CoWs on PASTURE: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation

Samir Yitzhak Gadre[◇] Mitchell Wortsman[†] Gabriel Ilharco[†] Ludwig Schmidt[†] Shuran Song[◇]

Abstract

For robots to be generally useful, they must be able to find arbitrary objects described by people (i.e., be language-driven) even without expensive navigation training on in-domain data (i.e., perform zero-shot inference). We explore these capabilities in a unified setting: language-driven zero-shot object navigation (L-ZSON). Inspired by the recent success of open-vocabulary models for image classification, we investigate a straightforward framework, CLIP on Wheels (CoW), to adapt open-vocabulary models to this task without fine-tuning. To better evaluate L-ZSON, we introduce the PASTURE *benchmark, which considers finding uncommon objects, objects described by spatial and appearance attributes, and hidden objects described relative to visible objects. We conduct an in-depth empirical study by directly deploying 22 CoW baselines across* HABITAT, ROBOTHOR, *and* PASTURE. *In total, we evaluate over 90k navigation episodes and find that (1) CoW baselines often struggle to leverage language descriptions but are proficient at finding uncommon objects. (2) A simple CoW, with CLIP-based object localization and classical exploration—and no additional training—matches the navigation efficiency of a state-of-the-art ZSON method trained for 500M steps on* HABITAT *MP3D data. This same CoW provides a 15.6 percentage point improvement in success over a state-of-the-art* ROBOTHOR *ZSON model.*[1]

1. Introduction

To be more widely applicable, robots should be language-driven: able to deduce goals based on arbitrary text input instead of being constrained to a fixed set of object categories. While existing image classification, semantic segmentation, and object navigation benchmarks like ImageNet-1k [65], ImageNet-21k [22], MS-COCO [45], LVIS [28], HABITAT [67], and ROBOTHOR [18] include a vast array of everyday items, they do not capture all objects that matter to people. For instance, a lost "toy airplane" may become relevant in a kindergarten classroom, but this object is not annotated in any of the aforementioned datasets.

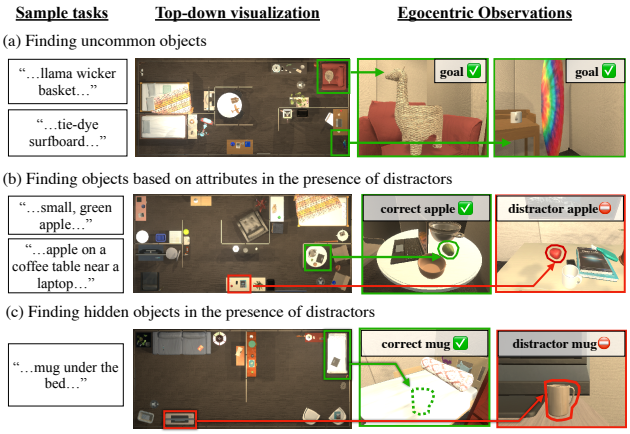

Figure 1. **The PASTURE benchmark for L-ZSON.** Text specifies navigation goal objects. Agents do not train on these tasks, making the evaluation protocol zero-shot. (a) *Uncommon object goals* like "llama wicker basket", not found in existing navigation benchmarks. (b) *Appearance, spatial descriptions*, which are necessary to find the correct object. (c) *Hidden object descriptions*, which localize objects that are not visible.

In this paper, we study *Language-driven zero-shot object navigation (L-ZSON)*, a more challenging but also more applicable version of object navigation [5, 18, 67, 79, 89] and ZSON [38, 46] tasks. In L-ZSON, an agent must find an object based on a description, which may contain different levels of granularity (e.g., "toy airplane", "toy airplane under the bed", or "wooden toy airplane"). L-ZSON encompasses ZSON, which specifies only the target category [38, 46]. Since L-ZSON is "zero-shot", we consider agents without access to navigation training on the target objects or domains. This reflects realistic application scenarios, where the environment and object set may not be known a priori.

Performing L-ZSON in *any* environment with *unstructured* language input is challenging; however, recent advances in open-vocabulary models for image classification [35, 58, 61], object detection [4, 21, 27, 36, 43, 47, 49, 62, 88], and semantic segmentation [3, 6, 15, 33, 36, 37, 86] present a promising foundation. These models provide an interface where one specifies—in text—the arbitrary ob-

[◇]Columbia University, [†]University of Washington. Correspondence to sy@cs.columbia.edu.

[1]For code, data, and videos, see cow.cs.columbia.edu/

jects they wish to classify, detect, or segment. For example, CLIP [61] open-vocabulary classifiers compute similarity scores between an input image and a set of user-specified captions (e.g., "a photo of a toy airplane.", ...), selecting the caption with the highest score to determine the image classification label. Given the flexibility of these models, we would like to understand their capability to execute embodied tasks even without additional training.

To this end, we present baselines and benchmarks for L-ZSON. More specifically:

- *A collection of baseline algorithms, CLIP on Wheels (CoW), which adapt open-vocabulary models to the task of L-ZSON*. CoW takes inspiration from the semantic mapping line of work [11, 41, 53], and decomposes the navigation task into exploration when the language target is not confidently localized, and target-driven planning otherwise. CoW retains the textual user interface of the original open-vocabulary model and does not require any navigation training. We evaluate 22 CoWs, ablating over many open-vocabulary models, exploration policies, backbones, prompting strategies, and post-processing strategies.
- *A new benchmark,* PASTURE*, to evaluate CoW and future methods on L-ZSON*. We design PASTURE, visualized in Fig. 1, to study capabilities that traditional object navigation agents, which are trained on a fixed set of categories, do not possess. We consider the ability to find (1) uncommon objects (e.g., "tie-dye surfboard"), (2) objects by their spatial and appearance attributes in the presence of distractor objects (e.g., "green apple" vs. "red apple"), and (3) objects that cannot be visually observed (e.g., "mug under the bed").

