IMAGE GOAL
the white book on the coffee table in the living room
LANGUAGE GOAL
the black telephone on the brown dresser below the painting
LANGUAGE GOAL
printer
OBJECT GOAL
the white sink in the bathroom below the mirror
LANGUAGE GOAL
IMAGE GOAL
LIFELONG

Figure 1. We study the Go to Any Thing (GOAT) task, which involves agents navigating to a sequence of open vocabulary goals specified through any of the three modalities – category name, a language description, or an image. We propose GOAT-Bench, a benchmark for the GOAT task, where we evaluate modular and monolithic, explicit and implicit map-based navigation approaches. In the above example, we task the agent with sequentially navigating to 1) a recliner chair (from a closed set of k categories), 2) the oven shown in the picture, 3) *"the white book on the coffee table in the living room"*, and some other objects in the scene. The goal of the benchmark is to facilitate progress towards building such universal, multi-modal, lifelong agents.

Abstract

The Embodied AI community has made significant strides in visual navigation tasks, exploring targets from 3D coordinates, objects, language descriptions, and images. However, these navigation models often handle only a single input modality as the target. With the progress achieved so far, it is time to move towards universal navigation models capable of handling various goal types, enabling more effective user interaction with robots. To facilitate this goal, we propose GOAT-Bench, a benchmark for the universal navigation task referred to as GO to AnyThing (GOAT). In this task, the agent is directed to navigate to a sequence of targets specified by the category name, language description, or image in an open-vocabulary fashion. We benchmark monolithic RL and modular methods on the GOAT task, analyzing their performance across modalities, the role of explicit and implicit scene memories, their robustness to noise in goal specifications, and the impact of memory in lifelong scenarios.

*Equal contribution

1. Introduction

In recent years, the Embodied AI community has established standardized evaluation metrics and benchmarks for navigation [1–4] and developed novel algorithms and architectures [5–8]. Notably, four different variants of navigation have emerged, depending on how the goal is specified – point-goal navigation (PointNav) [9–12], object-goal navigation (ObjectNav) [13–16], image-goal navigation (ImageNav) [17–20], and langauge-goal navigation (referring expression or step-by-step instructions) [3, 21].

While significant progress has been made in task-specific solutions for these tasks, it is time to systematically study universal navigation methods capable of seamlessly handling goals across all of these modalities. Such a universal navigation system is crucial, as it is infeasible to specify all types of goals by a single modality. For instance, consider image-goal navigation [18], where users specify the goal using an image of the target object. Capturing images of all objects within a house is infeasible if users intend to deploy the robot in a household setting. In object-goal navigation (e.g., [13]), providing the object category alone might lack the required specificity. For instance, a user might need a *plate with a red pattern*, and merely providing the *plate* category is insufficient to convey this level of detail.

In addition, prior works on navigation have focused on building solutions for episodic settings, *i.e.* in each episode the agent is spawned in an indoor environment and tasked with navigating to one instance of an object category with no past memory from the environment. However, in real-world scenarios these agents will predominantly operate in indoor environments for extended periods of time (*i.e.* a lifelong setting), where we expect them to leverage past experiences within the same environment to become efficient over time. Doing so requires the ability to recall previously encountered objects and specific areas within houses, enabling them to navigate more efficiently when a new goal is specified.

Towards developing a universal, multi-modal, lifelong navigation system, we introduce a benchmark named GOAT-Bench, designed to accommodate target object specifications across multiple modalities and be capable of leveraging past experiences in the same environment *i.e.* operate lifelong. Fig. 1 illustrates an example episode in GOAT-Bench. An embodied agent is spawned in a new environment and tasked with locating a recliner chair (object category goal) initially. Subsequently, it is directed to find an oven specified through an image (image goal). It is then instructed to locate *"the white book on the coffee table in the living room"* (language goal) and subsequently find other objects throughout the scene. We construct GOAT-Bench using 181 HM3DSem [22] scenes, 312 object categories, and 680k episodes. GOAT-Bench has two notable features:

- **Open vocabulary, muti-modal goals**: it is an open vocabulary benchmark, enabling the incorporation of a broad range of targets, including those not encountered during training. This is a departure from prior work, that is often limited to a small set of 6 to 21 categories [13, 14, 23, 24].
- **Lifelong**: each episode consists of 5 to 10 targets specified through distinct modalities (*i.e.* image, object, or language goal). This contrasts with most prior navigation benchmarks where the scene is reset after a target is reached, providing a benchmark for evaluating lifelong learning.

We compare two classes of methods in our benchmark: a.) Sensors-to-Action using Neural Network (SenseAct-NN): Neural network policies trained using end-to-end RL (with and without implicit memory), b.) Modular Learning methods: chaining separate modules for each task component (exploration, last-mile navigation, and object detection) to solve the task (with explicit memory). We find SenseAct-NN methods achieve overall higher success rates ($2.9 - 4.6\%$ better) compared to modular methods, but achieve poor efficiency ($4.7 - 9.2\%$ worse) as measured by Success Weighted by Path Length (SPL). This can be attributed to the inability of SenseAct-NN methods to build/leverage implicit map representations. In contrast, modular methods which leverage semantic maps are more effective. These results highlight an area for future research - building effective memory representations for SenseAct-NN methods.

Our comprehensive analysis underscores the general importance of memory representations for improving efficiency of both SenseAct-NN and modular methods on the GOAT task. Specifically, we find that when given access to memory, the efficiency (SPL) of both SenseAct-NN and modular methods improves for subtasks in later stages of an episode (∼1.9x for SenseAct-NN and ∼1.5x for modular). We also investigate how performance of these methods vary across different modalities. We find these methods perform poorly on language and image goals, particularly when relying on CLIP [25] features. This suggests the inability of CLIP [25] features in capturing crucial instance-specific and spatial features in language and image goals. In addition, we also study how robust these methods are to noise in goals specified to the agent – by adding gaussian noise to image goals, paraphrasing language goals, and using synonyms for object goals (e.g., sofa → couch). We find SenseAct-NN methods to be more robust to noise compared to modular methods, with a smaller drop in performance (Sec. 7.4).