Together the CoW baselines and PASTURE benchmark allow us to conduct extensive studies on the capabilities of open-vocabulary models in the context of L-ZSON embodied tasks. Our experiments demonstrate CoW's potential on uncommon objects and limitations in taking full advantage of language descriptions—thereby providing empirical motivation for future studies. To contextualize CoW relative to prior zero-shot methods, we additionally evaluate on the HABITAT MP3D dataset. We find that our best CoW achieves navigation efficiency (SPL) that matches a state-of-the-art ZSON method [46] that trains on MP3D training data for 500M steps. On a ROBOTHOR object subset, considered in prior work, the same CoW beats the leading method [38] by 15.6 percentage points in task success.

2. Related Work

Mapping and exploration. Exploring effectively with a mobile robot is a long-standing problem in vision and robotics. Classical methods often decompose the task into map reconstruction [30, 32, 51, 52, 72], agent localization [17, 20, 54], and planning [41, 78]. Recent work investigates learned alternatives for exploration [7, 14, 56, 57, 63]. Here, agents are often trained end-to-end with self-supervised rewards (e.g., curiosity [57]) or supervised rewards (e.g., state visitation counts [25, 73, 75]). Learning-based methods typically need less hand-tuning, but require millions of training steps and reward engineering. We test both classical and learnable exploration strategies in the context of CoW to study their applicability to L-ZSON.

Goal-conditioned navigation. Apart from open-ended exploration, many navigation tasks are goal-conditioned, where the agent needs to navigate to a specified position (i.e., point goal [11, 12, 26, 31, 66, 77, 81, 83]), view of the environment (i.e., image goal [48, 64, 89]), or object category (i.e., object goal [1, 9, 10, 12, 19, 44, 74, 79, 84]). We consider an object goal navigation task.

Vision-Language Navigation. Prior work investigates language-based navigation, where language provides step-by-step instructions for the task [2, 34, 39, 40, 71]. This line of work demonstrates the benefits of additional language input for robot navigation, especially for long-horizon tasks (e.g., room-to-room navigation [40]). However, providing detailed step-by-step instructions (e.g., move 3 meters south [34]) could be challenging and time-consuming. In our L-ZSON task, an algorithm gets natural language as the goal description instead of low-level instructions. Prior work also investigates navigation with target descriptions in supervised settings [42, 60, 87]. In contrast, we explore a zero-shot evaluation protocol and consider finding hidden objects (e.g., "mug under bed") and uncommon objects (e.g., "tie-dye surfboard").

Zero-shot object navigation (ZSON). Recent work studies object navigation in zero-shot settings, where agents are evaluated on object categories that they are not explicitly trained on [38, 46]. Our task encompasses ZSON; however it also considers cases where more information—object attributes or hidden objects descriptions—is specified. Khandelwal *et al*. [38] train on a subset of ROBOTHOR categories and evaluate on a held-out set. In concurrent work, Majumdar *et al*. [46] train on an image goal navigation task and evaluate on object navigation downstream by leveraging CLIP multi-modal embeddings. Both algorithms necessitate navigation training for millions of steps and train separate models for each simulation domain. In contrast, CoW baselines do not necessitate any simulation training and can be deployed in multiple environments.

3. The L-ZSON Task

Language-driven zero-shot object navigation (L-ZSON) involves navigating to goal objects, specified in language, without explicit training to do so. Let $\mathcal{O}$ denote a set of

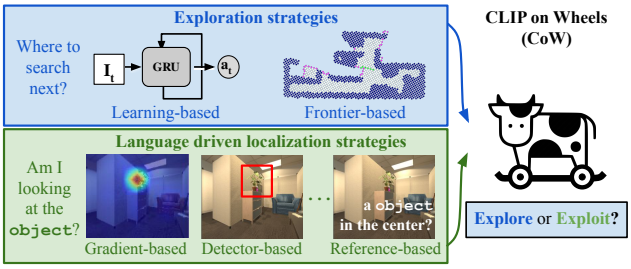

Figure 2. **CLIP on Wheels (CoW) overview.** A CoW uses a policy to explore and an object localizer (e.g., an open-vocabulary object detector) to determine if an object goal is in view.

natural language descriptions of target objects with potentially many attributes (e.g., "plant", "snake plant", "plant under the bed", etc.). This contrasts with definitions studied in prior object navigation [5, 18] and ZSON [38, 46] tasks, which focus on high-level categories like "plant". Let $\mathcal{S}$ denote the set of navigation scenes. Let p_0 describe the initial pose of an agent. A navigation episode $\tau \in \mathcal{T}$ is written as a tuple $\tau = (s, o, p_0)$, $s \in \mathcal{S}, o \in \mathcal{O}$. Each τ is a *zero-shot task* as tuples of this form are not seen during training. Starting at p_0, an embodied agent's goal is to find o. The agent receives observations and sensor readings I_t (e.g., RGB-D images). At each timestep t, the agent executes a navigation action $a \in \mathcal{A}$. A special action STOP $\in \mathcal{A}$ terminates the episode. If the agent is within c units of o and o meets a visibility criteria, the episode is successful.

4. CLIP on Wheels (CoW) Baselines

To address L-ZSON, we investigate a simple baseline approach, CoW, which adapts open-vocabulary models like CLIP to make them suitable for the task. A CoW takes as input an egocentric RGB-D image and an object goal specified in language. As a CoW moves, it updates a top-down map of the world created using RGB-D observations and pose estimates (Sec. 4.1). Each CoW gets an exploration policy and a zero-shot object localization module as seen in Fig. 2. To observe diverse views of the scene, a CoW explores using a policy (Sec. 4.2). As the CoW roams, it keeps track of its confidence about the target object's location using an object localization module (Sec. 4.3) and its top-down map. When a CoW's confidence exceeds a threshold, it plans to the location of the goal and issues the STOP action. We now describe the ingredients used to make the CoWs evaluated in our experiments (Sec. 6).

4.1. Depth-based Mapping

As a CoW moves, it constructs a top-down map using input depth, pose estimates, and known agent height. The map is initialized using known camera intrinsics and the first depth image. Since a CoW knows the intended consequences of its actions (e.g., MOVEFORWARD should result in a 0.25m translation), each action is represented as a pose delta transform to approximate a transition. To deal with noise associated with actuation or depth, a CoW maintains a map at 0.125m resolution. To improve map accuracy, a CoW checks for failed actions by comparing successive depth frames for movements (see Appx. A for details). Using known agent height (0.9m), map cells are projected to the ground plane to maintain a top-down representation of the world, which suffices for most navigation applications. Cells close to the floor are considered free space (white points in Fig. 3 (a)), while other cells are considered occupied (blue points in Fig. 3 (a)).