To summarize, our contributions are:

- A novel reproducible benchmark for building and evaluating multi-modal lifelong navigation systems.
- Benchmarking of modular and end-to-end trained methods with and without memory representations.
- A comprehensive analysis of these methods on memory dependency, performance across modalities, and robustness to noise.

2. Related work

Navigation in virtual environments. In recent years, most benchmarks have assessed navigation performance on individual goal types (image, object, language, and 3D coordinates) [6, 7, 20, 26–32]. Methods on these benchmarks often use modality-specific goal encoders or recognition modules. For example, one-hot encodings or detections for object goals, SuperGLUE-based keypoint matching [18, 33] or cross-view consistent encoders [34] for image goals, or linear projection for 3D coordinates [6]. These approaches are tailored to individual modalities and are unable to generalize across modalities out-of-the-box. In contrast, our focus is to study the performance of general-purpose architectures that can handle multiple modalities. Recent efforts [35] have tackled this problem by leveraging advances in vision-and-language aligned models (CLIP [25]) to bridge this gap by using a single goal encoder for handling image and object goals. However, [35] does not focus on longer natural language descriptions that are required to disambiguate and identify specific object instances. Additionally, as we show in our experiments (Sec. 6.1), CLIP goal encodings don't help when navigating to specific object instances.

Embodied Multi-Modal Benchmarks. Existing embodied tasks [36–39] require embodied agents to work with inputs from multiple modalities (language, vision, audio, etc) but they seldom have agents leveraging past experiences from the same environment, *i.e*. through lifelong agent scenarios. For example, ALFRED [36] involves following instructions to achieve long-horizon tasks and EmbodiedQA [37] requires an agent to answer a question by exploring or interacting with the environment. Both of these tasks require agents to leverage multi-modal inputs (language and image) but they are studied in single episode settings. In contrast, our primary focus is on navigation agents capable of understanding multi-modal open-vocabulary goals in lifelong scenarios. Most similar to our work is [40], where an agent is tasked to navigate to multiple objects from a closed-set of object categories in the same environment but a key difference in our work is that goals in the GOAT task are multi-modal (object category, image, and language description).

Concurrent Work. In tandem with our efforts, there is concurrent work that proposes the HM3D Open-Vocabulary ObjectNav task [41]. In contrast to object category-based single-goal-per-episode setup in [41], we focus on navigating to a sequence of goals specified across three different modalities. Similarly, there is concurrent work that proposes a modular system for solving the GOAT task in real world houses for a closed set of 15 object categories [42]. In contrast to [42], our work focuses on a practical, open-vocabulary setting and contributes a reproducible benchmark that the community can use to facilitate progress towards universal navigation agents. Having access to a reproducible benchmark in simulation allows us to ask and answer questions about various aspects of these universal navigation agents, such as the role of effective memory representations, compare against existing and future methods, and analyze robustness of these methods to noise across modalities. Such questions remain unanswered in [42] due to the time-intensive nature of real world evaluations. Furthermore, the efforts to construct such a reproducible benchmark aligns with the objectives of [42] and should be viewed as a complementary effort, meant to augment, not replace, real-world benchmarking.

3. Task

In the Go to Any Thing (GOAT) task, an agent is spawned randomly in an unseen indoor environment and tasked with sequentially navigating to a variable number (in 5-10) of goal objects, described via the category name of the object (*e.g*. 'couch'), a language description (*e.g. "a black leather couch next to coffee table"*), or an image of the object uniquely identifying the goal instance in the environment. We refer to finding each goal in a GOAT episode as a *subtask*. Each GOAT episode comprises 5 to 10 subtasks.

We set up the GOAT task in an open-vocabulary setting; unlike many prior works, we are not restricted to navigating to a predetermined, closed set of object categories [13, 24, 40, 43, 44]. The agent is expected to reach the goal object g^k for the k^{th} subtask as efficiently as possible within an allocated time budget. Once the agent completes the k^{th} subtask by reaching the goal object or exhausts the allocated time budget, the agent receives next goal g^{k+1} to navigate to. This contrasts with most prior navigation benchmarks [13, 24, 44, 45] where the scene/episode is reset once the agent reaches the goal. Chaining multi-modal navigation goals enables us to benchmark lifelong learning methods that leverage past agent experience in the same environment.

We use HelloRobot's Stretch robot embodiment for the GOAT agent. The agent has a height of 1.41m and base radius of 17cm. At each timestep, the agent has access to an 360 x 640 resolution RGB image I_t, depth image D_t, relative pose sensor with GPS+Compass information $P_t = (\delta x, \delta y, \delta z)$ from onboard sensors, as well as the current subtask goal g_t^k, $k \; \forall \; \{1, 2, ..., 5 - 10\}$. The agent's action space comprises MOVE_FORWARD (by 0.25m), TURN_LEFT and TURN_RIGHT (by 30º), LOOK_UP and LOOK_DOWN (by 30º), and STOP actions. A sub-task in a GOAT episode is deemed successful when the agent calls STOP action within 1m euclidean distance from the current goal object instance – within a budget of 500 agent actions (per sub task).

4. Dataset

In this section, we describe the procedure used to build an open-vocabulary GOAT-Bench. We use real-world 3D scans from HM3DSem [22], consisting of 145 training and 36 validation scenes. In total, GOAT-Bench consists of 264