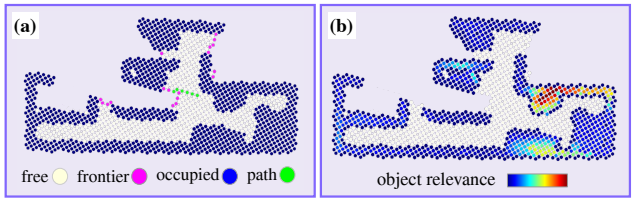

Figure 3. **Mapping.** Top-down map created from egocentric depth observations as a CoW roams a space. (a) Frontier Based Exploration [82] showing a planned path exploration path to the next frontier. (b) Back-projected object relevance scores provide object goal targets when a CoW has found an object.

4.2. Exploration

Exploration generates diverse egocentric views so a CoW is more likely to view the language-specified target object. We consider two exploration methods, *frontier-based* and *learning-based*.

Frontier based exploration (FBE) [82]. Using the top-down map discussed in Sec. 4.1, a CoW can navigate using a simple exploration heuristic: move to the frontier between free and unknown space to discover new regions. Once the navigator reaches a frontier (visualized as purple points in Fig. 3 (a)), it moves greedily to the next closest frontier. Since the map is updated at every timestep, a noisy pose estimate can contribute to inaccuracies. For example, narrow passages may collapse in the map due to pose drift. To circumvent such problems, we reinitialize the map when no paths exist to any frontiers in the map.

Learnable exploration. In addition to FBE, we consider learnable alternatives, which may explore more intelligently but incur substantial training costs. We investigate an architecture and reward structure similar to prior work in embodied AI (e.g., [25, 38, 75]). Specifically, we adopt a frozen CLIP backbone with a trainable GRU [16] and linear heads for the actor and critic networks. We train agents independently in HABITAT [67] and ROBOTHOR [18] simulation environments for 60M steps each, using DD-PPO [69, 77] in the AllenAct [76] framework. We employ a simple count-based reward [73]. All training scenes are disjoint from downstream navigation test scenes. For details on reward, hyperparameters, and training, see Appx. B.

4.3. Object Localization

Successful navigation depends on object localization: the ability to tell *if* and *where* an object is in an image. Regions of high object relevance, extracted from 2D images, are projected to the depth-based map (Fig. 3 (b)) where they serve as natural navigation targets. To determine if and when a target is in an image, we consider the following object localization modules, used in our experiments (Sec. 6). For more details see Appx. C.

Adapting open-vocabulary classifiers. We experiment with three strategies to turn CLIP [61] models into object localizers. First, we utilize the CLIP text encoder to embed k referring expressions, which specify regions where the target object may appear in the image. For example, "a plant in the top left of the image." We then match the language embeddings against a CLIP visual embedding for the current observation. We compute similarity between the image and text features to determine relevance scores over the regions. Second, we discretize the image into k smaller patches and obtain CLIP patch embeddings. We then convolve each patch embedding with a CLIP text embedding for the target object. If the object is in a patch, the relevance score for that patch should be high. Third, we modify an interpretability method [13, 70] designed to extract object relevancy from vision transformers (ViTs) [24]. Using a target CLIP text embedding and gradient information accumulated through the CLIP vision encoder, we construct a relevance map over input pixels, which qualitatively segments the target when it is in view.

Adapting open-vocabulary detectors and segmentors. In addition to CLIP-based methods, we consider two additional open-vocabulary models for object localization. First, the MDETR segmentation model [36], which extends the DETR detector [8] to take arbitrary text and images as input and output box detections. The base model is fine-tuned on PhraseCut [80], a dataset of paired masks and attribute descriptions, to support segmentation. Second, we consider the OWL-ViT detector [49], which uses a set prediction fine-tuning recipe to turn CLIP-like models into object detectors. We use this MDETR and OWL-ViT models to directly query images for targets.

Post-processing. The aforementioned models give continuous valued predictions. However, we are interested in binary masks giving if and where objects are in images. Hence, we threshold predictions for each model (see Appx. C for details). We further investigate two strategies for using the masks downstream: (1) using the entire mask or (2) using the center pixel. The second strategy is reasonable because only part of an object needs to be detected for successful navigation.

Target driven planning. Recall, CoWs have depth sensors. We back-project object relevance from 2D images into the depth-based map (Sec. 4.1). We keep only the max relevance for each map cell (Fig. 3 (b)). CoWs can then plan to high relevance areas in the map. See Appx. D for additional method visualization.

Figure 4. **Uncommon objects** in PASTURE.

Incorporating object priors. Since CoW does not train or fine-tune on navigation datasets, we investigate alternative approaches to inject object-level priors into the model. For each target object, we prompt GPT-3.5 [55] to generate rooms where the target objects are likely to be found. For example, GPT-3.5 states that apples are likely to be found in "kitchen" or "dining room" scenes. Following this prior, a GPT-3.5 enabled CoW first uses its object localization module to localize a kitchen or a dining room, and then looks for an apple. This straightforward extension, demonstrates how outside information can be incorporated into CoW.

5. The PASTURE Benchmark

To evaluate CoW baselines and future methods on L-ZSON, we introduce the PASTURE evaluation benchmark. PASTURE builds on ROBOTHOR validation scenes, which have parallel environments in the real-world. We target ROBOTHOR to facilitate future real-world benchmarking. PASTURE probes for seven capabilities explained in Sec. 5.1. We provide dataset statistics in Sec. 5.2.

5.1. PASTURE Tasks

PASTURE evaluates seven core L-ZSON capabilities.

Uncommon objects. Traditional benchmarks (e.g., ROBOTHOR and HABITAT MP3D) evaluate agents on common object categories like TVs; however, given the rich diversity of objects in homes, we would like to understand navigation performance on *uncommon objects*. Hence we add 12 new objects to each room. We use names shown in Fig. 4 as instance labels, which are minimal descriptions to identify each object. Some identifiers refer to text in images (e.g., "whiteboard saying CVPR") or to appearance attributes (e.g., "wooden toy airplane"). Other objects are less common in North America, like "maté", which is a popular Argentinian drink.

Appearance descriptions. To evaluate if baselines can take advantage of visual attributes, we introduce descriptions of the form: "{*size*}, {*color*}, {*material*} {*object*}". For

example: "small, red apple", "orange basketball", "small, black, metallic alarm clock". Objects are considered small if their 3D bounding box diagonal is below a threshold. We determine color and materials by inspection.

Spatial descriptions. To test if agents can leverage spatial information in navigation, we introduce descriptions: "{*object*} on top of {x}, near {y}, {z}, ...". For example, "house plant on a dresser near a spray bottle". To determine [on top of] relations, we use THOR metadata and to determine [nearness] we use a distance threshold between pairs of objects. We inspect all descriptions for correctness.

Appearance descriptions with distractors. To probe if appearance attributes better facilitate finding objects in the presence of distractors, we reuse the appearance captions from before, but evaluate on an modified environment with two visually distinct instances of each RoboTHOR object category. For example, for the task of finding a "red apple", we have both a red apple and a green apple in the room. A navigator must leverage appearance information—and not just class information—to successfully complete the task. Distractor objects are sufficiently far from the target objects so that finding a distractor cannot count as success.

Spatial descriptions with distractors. This capability is similar to the one above; however, we evaluate with spatial descriptions in the presence of distractor objects.

Hidden object descriptions. An ideal object navigator should find objects, even when they are hidden. Hence, we introduce descriptions: "{*object*} under/in {x}". For example, "basketball in the dresser drawers" or "vase under the sofa". We sample large objects (e.g., beds, sofas, dressers) in each scene to determine [under/in] relations. Additionally we remove visible instances of {*object*} from the room.

Hidden object descriptions with distractors. We use the hidden object descriptions from before, but reintroduce visible instances of {*object*} to serve as distractors. Consider finding a "mug under the bed". A distractor mug will also appear in the scene making the task more challenging.

5.2. Dataset Creation and Statistics

PASTURE contains three variations for each of the original 15 validation RoboTHOR rooms: uncommon objects added, additional object instances added, and target objects removed. For each of the seven settings presented above, we evaluate over 12 object instances in 15 rooms with two initial agent starting locations. Hence PASTURE consists of 2,520 tasks, which is a similar order of magnitude to RoboTHOR (1,800) and HABITAT MP3D (2,195) validation sets. For appearance attributes, 47% of the objects are considered "small". Each object gets an average of 1.2 color descriptors out of 22 possible choices, and 0.6 material descriptors out of 5 possible choices. Similarly, for spatial attributes, each object gets one object it is on top of or in (e.g., "vase in a shelving unit") and an average of 1.9 objects it is near. For a sample of appearance and spatial attributes see Fig. 1. For more dataset details and statistics see Appx. E.

6. Experiments

We first present our experimental setup, including the datasets, metrics, embodiment, and baselines considered in our study (Sec. 6.1). Then we present results on PASTURE, thereby elucidating the strengths and weakness of CoW baselines for L-ZSON (Sec. 6.2). Finally, we compare to prior ZSON art in RoboTHOR and HABITAT (MP3D) environments (Sec. 6.3).

6.1. Experimental setup

Environments. We consider PASTURE (Sec. 5), RoboTHOR [18], and HABITAT (MP3D) [67] validation sets as our test sets. We utilize validation sets for testing because official test set ground-truth is not publicly available. Domains are setup with noise that is faithful to their original challenge settings. For RoboTHOR—and by extension PASTURE—this means actuation noise but no depth noise. For HABITAT this means considerable depth noise and reconstruction artifacts, but no actuation noise.

Navigation Metrics. We adopt standard object navigation metrics to measure performance:

- SUCCESS (SR): the fraction of episodes where the agent executes STOP within 1.0m of the target object.
- Success weighted by inverse path length (SPL): Success weighted by the oracle shortest path length and normalized by the actual path length [5]. This metric points to the success efficiency of the agent.

In RoboTHOR and PASTURE, the target must additionally be visible for the episode to be a success, which this is not the case in HABITAT—as specified in their 2021 challenge.

Embodiment. The agent is a LoCoBot [29]. All agents have discrete actions: {MOVEFORWARD, ROTATERIGHT, ROTATELEFT, STOP}. The move action advances the agent by 0.25m, while rotation actions pivot the camera by 30°.

CoW Baselines. For exploration we consider policies presented in Sec. 4.2: FBE heuristic exploration, learnable exploration optimized on HABITAT (MP3D) train scenes, and learned exploration optimized on RoboTHOR train scenes. Learned exploration requires training in simulation—which is counter to our zero-shot goals; nonetheless, we ablate these explorers to contextualize their performance within the CoW framework. *FBE is the default CoW exploration strategy.*

For object localization, we consider:

- CLIP with $k = 9$ referring expressions (CLIP-Ref.)
- CLIP with $k = 9$ patches (CLIP-Patch)
- CLIP with gradient relevance (CLIP-Grad.)