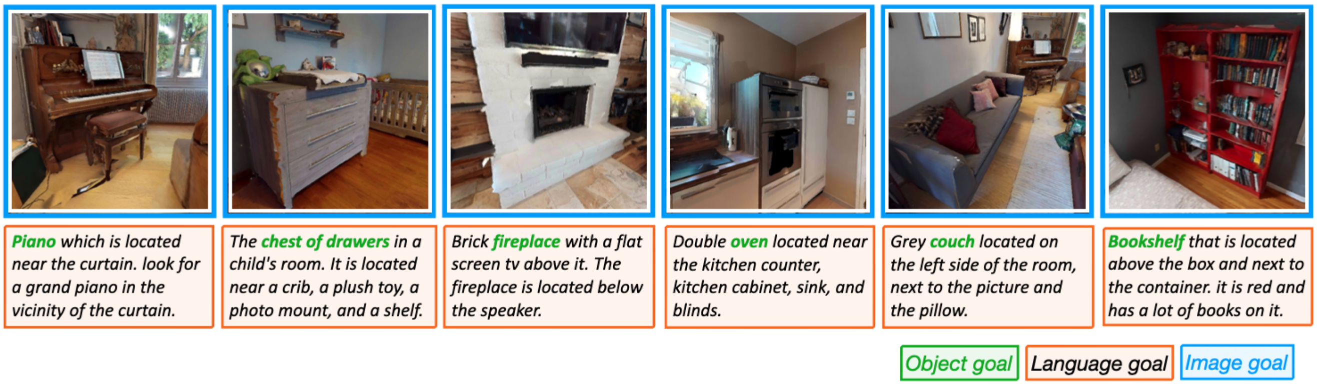
Piano which is located near the curtain. look for a grand piano in the vicinity of the curtain.
The chest of drawers in a child's room. It is located near a crib, a plush toy, a photo mount, and a shelf.
Brick fireplace with a flat screen tv above it. The fireplace is located below the speaker.
Double oven located near the kitchen counter, kitchen cabinet, sink, and blinds.
Grey couch located on the left side of the room, next to the picture and the pillow.
Bookshelf that is located above the box and next to the container. it is red and has a lot of books on it.
Object goal
Language goal
Image goal

Figure 2. **Preview of the GOAT-Bench dataset.** We show multi-modal examples of goal instances from the dataset: images of objects (blue), language descriptions (orange) and object category annotations (green).

	TRAIN		VAL SEEN		VAL SEEN SYNONYMS		VAL UNSEEN	
Dataset	Categories	Goals	Categories	Goals	Categories	Goals	Categories	Goals
RoboTHOR Challenge [13]	12	420	12	105	–	–	–	–
ObjectNav-MP3D [23]	21	7509	21	1316	–	–	–	–
ObjectNav-HM3D [14]	6	5216	6	1168	–	–	–	–
InstanceImageNav-HM3D [24]	6	3516	6	780	–	–	–	–
OVON [41]	280	10987	79	2219	50	1177	49	1278
GOAT-Bench	193	13025	52	1760	31	877	36	1282

Table 1. **Dataset statistics for popular embodied navigation benchmarks.** GOAT-Bench has at least ∼9x more object categories for training (21 vs 193) and about ∼6x more for validation (21 vs 119) than prior closed-set navigation benchmarks.

training categories and a total of $\sim 13k$ goal specifications across a total of $725k$ training episodes (with $5k$ episodes per training scene). This is in contrast with most prior embodied navigation datasets that focus on a closed set of 6 to 21 object categories – with goals for one modality. We compare the scale of our dataset against prior work in Tab. 1. GOAT-Bench has about 9x more object categories for training and about 6x more for validation than prior closed-set datasets. Next, we describe how we generate goals for each modality.

Open-Vocabulary ObjectNav goals (OVON). The OVON task from HM3D-OVON [41] has embodied agents navigate to object goals from an open vocabulary (as opposed to from a fixed set). This involved extending the Object-Goal Navigation task from [22, 44] to an open-vocabulary setting with hundreds of categories by leveraging the dense semantic annotations provided in HM3DSem [22]. Specifically, HM3D-OVON[41] extends the 6 category OBJECTNAV dataset from [22] to 280 object categories for training and 179 object categories for evaluation (similar to [41]). To do so, they leverage the ground truth semantic annotations from HM3DSem dataset [22] and sample objects which are large enough to be visible, *i.e.* objects with frame coverage $\geq 5\%$ from any viewpoint within $1m$ of the object. Frame coverage refers to the ratio of goal object's pixels to the total number of pixels. We use these goals from [41] – including both seen and unseen categories – to test generalization to novel objects (see supplementary for full list).

Open-Vocabulary Instance-ImageNav goals (OVIIN). INSTANCE IMAGENAV [24] is the task of navigating to object instances specified by images for the canonical 6 OBJECTNAV categories in HM3D scenes [22, 46]. However, in this work, we are interested in studying generalization to a wide variety of novel, unseen instances and categories of image goals. We do so by extending the INSTANCE IMAGENAV task to an open-vocabulary setting by creating training and evaluation splits using same heuristics as OVON goals to build the OVIIN goals. As a result, we generate a total of $7.7k$ image goal instances across 264 training categories and $2.9k$ instances across 164 validation categories for evaluation. We show some samples from the OVIIN dataset in Fig. 2. Refer Appendix A for details on goal image sampling.

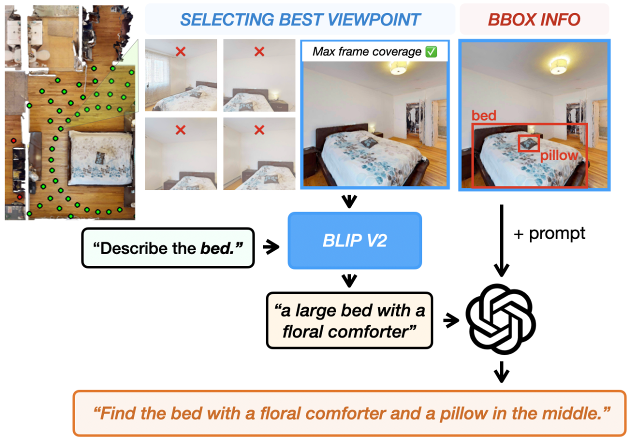
SELECTING BEST VIEWPOINT
BBOX INFO
Max frame coverage ✅
bed
pillow
"Describe the bed."
BLIP V2
+ prompt
"a large bed with a floral comforter"
"Find the bed with a floral comforter and a pillow in the middle."

Figure 3. **LanguageNav dataset generation pipeline.** We automatically generate language descriptions for object goals by leveraging VLMs, LLMs and ground truth information from simulator. We first capture an image of the goal object from a valid viewpoint. Next, we retrieve spatial and semantic information of the nearby objects from the simulator. We then prompt BLIP-2 [47] to extract appearance attributes of the object. These are then combined to prompt ChatGPT-3.5 to output a language description of the goal.