ID	Localizer	Arch.	PASTURE Uncom. SR	Appear. SR	Space SR	Appear. distract SR	Space distract SR	Hid. SR	Hid. distract SR	Avg. SPL	Avg. SR	ROBOTHOR SPL	ROBOTHOR SR
▲	CLIP-Ref.	B/32	3.6	0.6	1.7	0.6	1.7	2.2	2.5	0.9	1.8	1.0	1.8
■	CLIP-Ref.	B/16	1.4	2.8	2.8	3.1	3.3	1.7	1.9	1.7	2.4	2.1	2.7
▲	CLIP-Patch	B/32	18.1	13.3	13.3	8.6	10.8	17.5	**17.8**	9.0	14.2	10.6	20.3
■	CLIP-Patch	B/16	10.6	11.4	7.8	10.8	8.1	16.4	15.6	7.7	11.5	9.7	15.7
▲	CLIP-Grad.	B/32	16.1	11.9	11.7	9.7	10.3	14.4	16.1	9.2	12.9	9.7	15.2
■	CLIP-Grad.	B/16	8.1	10.8	8.6	8.6	6.7	11.1	11.4	6.7	9.3	8.6	11.6
♦	MDETR	B3	3.1	7.2	5.0	6.9	4.7	8.1	8.9	5.4	6.3	8.4	9.9
▲	OWL	B/32	**32.8**	26.4	**19.4**	**19.4**	**16.1**	**19.2**	14.4	**12.6**	**21.1**	16.9	26.7
■	OWL	B/16	31.9	**26.9**	18.9	**19.4**	14.7	18.1	15.8	**12.6**	20.8	**17.2**	**27.5**
ProcTHOR fine-tune (supervised) [19]			n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	27.4	66.4

Table 1. **Benchmarking CoWs on PASTURE for L-ZSON.** On PASTURE we identify several key takeaways. (1) Average success on PASTURE is lower than on ROBOTHOR; however, CoWs are surprisingly good at finding uncommon objects (Uncom.), often finding them at higher rates than more common ROBOTHOR objects. (2) Comparing square (■) vs. triangle (▲) IDs, we see that architectures (Arch.) using *more compute* (i.e., ViT-B/16) often perform comparably or worse than their competitors (i.e., ViT-B/32). This is especially true for CLIP [61] models (indicated in pink, orange, and purple). (3) Blue OWL-ViT [49] models perform best. (4) PASTURE tasks with distractor objects (distract) hurt performance and natural language specification is not sufficient to mitigate against the added difficulties in these tasks. (5) A supervised baseline shown in gray significantly outperforms CoWs on ROBOTHOR; however, it is unable to support PASTURE tasks out-of-the-box.

- MDETR segmentation model (MDETR)
- OWL-ViT detector (OWL)

Descriptions of these models are in Sec. 4.3 and additional details are in Appx. C. All models are open-vocabulary. *No models are fine-tuned on navigation*, and hence we consider their inference *zero-shot* on our tasks.[2] We also consider various backbone architectures:

- A vision transformer [24], ViT-B/32 (▲ B/32)
- ViT-B/16 (■ B/16), which uses a smaller patch size of 16x16 and hence more compute.
- EfficentNet B3 (♦ B3), which is convolutional and similar in compute requirements to a ViT/B32.

For every model we evaluate with post-processing as the default setting, where only the center pixel of detections is registered in the top-down CoW map. Recall, this is a sensible strategy as only some part of the object needs to be found for an episode to be successful. We find that this decision improves performance on our best performing models from 0.1 to 6.0 percentage points on ROBOTHOR SUCCESS. For a full comparison between models with and without post-processing see Appx. F. For details on hyperparameters, learned agents, object localization threshold tuning, and CLIP prompt-tuning, see Appx. C, G.

End-to-end learnable baselines. We also compare against methods that are trained in simulation for millions of steps:

- *EmbCLIP-ZSON* [38] trains on eight ROBOTHOR categories, using CLIP language embeddings to specify the goal objects. At test time, the model is evaluated on four held-out object categories, which are also specified CLIP language embeddings for the target category names.
- *SemanticNav-ZSON* [46] trains models separately, one for each dataset, for image goal navigation. Image goals are specified with CLIP visual embeddings. At test time, image goals are swapped for CLIP language embeddings for the object goals. We compare to the MP3D model.

Both EmbCLIP-ZSON and SemanticNav-ZSON leverage multi-modal CLIP visual and language embeddings in learnable frameworks that require simulation training.

6.2. CoWs on PASTURE

Tab. 1 shows our main results of CoWs evaluated on PASTURE. For category-level results see Appx. H. We now discuss several salient questions.

How well can CoWs find common objects vs. uncommon objects? Comparing ROBOTHOR and uncommon (Uncom.) PASTURE success rate (SR) in Tab. 1—first and last columns—we notice that CoWs often find uncommon objects at higher rates than common ROBOTHOR objects (e.g., by ∼6 percentage points SUCCESS for the OWL ViT-B/32 CoW (▲)). We hypothesize that though uncommon objects are less prevalent in daily life, they are still represented in open-vocabulary datasets and hence recognizable for the object localization modules. We further ex-

[2]The claim is not that these models have never seen any synthetic data in their large-scale training sets, only that they are not trained to navigate.

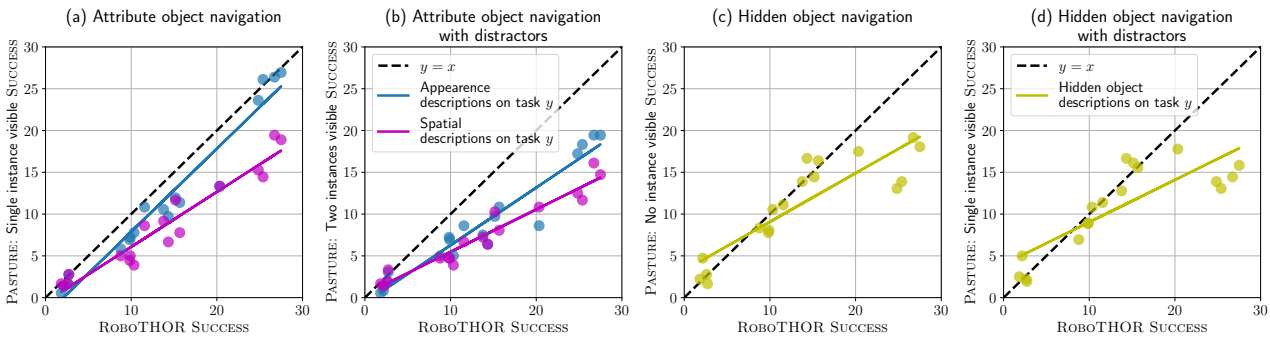

Figure 5. **PASTURE object navigation with descriptions.** In general object navigation with descriptions is more challenging than the ROBOTHOR object navigation task, as indicated by trend lines lying below the $y = x$ line. (a) Appearance descriptions are more helpful than spatial descriptions. (b) Performance further drops when distractor objects are introduced to the environment. However, CoWs are still able to make better use of appearance description than spatial descriptions. (c) Models in the lower success regime (<15% ROBOTHOR SUCCESS) perform comparably on finding hidden objects. However, this trend plateaus for higher success models. (d) Trends are similar when distractor objects are introduced for hidden object navigation.

plore this hypothesis in Appx. E by visualizing CLIP retrieval results on LAION-5B [68] for the uncommon object categories. The relatively high performance on uncommon objects speaks to the flexibility of CoW baselines and their ability to inherit desirable properties from the open-vocabulary models that they are constructed from.

Can CoWs utilize appearance and spatial descriptions? Looking at Fig. 5 (a) we see that neither appearance nor spatial descriptions improve CoW performance compared to their ROBOTHOR baseline performance (i.e., most points lie under the $y = x$ line). However, CoW is able to take better advantage of appearance descriptions than spatial descriptions. These results motivate future investigation on open-vocabulary object localization with a greater focus on textual object attributes.

Can CoWs find visible objects in the presence of distractors? In Fig. 5 (b) we see that CoWs experience performance degradation when compared to Fig. 5 (a). We conclude that appearance and spatial attributes added as language input are not sufficient to deal with the complexity of distractors given current open-vocabulary models.

Can CoWs find hidden objects? Looking at Fig. 5 (c) we notice that models in the lower success regime (less than 15% SUCCESS on ROBOTHOR) are able to find hidden objects at about the same rate as ROBOTHOR objects (i.e., they lie on the $y = x$ line). OWL models in the higher success regime (>15%) do not continue this trend; however, they do achieve higher absolute accuracy as seen in Tab. 1. Dealing with occlusion is a longstanding problem in computer vision, and these results provide a foundation upon which future hidden object navigation work can improve.

Can CoWs find hidden objects in the presence of distractors? Comparing Figs. 5 (c) and (d), we notice similar trends lines, with the best models performing worse with distractors. This suggests that distractors are less of a concern in the case of hidden objects than for visible object targets. In light of the fact that detection methods generally work better on larger objects, we hypothesize this effect is because distractor objects are smaller (e.g., apples, vases, basketballs) than objects used to conceal target categories (e.g., beds, sofas, etc.).

ID	CoW breeds Loc.	Arch.	Exp. Strategy	PASTURE (Avg.) SPL	SR	ROBOTHOR SPL	SR
▲	OWL	B/32	ROBOTHOR learn.	10.2	17.3	13.1	20.9
▲	OWL	B/32	HABITAT learn.	8.6	19.4	9.8	20.4
▲	OWL	B/32	FBE	**12.6**	**21.1**	**16.9**	**26.7**

Table 2. **Exploration ablation.** For a fixed object localizer (OWL-ViT B/32 with post processing), we ablate over different choices of exploration policy: the FBE heuristic, agents trained in ROBOTHOR, and HABITAT (MP3D). We find that FBE outperforms learnable alternatives on both PASTURE and ROBOTHOR. HABITAT learnable model perform worst, but are not trained on any PASTURE or ROBOTHOR data.

ID	CoW breeds Loc.	Arch.	Obj. Prior	PASTURE Uncom. SPL	SR	ROBOTHOR SPL	SR
▲	OWL	B/32	None	20.5	32.8	16.8	26.7
▲	OWL	B/32	GPT-3.5	**22.2**	**36.9**	**17.0**	**27.5**

Table 3. **CoW with GPT-3.5 priors.** Leveraging GPT-3.5, we generate priors for objects (e.g., apples are likely to be in *dining room* scenes). Instead of directly searching for the target object, CoW first searches for the scenes, which boosts performance.

What exploration method performs best? We ablate the decision to use FBE for most experiments by fixing an object localizer (OWL, B/32 (▲)) and comparing against ROBOTHOR learnable exploration and HABITAT learnable exploration in Tab. 2. We notice that FBE performs best in all cases; however, learnable exploration still performs well suggesting that these models do learn useful strategies for the downstream tasks.

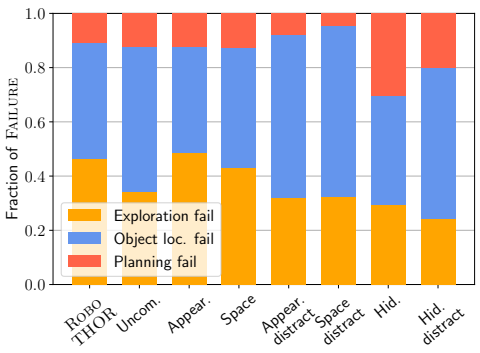

Figure 6. **Failure analysis for OWL, B/32 (▲).** Exploration and object localization errors occur at similar ratios, with increased localization failures in the presence of distractors.

Can CoW incorporate object priors? Examining Tab. 3, we see that incorporating GPT-3.5 object-level priors improves performance on both PASTURE uncommon objects and ROBOTHOR. These initial results suggest positive trends for incorporating outside knowledge into CoW. Future work may consider more sophisticated methods for injecting priors to steer navigation.

How do CoWs fail? We identify three high-level failure modes. (1) *Exploration fail*: the target is never seen. (2) *Object localization fail*: the target is seen but the localizer never fires. (3) *Planning fail*: the target is seen and the localizer fires, but planning fails due to inaccuracy in the map representation (Sec. 4.2). Looking at Fig. 6, we notice a large fraction of failures are due to exploration and object localization. This suggests CoWs may continue to improve as research in these fields progress. In Fig. 6 we also see that in cases where distractors are present a higher fraction of object localization failures occur, further suggesting that open-vocabulary models currently struggle to make full use of attribute prompts. See Appx. I for more failure analysis.

6.3. Comparison to Prior Art

We primarily evaluate CoWs in general L-ZSON settings; however, we further evaluate CoWs on ZSON benchmarks to establish them as a strong baseline for these tasks. Recall, ZSON can be seen as a case of L-ZSON where only object goals are specified (no attributes).