Language goals (LanguageNav). LanguageNav involves agents navigating to objects described by natural language (*e.g. "plate with a red flower pattern"*). Prior works in language-based navigation either provide verbose step-by-step instructions to reach a goal [3, 48] or limited human annotations for evaluations [21, 36, 48] to describe the goals. However, collecting human annotations for language descriptions for thousands of object instances and scaling them with increasing scene dataset size is challenging and expensive. Generating descriptions for these objects in an automated fashion, on the other hand, is also challenging. It requires distilling object's visual appearance, spatial, and semantic context into coherent sentences. This includes information about the object, its attributes (color, shape, material properties, etc.), its spatial relationship to surrounding objects, what room it is in, etc. To tackle these challenges, we present an automatic pipeline for generating language descriptions by leveraging ground truth semantic and spatial information from simulators along with reasoning capabilities of popular vision-and-language (VLM) and large language models (LLM). As shown in Fig. 3, for each object goal instance in OVON, we sample the viewpoint image of a goal with the maximum frame coverage (*i.e.* ratio of goal object's pixels to total number of pixels), extract semantic and spatial information from the simulator [9] such as names and 2D bounding box coordinates of visible objects. For each sampled goal instance, we prompt BLIP-2 [47] to extract attributes like color, shape, material, etc. Finally, we combine the spatial and semantic information from the simulator with object attribute metadata from BLIP-2 [47] predictions and use that to prompt ChatGPT-3.5 to output a language description of the object instance. Using this pipeline, we generate a total of $5.4k$ unique language goal instances from 225 training categories and $1.9k$ goal instances from 137 held out categories. We show examples for these in Fig. 2 and Appendix B.

GOAT-Bench Dataset Episode Generation. Combining the above mentioned open vocabulary datasets provides us with (category name, language description, and image) tuples associated with each goal object instance. These are used to generate training and evaluation episodes with multi-modal goal specifications for lifelong navigation. Each episode consists of a scene, the agent's starting position (at timestep $t = 0$), and sequences of 5 to 10 (sub-task) goals across three modalities. To generate each episode, we first uniformly sample a number of subtasks between 5 to 10. For each subtask, we uniformly sample a goal modality (category, description, or image), and then randomly sample a goal instance – uniformly across all categories. We then randomly sample a starting position that satisfies the following constraints: a.) all subtask goal locations are on the same floor as the starting position as we do not expect the agent to climb stairs, and b.) distance to nearest goal location for first subtask must lie between $1m$ to $30m$. This is similar to the episode generation process for the OBJECTNAV task [43]. See Fig. 1 for an example of what goals for a single episode look like. Following this procedure, we generate $5k$ train GOAT episodes (25 to $50k$ subtasks) per scene for 145 training scenes. For the validation set, we generate 10 GOAT episodes (50 to 100 subtasks) per scene for 36 val scenes.

Evaluation Splits. To test generalization of navigation agents we evaluate these agents in unseen environments, which means each goal instance is novel. In addition, to test generalization to objects at different levels, we generate 3 evaluation splits: VAL SEEN, VAL SEEN SYNONYMS, and VAL UNSEEN by manually segregating object categories depending on whether they were observed during training.

- VAL SEEN - goals generated using object categories seen during training.
- VAL SEEN SYNONYMS - goal categories synonymous to those seen during training (*i.e.* "couch" category seen during training, evaluated on "sofa" during evaluation).
- VAL UNSEEN - goals generated using object categories not seen during training.

5. Baselines

In this section, we present multi-modal policies trained on the GOAT task using the HM3DSem [22] scene dataset in the Habitat simulator [9]. We benchmark two types of methods: 1) Modular methods: semantic mapping and planning-based, and 2) Reinforcement Learning: sensor-to-action using neural network (SenseAct-NN) policies trained using RL.

5.1. Modular Baseline

Modular navigation approaches have emerged as a popular paradigm for training policies for various Embodied AI tasks [7, 15, 20, 49–56]. The key idea in these approaches is to decouple low-level control for navigation from goal recognition. This allows us to have separate modules dedicated for each task component (like detection, exploration, last-mile navigation), which are then chained to solve the task. Prior work [50] builds a top-down semantic map by projecting first-person semantic predictions with depth. It then selects an exploration goal on the semantic map using the goal query through a learned or heuristic exploration policy, and plans low-level actions to the goal.

Modular GOAT [42] extends prior modular navigation methods [7, 20, 49] to handle multi-modal goal prompts (*i.e.* object category, language, and image goals). Specifically, they build an instance-specific memory (alongside the semantic memory) by clustering together projected pixels of same categories on the semantic (top-down) map [42]. This instance memory captures egocentric views and CLIP features of object instances seen during exploration. Depending on the current goal modality, the agent then matches the current goal (image or CLIP embedding of description) with object instances – through keypoint-matching for image-to-image matching and cosine similarity for language-to-image feature matching. Instances with the best matching score are then localized and marked as goals for the agent to navigate

to. Note that this method assumes access to the object category information to filter out instances during goal matching.
Modular CLIP on Wheels (CoW). Similar to [42], we also present results with CoW [57], that uses only CLIP features to match (image, object, and language) goals against all of the images seen during exploration to localize the goal.

5.2. SenseAct-NN Baselines

In addition to evaluating modular approaches, we also train sensor-to-action neural network policies using RL for the GOAT task. Specifically, we consider two methods:
SenseAct-NN Skill Chain. Learning a single sensor-to-action neural network (*i.e.* monolithic policies) using end-to-end RL for GOAT task is difficult due to the long horizon nature of the task. As an alternative, we train *individual* navigation policies for each GOAT subtask – ObjectNav, Instance ImageNav, and LanguageNav. We combine these using a high level planner which executes one of the available policies based on the navigation goal modality at each timestep. Specifically, we extend the policy architecture from [58] and use a simple CNN+RNN policy architecture. To encode RGB input (i_t = CNN(I_t)), we use a frozen CLIP [25] ResNet50 [59] encoder. The GPS+Compass inputs, $P_t = (\Delta x, \Delta y, \Delta z)$, and $R_t = (\Delta\theta)$, are passed through fully-connected layers $p_t = \text{FC}(P_t), r_t = \text{FC}(R_t)$ to embed them to 32-d vectors. Finally, we convert the goal observation to d-dimensional vector using a modality-specific goal encoder $g_t^{(m)} = \text{ENC}(G_t^{(m)})$. All of these input features are concatenated to form an observation embedding, and fed into a 2-layer, 512-d GRU at every timestep to predict a distribution over actions a_t - formally, given current observations $o_t = [i_t, p_t, r_t, g_t]$, $(h_t, a_t) = \text{GRU}(o_t, h_{t-1})$. For each subtask type in GOAT task we ablate the choice of visual encoder and goal encoder for training task-specific policies and choose the one which performs the best for that subtask (refer Sec. 6.1 for results). For ObjectNav goals, we use a frozen CLIP [25] text goal encoder and CLIP ResNet50 [59] visual encoder, and for LanguageNav, we use a BERT [60] sentence goal encoder and CLIP ResNet50 [59] visual encoder. For Instance ImageNav, we use the recently released CroCo-v2 [34] to generate cross-view consistent goal and visual embeddings. We train each of these policies using VER [61] till convergence on task-specific datasets (refer Appendix C.1 for training details).
SenseAct-NN Monolithic Policy. We also benchmark a monolithic sensor-to-action neural network policy trained using RL for the GOAT task. Training an effective multi-modal policy capable of leveraging past experience from previous GOAT subtasks requires two important properties: a.) a multimodal goal encoder which can map goals from different modalities into a common latent space for the policy (e.g. CLIP), and b.) an implicit or explicit memory representation for capturing past experience. For encoding the goals, we use CLIP [25] text and image encoders. Because CLIP is trained with a vision-and-language alignment loss, we expect it to output meaningful representations in a common latent space for effective goal encoding. Next, to leverage past experience we carry forward hidden state of the policy from last subtask $h_T^{(s_{t-1})}$ as initial hidden state for a new subtask $h_0^{(s_t)}$ in a single GOAT episode. Wijmans *et al.* [62] showed blind agents modeled using RNNs are capable of building map-like representations for the PointNav [45] task. Motivated by these experiments, we expect maintaining an RNN hidden state across subtask provides our policy with a implicit memory representation which can be effectively used for efficient navigation. Towards this end, we extend the policy from [58] to train a monolithic policy with CLIP as our goal encoder $g_t^{(m)} = \text{ENC}(G_t^{(m)})$ and maintain hidden states across subtasks during training. We train this policy using VER [61] for 500 million steps on GOAT train dataset; refer Appendix C.2 for more details.

6. Results

For our experiments, we report two metrics – success rate (SR) and Success Weighted by Path Length (SPL). Success rate represents the percentage of sub-tasks where the agent successfully navigates to a goal. Efficiency, on the other hand is measured using SPL [2] – where the shortest path for each sub-task is considered from the final location of the agent from the previous sub-task to the goal location for the current sub-task. For the first sub-task, this corresponds to the starting position of the episode.

6.1. Modular vs. SenseAct-NN approaches

We present comparisons between modular approaches (with explicit maps) and SenseAct-NN RL approaches (with and without implicit maps) in Tab. 2. In terms of success rate, we observe that the SenseAct-NN Skill Chain baseline (row 3) outperforms all other baselines (that do not use ground truth semantics, shown in row 1) across all three validation splits. It also appears to be generalizing better to unseen instances and categories – performing better than the modular baselines (row 2) by at least an average margin of about 4% across all splits. However, this baseline does not do as well on SPL – on average 6.6% lower than the best modular (GOAT) baseline. This is because it does not maintain any memory across sub-tasks to keep track of previously encountered objects and regions of the scene. Specifically, as we have separate navigation policies for each modality, the policy hidden state is not propagated across sub-tasks.

On the other hand, the Modular GOAT [42] (row 2), which maintains an explicit semantic and instance map of the environment, does much better on SPL (6.6% than SenseAct-NN Skill Chain and 10.9% better than SenseAct-NN Monolithic). After the agent has sufficiently explored the scene,

	VAL SEEN		VAL SEEN SYNONYMS		VAL UNSEEN	
Method	SR (↑)	SPL (↑)	SR (↑)	SPL (↑)	SR (↑)	SPL (↑)
GOAT-GTSem [42]	56.7	40.3	58.4	43.5	54.3	41.0
Modular GOAT [42]	26.3	**17.5**	33.8	**24.4**	24.9	**17.2**
Modular CLIP on Wheels [57]	14.8	8.71	18.5	11.5	16.1	10.4
SenseAct-NN Skill Chain	**29.2**	12.8	**38.2**	15.2	**29.5**	11.3
SenseAct-NN Monolithic	16.8	9.4	18.5	10.1	12.3	6.8

Table 2. **Results**. Comparison of end-to-end RL and modular methods on GOAT-Bench HM3D benchmark on 3 evaluation splits: 1) VAL SEEN, 2) VAL SEEN SYNONYMS, 3) VAL UNSEEN.

it is able to leverage this memory for localizing new goal instances in already seen parts of the map and navigating to them directly. Modular GOAT also does better than the modular CLIP on Wheels (CoW) baseline (row 3) – highlighting the usefulness of maintaining an instance-specific memory, using category information to filter instances, and relying on image keypoints instead of CLIP features for matching against image goals. To decouple limitations of Modular GOAT's perception module (for object detection and map building) from the instance-to-goal matching, heuristic planning, and last-mile navigation, we also show its results with ground truth semantics (row 1). This reflects an average improvement of ∼30% in success and ∼22.0% in SPL.

We also observe that the SenseAct-NN Monolithic policy (row 4) does not perform well compared to the other baselines. We hypothesize this is due to: 1.) CLIP's limited efficacy in capturing instance-specific features for language and image goals, 2.) difficulty of learning effective long horizon navigation using RL. Later, in Sec. 7.1, we also see that this policy performs much worse on image and language goals. This shows that policy is having difficulty improving on these sub-tasks, causing the average performance (across all sub-tasks to be low). This trend of poor instance-specific image-goal performance is also evident when comparing image-goal policies trained using CLIP features vs. CroCo-v2 image features [34] (refer Sec. 7.1 for additional analysis).