In Tab. 4, we see there exists a CoW that outperforms the end-to-end baselines in all cases except SUCCESS on HABITAT (MP3D). For instance, the CLIP-Grad., B/32 (▲) matches the SemanticNav-ZSON model on HABITAT (MP3D) SPL: 4.9 for CoW v.s. 4.8 for the competitor, while improving over EmbCLIP-ZSON ROBOTHOR by 15.6 percentage points. To contextualize this result, CoWs train for *0* navigation steps, while SemanticNav-ZSON and EmbCLIP-ZSON train in the target evaluation simulators for 500M and 60M steps respectively.

ID	CoW breeds Loc.	Arch.	HABITAT (MP3D) SPL	HABITAT (MP3D) SR	ROBOTHOR (subset) SPL	ROBOTHOR (subset) SR	ROBOTHOR (full) SPL	ROBOTHOR (full) SR	Nav. training steps
▲	CLIP-Grad.	B/32	**4.9**	9.2	15.0	23.7	9.7	15.2	**0**
▲	OWL	B/32	3.7	7.4	**20.8**	**32.5**	**16.9**	**26.7**	**0**
EmbCLIP-ZSON [38]			–	–	–	8.1	–	14.0*	60M
SemanticNav-ZSON [46]			4.8	**15.3**	–	–	–	–	500M

Table 4. **Comparison to prior art on existing ZSON benchmarks.** CoWs are able to match or out-compete existing methods that leverage millions of steps of navigation training in the evaluation simulator. *indicates a result from prior work that includes, non-zero-shot evaluation. Specifically, only 1/4 of the evaluations are zero-shot on ROBOTHOR (subset) and the remaining 3/4 on categories seen during training.

The superior performance of SemanticNav-ZSON in terms of MP3D SUCCESS indicates that there can be benefits to in-domain learning. Future work may consider unifying the benefits of CoW-like models and fine-tuned models.

7. Limitations and Conclusion

Limitations. While our evaluation of CoWs on HABITAT, ROBOTHOR, and PASTURE is a step towards assessing their performance in different domains, ultimately, real-world performance matters most. Hence, the biggest limitation of our work is the lack of large-scale, real-world benchmarking—which is also missing in much of the related literature. Additionally, CoW inherents the meta-limitations of the object localization and exploration methods considered. For example, object localizers require tuning a confidence threshold to balance precision and recall. Finally, we do not consider different agent embodiment or continuous action spaces. This is a pertinent investigation given recent findings of Pratt *et al*. [59] that agent morphology can be a big determinant of downstream performance.

Conclusion. This paper introduces the PASTURE benchmark for language-driven zero-shot object navigation and several CLIP on Wheels baselines, translating the successes of existing zero-shot models to an embodied task. We view CoW as an instance of using open-vocabulary models, with text-based interfaces, to tackle robotics tasks in more flexible settings. We hope that the baselines and the proposed benchmark will spur the field to explore broader and more powerful forms of zero-shot embodied AI.

Acknowledgement. We would like to thank Jessie Chapman, Cheng Chi, Huy Ha, Zeyi Liu, Sachit Menon, and Sarah Pratt for valuable feedback. This work was supported in part by NSF CMMI-2037101, NSF IIS-2132519, and an Amazon Research Award. SYG is supported by a NSF Graduate Research Fellowship. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

References

[1] Ziad Al-Halah, Santhosh K. Ramakrishnan, and Kristen Grauman. Zero experience required: Plug & play modular transfer learning for semantic visual navigation. *arXiv*, 2022. 2

[2] 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. *CVPR*, 2018. 2

[3] Donghyeon Baek, Youngmin Oh, and Bumsub Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. *ICCV*, 2021. 1

[4] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. *ECCV*, 2018. 1

[5] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv*, 2020. 1, 3, 5

[6] Maxime Bucher, Tuan-Hung VU, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *NeurIPS*, 2019. 1

[7] Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. *ICLR*, 2018. 2

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ECCV*, 2020. 4, 14

[9] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. *NeurIPS*, 2020. 2

[10] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *NeurIPS*, 2020. 2

[11] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Kumar Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *ICLR*, 2020. 2

[12] Prithvijit Chattopadhyay, Judy Hoffman, Roozbeh Mottaghi, and Aniruddha Kembhavi. Robustnav: Towards benchmarking robustness in embodied navigation. *ICCV*, 2021. 2

[13] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. *CVPR*, 2021. 4, 13, 14

[14] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. *ICLR*, 2019. 2

[15] Jiaxin Cheng, Soumyaroop Nandi, Prem Natarajan, and Wael Abd-Almageed. Sign: Spatial-information incorporated generative network for generalized zero-shot semantic segmentation. *ICCV*, 2021. 1

[16] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *SSST@EMNLP*, 2014. 3

[17] Andrew J Davison and David W Murray. Mobile robot localisation using active vision. *ECCV*, 1998. 2

[18] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. Robothor: An open simulation-to-real embodied ai platform. *CVPR*, 2020. 1, 3, 5

[19] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *NeurIPS*, 2022. 2, 6

[20] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. *ICRA*, 1999. 2

[21] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. Zero-shot object detection by hybrid region embedding. *BMVC*, 2018. 1

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009. 1

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, 2020. 13

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020. 4, 6

[25] Samir Yitzhak Gadre, Kiana Ehsani, Shuran Song, and Roozbeh Mottaghi. Continuous scene representations for embodied ai. *CVPR*, 2022. 2, 3

[26] Daniel Gordon, Abhishek Kadian, Devi Parikh, Judy Hoffman, and Dhruv Batra. Splitnet: Sim2sim and task2task transfer for embodied visual navigation. *CVPR*, 2019. 2

[27] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Zero-shot detection via vision and language knowledge distillation. *arXiv*, 2021. 1

[28] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. Lvis: A dataset for large vocabulary instance segmentation. *CVPR*, 2019. 1

[29] Abhinav Kumar Gupta, Adithyavairavan Murali, Dhiraj Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. *NeurIPS*, 2018. 5

[30] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. *CVPR*, pages 2616–2625, 2017. 2