7. Analysis

Here, we further analyze the performance of the best-performing modular method against SenseAct-NN methods.

7.1. How do agents perform on each modality?

To understand how effective these agents are across the three modalities, we also plot modality-wise success rate and SPL numbers for the baselines on the VAL SEEN dataset in Fig. 4. For object goals, we observe that Modular GOAT [42] performs better on success rate than the SenseAct-NN Skill Chain and Monolithic baselines (29.4% vs 25.8% and 25.7%). In terms of efficiency, we see that both Modular GOAT and SenseAct-NN Skill Chain perform equally well, and better than the SenseAct-NN Monolithic baseline. For language goals, Modular GOAT performs better than SenseAct-NN Skill Chain on both – success (about 5% better) and SPL (more than 2x better). This speaks to limitations of CLIP embeddings for capturing instance-specific features.

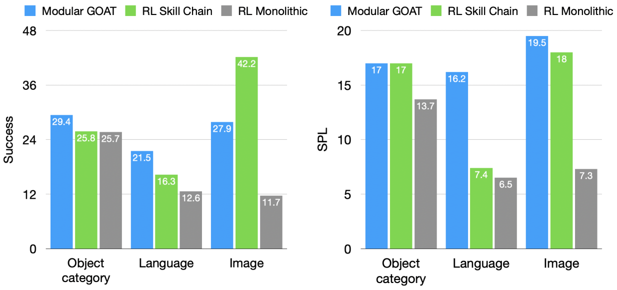

Figure 4. **Performance across types of modalities.** We breakdown the performance of all 3 baselines by modalities used subtask type: object category, language or image.

For image goals, we see that the CroCo-v2 Instance ImageNav policy used in the SenseAct-NN Skill Chain baseline outperforms Modular GOAT on success rate – by a huge margin of about 15%. On SPL, however, Modular GOAT does better because of its persistent memory. The SenseAct-NN Monolithic baseline, on the other hand, significantly underperforms on both success and SPL. As shown in Fig. 4, the ranks of the baselines using average performance is not indicative of performance across each modality. This is because, task-specific policies trained for SenseAct-NN Skill Chain baseline outperform other methods on OVIIN task and is comparable on OVON.

7.2. How important is memory for efficient navigation?

Memory or the ability to remember previously seen objects or parts of the house can enable agents to be more efficient at navigation. For instance, an agent that has already seen the kitchen is expected to navigate directly to it (without exploring) when asked to find an oven. For methods using memory (*i.e.* modular GOAT and monolithic policies), we evaluate the importance of memory towards success rate and efficiency by dropping the memory after each subtask. For Modular GOAT, we do this by building the map from scratch for each subtask, whereas for the SenseAct-NN Monolithic policy, we do this by dropping the hidden state between subtasks. This forces the policies to explore the environment from scratch for each subtask, and does not allow it to leverage past experience in the scene. As shown in Fig. 5, this results in a significant drop in SPL for Modular GOAT – by a factor of approximately 2x – from 17.6 to 9.4. The success rate also reduces by around 5% (from 26.4 to 21.2). As highlighted in [42], success rate drops because a prebuilt scene memory leads to improved instance-to-goal matching. The SenseAct-NN Monolithic policy, on the other hand, sees only a minor drop in SPL (from 9.4 to 9.0) and success (from 16.8 to 14.9) when memory is dropped. This suggests an inability (or lack of expressiveness) of the policy's hidden state to capture useful information about the explored scene. During evaluation, we often find the agent continuing to explore

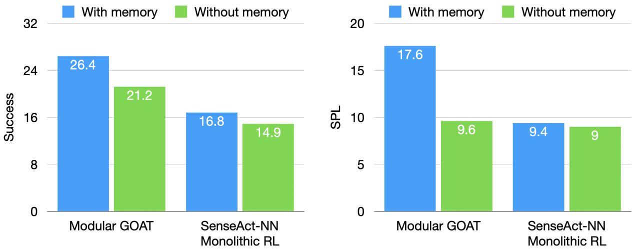

Figure 5. **Usefulness of memory**: We benchmark the drop in performance for when no memory is maintained across subtasks for modular GOAT [42] and SenseAct-NN Monolithic RL baselines.

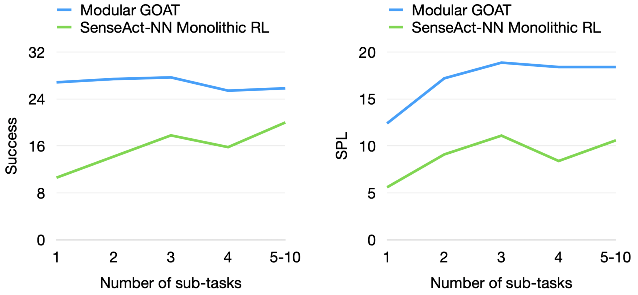

Figure 6. **Average performance over time in a GOAT episode.** We plot the success rate and SPL of memory based baselines against the number of subtasks completed in a GOAT episode.

the scene when asked to navigate to an object it has seen previously (see Appendix D for qualitative visualizations).

7.3. Does success and efficiency improve over time?

As agents perform more subtasks in the same environment, it is reasonable to expect them to get better over time. Efficient agents will ideally keep track of already seen objects and areas of the house and will have an internal model of paths to follow to reach already seen goals. To evaluate this, we plot average success and SPL over number of subtasks in an episode for these methods in Fig. 6.

We observe that for Modular GOAT, the success rate does not improve over subtasks, whereas the SPL does see gains over the first three subtasks (from 12.4 to 18.7) before it roughly saturates (at 18.4). For the SenseAct-NN Monolithic policy, both SPL and success rate does see gains over time, from 5.6 to 10.6 on SPL and 10.6% to 20.0% on success. These results highlight the importance of having effective memory representations (implicit or explicit) to perform efficiently on the GOAT task.

7.4. How robust are these methods to noise in goal specifications?

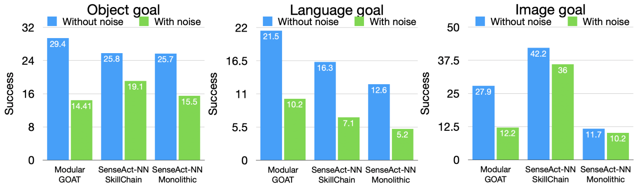

Figure 7. **Robustness to noise.** We breakdown the effect of noise on performance of different baselines by goal modality.