[31] Meera Hahn, Devendra Singh Chaplot, and Shubham Tulsiani. No rl, no simulation: Learning to navigate without navigating. *NeurIPS*, 2021. 2

[32] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. *Experimental robotics*, 2014. 2

[33] Ping Hu, Stan Sclaroff, and Kate Saenko. Uncertainty-aware learning for zero-shot semantic segmentation. *NeurIPS*, 2020. 1

[34] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*, 2022. 2

[35] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *ICML*, 2021. 1, 13

[36] Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdetr - modulated detection for end-to-end multi-modal understanding. *ICCV*, 2021. 1, 4, 13, 14

[37] Naoki Kato, Toshihiko Yamasaki, and Kiyoharu Aizawa. Zero-shot semantic segmentation via variational mapping. *ICCV-W*, 2019. 1

[38] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. *arXiv*, 2021. 1, 2, 3, 6, 8

[39] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020. 2

[40] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020. 2

[41] Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and autonomous systems*, 1991. 2

[42] Xinghang Li, Di Guo, Huaping Liu, and Fuchun Sun. Revece: Remote embodied visual referring expression in continuous environment. *RA-L*, 2022. 2

[43] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil S. Kanhere, and Huaxiang Zhang. Zero-shot object detection with textual descriptions. *AAAI*, 2019. 1

[44] Yiqing Liang, Boyuan Chen, and Shuran Song. Sscnav: Confidence-aware semantic scene completion for visual semantic navigation. *ICRA*, 2021. 2

[45] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. *ECCV*, 2014. 1

[46] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *arXiv preprint arXiv:2206.12403*, 2022. 1, 2, 3, 6, 8

[47] Qiaomei Mao, Chong Wang, Sheng Yu, Ye Zheng, and Yuqi Li. Zero-shot object detection with attributes-based category similarity. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2020. 1

[48] Lina Mezghani, Sainbayar Sukhbaatar, Thibaut Lavril, Oleksandr Maksymets, Dhruv Batra, Piotr Bojanowski, and Alahari Karteek. Memory-augmented reinforcement learning for image-goal navigation. *arXiv*, 2021. 2

[49] Matthias Minderer, Alexey A. Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. *ECCV*, 2022. 1, 4, 6, 13, 14

[50] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017. 13

[51] Hans Moravec and Alberto Elfes. High resolution maps from wide angle sonar. *ICRA*, 1985. 2

[52] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. *ISMAR*, 2011. 2

[53] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *RA-L*, 2019. 2

[54] Clark F Olson and Larry H Matthies. Maximum likelihood rover localization by matching range maps. *ICRA*, 1998. 2

[55] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022. 4, 15

[56] Simone Parisi, Victoria Dean, Deepak Pathak, and Abhinav Kumar Gupta. Interesting object, curious agent: Learning task-agnostic exploration. *NeurIPS*, 2021. 2

[57] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. *ICML*, 2017. 2

[58] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Hanxiao Liu, Adams Wei Yu, Minh-Thang Luong, Mingxing Tan, and Quoc V. Le. Combined scaling for zero-shot transfer learning. *arXiv*, 2021. 1, 13

[59] Sarah Pratt, Luca Weihs, and Ali Farhadi. The introspective agent: Interdependence of strategy, physiology, and sensing for embodied agents. *arXiv*, 2022. 8

[60] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. *CVPR*, 2020. 2

[61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *ICML*, 2021. 1, 2, 4, 6, 13

[62] Shafin Rahman, Salman Hameed Khan, and Fatih Murat Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. *ACCV*, 2018. 1

[63] Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. *ICLR*, 2020. 2

[64] Santhosh K. Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. *CVPR*, 2022. 2

[65] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 1

[66] Manolis Savva, Angel X Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. Minos: Multimodal indoor simulator for navigation in complex environments. *arXiv*, 2017. 2

[67] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. *ICCV*, 2019. 1, 3, 5

[68] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 2022. 7, 16, 18

[69] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv*, 2017. 3

[70] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *IJCV*, 2019. 4, 13, 14

[71] 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. *CVPR*, 2020. 2

[72] Shuran Song, Linguang Zhang, and Jianxiong Xiao. Robot in a room: Toward perfect object recognition in closed environments. *arXiv*, 2015. 2

[73] Alexander L. Strehl and Michael L. Littman. An analysis of model-based interval estimation for markov decision processes. *J. Comput. Syst. Sci.*, 2008. 2, 3

[74] Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. Multion: Benchmarking semantic map memory using multi-object navigation. *NeurIPS*, 2020. 2

[75] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. *CVPR*, 2021. 2, 3

[76] Luca Weihs, Jordi Salvador, Klemen Kotar, Unnat Jain, Kuo-Hao Zeng, Roozbeh Mottaghi, and Aniruddha Kembhavi. Allenact: A framework for embodied ai research. *arXiv*, 2020. 3, 12

[77] Erik Wijmans, Abhishek Kadian, Ari S. Morcos, Stefan Lee, Irfan Essa, D. Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *ICLR*, 2019. 2, 3

[78] Brian H Wilcox. Robotic vehicles for planetary exploration. *Applied Intelligence*, 1992. 2

[79] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. *CVPR*, 2019. 1, 2

[80] Chenyun Wu, Zhe Lin, Scott D. Cohen, Trung Bui, and Subhransu Maji. Phrasecut: Language-based image segmentation in the wild. *CVPR*, 2020. 4, 14

[81] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. *CVPR*, 2018. 2

[82] Brian Yamauchi. A frontier-based approach for autonomous exploration. *CIRA*, 1997. 3

[83] Claudia Yan, Dipendra Misra, Andrew Bennnett, Aaron Walsman, Yonatan Bisk, and Yoav Artzi. Chalet: Cornell house agent learning environment. *arXiv*, 2018. 2

[84] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv*, 2018. 2

[85] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. *CVPR*, 2022. 14

[86] Chong Zhou, Chen Change Loy, and Bo Dai. Denseclip: Extract free dense labels from clip. *arXiv*, 2021. 1

[87] Fengda Zhu, Xiwen Liang, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. *CVPR*, 2021. 2

[88] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Zero shot detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020. 1

[89] 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. *ICRA*, 2017. 1, 2