Goal specifications in real-world scenarios can often contain a lot of noise. Images of goal object can be noisy (for example in low-lit scenes), users might use uncommon synonyms to describe object categories, or they might phrase descriptions of instances differently. To simulate this type of noise in goal specifications, we perturb the goal inputs of the three modalities in the following ways. We add gaussian noise ($\mu = 0, \sigma = x \sim \mathcal{U}(0.1, 2.0)$) to goal images, replace object category names with corresponding synonyms and paraphrase language descriptions of instances (using ChatGPT). We evaluate the baselines on the VAL SEEN split of the dataset and report their performance and robustness to noise – across three modalities – in Fig. 7.

For object goals, we observe that Modular GOAT faces the biggest drop in performance when the object categories are replaced with synonyms. We find that this is because the object detector (DETIC [63] here) performs poorly on detecting these relatively uncommon synonyms. On the other hand, the skill chain and monolithic baselines don't suffer as much because they use CLIP goal embeddings (which capture these concepts better). For language goals, all three baselines suffer a reasonable drop in success rate. This can be attributed to the lack of instance-specific expressiveness of CLIP embeddings that are used as goal embeddings for the RL baselines and for goal matching in Modular GOAT. For image goals, the SenseAct-NN methods suffer very little with gaussian noise. This speaks to the robustness of the visual features from cross-view consistent representations from the CroCo-v2 encoder [34] used for the SenseAct-NN Skill Chain baseline and CLIP used for the monolithic policy. Overall, we observe that SenseAct-NN Skill Chain baseline is the most robust to noise (with a 25% average drop in success), whereas GOAT is the least robust (53% drop).

8. Conclusion

In this work, we propose GOAT-Bench, a novel reproducible benchmark for building and evaluating multi-modal lifelong navigation systems. We believe, this benchmark is a step towards building general purpose navigation agents that can handle multi-modal goals (*e.g.* image of an object, language description, and object categories) and leverage past experiences in the environment to perform the task efficiently. On GOAT-Bench, we benchmark two classes of methods, modular and end-to-end trained methods with and without memory representations. We find methods with effective memory representations perform well on GOAT task and achieve higher efficiency compared to methods without memory. In addition, we present a comprehensive analysis of dependency of these methods on memory, performance across modalities, and robustness to noise in goals specified via different modalities.

References

[1] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv*, 2018. 2

[2] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv*, 2020. 6

[3] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *CVPR*, 2018. 2, 5

[4] C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman, "Soundspaces: Audio-visual navigation in 3d environments," in *ECCV*, 2020. 2

[5] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," in *CVPR*, 2019. 2

[6] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," in *ICLR*, 2019. 3

[7] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in *NeurIPS*, 2020. 3, 5

[8] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "ViNT: A foundation model for visual navigation," in *CoRL*, 2023. 2

[9] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, *et al.*, "Habitat: A platform for embodied ai research," in *ICCV*, 2019. 2, 5, 14

[10] J. Ye, D. Batra, E. Wijmans, and A. Das, "Auxiliary tasks speed up learning pointgoal navigation," in *CoRL*, 2020.

[11] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, "Occupancy anticipation for efficient exploration and navigation," in *ECCV*, 2020.

[12] X. Zhao, H. Agrawal, D. Batra, and A. G. Schwing, "The surprising effectiveness of visual odometry techniques for embodied pointgoal navigation," in *ICCV*, 2021. 2

[13] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, *et al.*, "Robothor: An open simulation-to-real embodied ai platform," in *CVPR*, 2020. 2, 3, 4

[14] K. Yadav, J. Krantz, R. Ramrakhya, S. K. Ramakrishnan, J. Yang, A. Wang, J. Turner, A. Gokaslan, V.-P. Berges, R. Mootaghi, O. Maksymets, A. X. Chang, M. Savva, A. Clegg, D. S. Chaplot, and D. Batra, "Habitat challenge 2023." `https://aihabitat.org/challenge/2023/`, 2023. 2, 4, 12

[15] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal navigation with interaction-free learning," in *CVPR*, 2022. 5

[16] 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 *CVPR*, 2023. 2

[17] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *ICRA*, 2017. 2

[18] J. Krantz, T. Gervet, K. Yadav, A. Wang, C. Paxton, R. Mottaghi, D. Batra, J. Malik, S. Lee, and D. S. Chaplot, "Navigating to objects specified by images," in *ICCV*, 2023. 2, 3

[19] K. Yadav, A. Majumdar, R. Ramrakhya, N. Yokoyama, A. Baevski, Z. Kira, O. Maksymets, and D. Batra, "Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav," *arXiv*, 2023.

[20] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological slam for visual navigation," in *CVPR*, 2020. 2, 3, 5

[21] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen, and A. v. d. Hengel, "Reverie: Remote embodied visual referring expression in real indoor environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9982–9991, 2020. 2, 5

[22] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet, J. Turner, A. Gokaslan, N. Maestre, A. X. Chang, D. Batra, M. Savva, *et al.*, "Habitat-matterport 3d semantics dataset," *arXiv preprint arXiv:2210.05633*, 2022. 2, 3, 4, 5

[23] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," 2017. 2, 4

[24] J. Krantz, S. Lee, J. Malik, D. Batra, and D. S. Chaplot, "Instance-specific image goal navigation: Training embodied agents to find object instances," *arXiv preprint arXiv:2211.15876*, 2022. 2, 3, 4, 12

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021. 2, 3, 6, 13, 14

[26] M. Hahn, D. S. Chaplot, S. Tulsiani, M. Mukadam, J. M. Rehg, and A. Gupta, "No rl, no simulation: Learning to navigate without navigating," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[27] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal navigation with interaction-free learning," in *CVPR*, pp. 18890–18900, June 2022.

[28] R. Ramrakhya, E. Undersander, D. Batra, and A. Das, "Habitat-web: Learning embodied object-search strategies from human demonstrations at scale," in *CVPR*, 2022.

[29] J. Wasserman, K. Yadav, G. Chowdhary, A. Gupta, and U. Jain, "Last-mile embodied visual navigation," in *CoRL*, 2022.

[30] K. Yadav, R. Ramrakhya, A. Majumdar, V.-P. Berges, S. Kuhar, D. Batra, A. Baevski, and O. Maksymets, "Offline visual representation learning for embodied navigation," *arXiv preprint arXiv:2204.13226*, 2022.

[31] J. Ye, D. Batra, A. Das, and E. Wijmans, "Auxiliary tasks and exploration enable objectgoal navigation," in *CoRL*, pp. 16117–16126, 2021.

[32] Z. Al-Halah, S. K. Ramakrishnan, and K. Grauman, "Zero experience required: Plug & play modular transfer learning for semantic visual navigation," *arXiv preprint arXiv:2202.02440*, 2022. 3

[33] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *CVPR*, 2020. 3

[34] P. Weinzaepfel, T. Lucas, V. Leroy, Y. Cabon, V. Arora, R. Brégier, G. Csurka, L. Antsfeld, B. Chidlovskii, and J. Revaud, "CroCo v2: Improved Cross-view Completion Pretraining for Stereo Matching and Optical Flow," in *ICCV*, 2023. 3, 6, 7, 8, 14

[35] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," in *Neural Information Processing Systems (NeurIPS)*, 2022. 3

[36] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks," in *CVPR*, 2020. 3, 5

[37] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied Question Answering," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[38] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *CVPR*, 2018.

[39] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "Vima: General robot manipulation with multimodal prompts," in *ICML*, 2023. 3

[40] S. Wani, S. Patel, U. Jain, A. X. Chang, and M. Savva, "Multion: Benchmarking semantic map memory using multi-object navigation," 2020. 3

[41] N. Yokoyama, R. Ramrakhya, A. Kutumbaka, A. Das, S. Ha, and D. Batra, "Ovon: Open vocabulary objectgoal navigation benchmark," 2023. 3, 4, 12, 13, 14

[42] M. Chang, T. Gervet, M. Khanna, S. Yenamandra, D. Shah, S. Y. Min, K. Shah, C. Paxton, S. Gupta, D. Batra, R. Mottaghi, J. Malik, and D. S. Chaplot, "Goat: Go to any thing," 2023. 3, 5, 6, 7, 8, 15

[43] H. Team, "Habitat challenge, 2022." https://aihabitat.org/challenge/2022, 2020. 3, 5

[44] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "ObjectNav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020. 3, 4

[45] P. Anderson, A. X. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018. 3, 6

[46] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, "Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 4

[47] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *ICML*, 2023. 4, 5, 12

[48] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," *arXiv preprint arXiv:2010.07954*, 2020. 5

[49] D. S. Chaplot, S. Gupta, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural mapping," *8th International Conference on Learning Representations, ICLR 2020*, 2020. 5

[50] D. S. Chaplot, H. Jiang, S. Gupta, and A. Gupta, "Semantic curiosity for active visual learning," in *ECCV*, 2020. 5

[51] J. Krantz, A. Gokaslan, D. Batra, S. Lee, and O. Maksymets, "Waypoint models for instruction-guided navigation in continuous environments," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[52] D. S. Chaplot, M. Dalal, S. Gupta, J. Malik, and R. Salakhutdinov, "Seal: Self-supervised embodied active learning," in *NeurIPS*, 2021.

[53] G. Georgakis, B. Bucher, K. Schmeckpeper, S. Singh, and K. Daniilidis, "Learning to map for active semantic goal navigation," in *ICLR*, 2022.

[54] M. Hahn, D. Chaplot, S. Tulsiani, M. Mukadam, J. Rehg, and A. Gupta, "No rl, no simulation: Learning to navigate without navigating," 2021.

[55] S. Y. Min, D. S. Chaplot, P. Ravikumar, Y. Bisk, and R. Salakhutdinov, "Film: Following instructions in language with modular methods," in *International Conference on Learning Representations (ICLR)*, 2022.

[56] G. Sarch, Z. Fang, A. W. Harley, P. Schydlo, M. J. Tarr, S. Gupta, and K. Fragkiadaki, "Tidee: Tidying up novel rooms using visuo-semantic commonsense priors," in *European Conference on Computer Vision (ECCV)*, 2022. 5

[57] S. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *CVPR*, 2023. 6, 7

[58] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but Effective: CLIP Embeddings for Embodied AI," in *CVPR*, 2022. 6

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6, 14

[60] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," 2019. 6, 14

[61] E. Wijmans, I. Essa, and D. Batra, "Ver: Scaling on-policy rl leads to the emergence of navigation in embodied rearrangement," 2022. 6, 12, 14

[62] E. Wijmans, M. Savva, I. Essa, S. Lee, A. S. Morcos, and D. Batra, "Emergence of maps in the memories of blind navigation agents," in *ICLR*, 2022. 6

[63] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level super-

vision," in *ECCV*, 2022. 8

[64] G. Baruch, Z. Chen, A. Dehghan, T. Dimry, Y. Feigin, P. Fu, T. Gebauer, B. Joffe, D. Kurz, A. Schwartz, and E. Shulman, "Arkitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data," in *NeurIPS*, 2021. 14

[65] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," 2018. 14

[66] A. R. Zamir, T. Wekel, P. Agrawal, C. Wei, J. Malik, and S. Savarese, *Generic 3D Representation via Pose Estimation and Matching*, p. 535–553. Springer International Publishing, 2016. 14

[67] D. Lee, S. Ryu, S. Yeon, Y. Lee, D. Kim, C. Han, Y. Cabon, P. Weinzaepfel, N. Guérin, G. Csurka, and M. Humenberger, "Large-scale localization datasets in crowded indoor spaces," 2021. 14

[68] G. Bono, L. Antsfeld, B. Chidlovskii, P. Weinzaepfel, and C. Wolf, "End-to-end (instance)-image goal navigation through correspondence as an emergent phenomenon," 2023. 14

[69] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "Adaptformer: Adapting vision transformers for scalable visual recognition," 2022. 14