SPOC: Imitating *Shortest Paths* in Simulation Enables Effective Navigation and Manipulation in the Real World

Kiana Ehsani[†] Tanmay Gupta[†] Rose Hendrix[†] Jordi Salvador[†] Luca Weihs[†] Kuo-Hao Zeng[†]
Kunal Pratap Singh[‡] Yejin Kim[†] Winson Han[†] Alvaro Herrasti[†]
Ranjay Krishna[†ψ] Dustin Schwenk[†] Eli VanderBilt[†] Aniruddha Kembhavi[†ψ]

[†]Allen Institute for AI [ψ]University of Washington [‡]EPFL

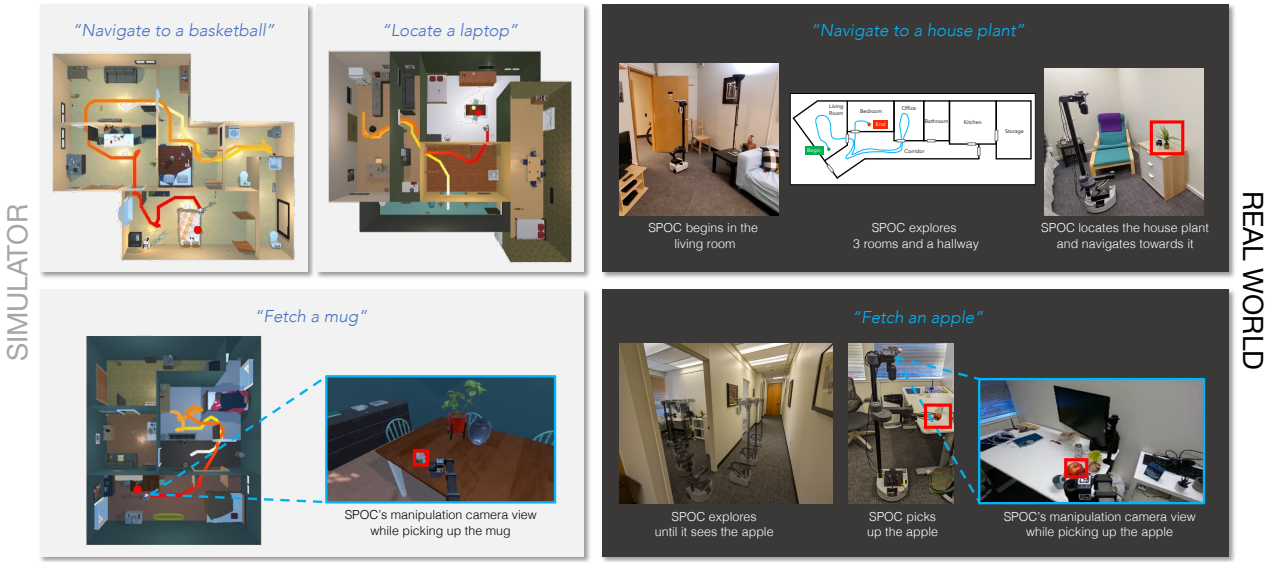

Figure 1. We present SPOC, an embodied navigation and manipulation agent trained by imitating shortest-path experts in simulation. Visualized paths in simulation are white in the beginning and red at the end; red circles and squares highlight target object locations only for illustration. *Top-Left:* A variant of SPOC, **trained only on shortest path episodes for object goal navigation** demonstrates complex behavior like exploration, obstacle avoidance, and back-tracking in novel environments; *Top-Right*: The same agent transfers to the real world with no further adaptation and navigates to the house plant; *Bottom-Left:* SPOC exploring the house to navigate to the mug and then picking it up; *Bottom-Right:* SPOC navigating and picking up objects in the real world, again with no change in weights.

Abstract

Reinforcement learning (RL) with dense rewards and imitation learning (IL) with human-generated trajectories are the most widely used approaches for training modern embodied agents. RL requires extensive reward shaping and auxiliary losses and is often too slow and ineffective for long-horizon tasks. While IL with human supervision is effective, collecting human trajectories at scale is extremely expensive. In this work, we show that imitating shortest-path planners in simulation produces agents that, given a language instruction, can proficiently navigate, explore, and manipulate objects in both simulation and in the real world using only RGB sensors (no depth map or GPS coordinates). This surprising result is enabled by our end-to-end, transformer-based, SPOC *architecture, powerful visual encoders paired with extensive image augmentation, and the dramatic scale and diversity of our training data: millions of frames of shortest-path-expert trajectories collected inside approximately 200,000 procedurally generated houses containing 40,000 unique 3D assets. Our models, data, training code, and newly proposed 10-task benchmarking suite* CHORES *are available in* spoc-robot.github.io.

1. Introduction

The prevalent method to build embodied agents employs a rich set of sensors such as RGB, depth maps, and GPS co-

ordinates that feed into modular architectures with mapping components [7–9, 27], off-the-shelf computer vision models [38, 46, 79], and even large language models (LLM) that can be used for planning [32, 34, 44, 66, 70, 71]. These agents are either trained with on-policy reinforcement learning (RL) using careful reward shaping and auxiliary losses [28, 29, 65, 84], which tends to be slow and ineffective, or trained with imitation learning (IL) on large corpora of human demonstrations, which is incredibly expensive. The transfer gap from simulators to real is often mitigated by optimizing the photorealism of the simulator or via sim-to-real image translation models. Finally, at test time, it is common practice if navigation is required to provide the locations of objects or assume a map of the environment is known [4, 34, 83].

In contrast to the above, in this paper, we surprisingly find that *imitating shortest path experts in simulation can produce embodied agents effective at navigation, exploration, and manipulation, in both simulation and the real world*. Our model, SPOC (**S**hortest **P**ath **O**racle **C**lone), uses (a) only RGB observations with no depth and no GPS sensors, (b) a transformer-based architecture with no mapping module and no LLM, and (c) imitation learning on heuristic *shortest path* planners with no human demonstrations and no RL. SPOC is trained in the AI2-THOR [39] simulator transfers well into the real world with no adaptation or fine-tuning. This all without making any assumptions about scene layout or object appearance at test time.

Notably and unexpectedly SPOC, when trained to imitate a *shortest path* expert for the singular task of object-goal navigation, demonstrates the capability to explore its environment comprehensively, peek into rooms, and backtrack along its path as it searches for its target, despite never having seen this behavior in its training data. Fig. 1 visualizes paths that showcase the exploration capabilities of SPOC. We hypothesize that SPOC's ability to be an effective explorer is less hampered by the use of *shortest path* training data and instead seems gated by its object perception. Indeed, perception errors, not exploration failures, appear to be the primary cause of failures: employing ground truth target object detection alongside raw RGB results in a very high success rate of 85% for OBJNAV.

SPOC is a very effective multi-tasking agent. When trained jointly on four tasks – Object-Goal Navigation (OBJNAV), Room Visitation (ROOMVISIT), PickUp Object (PICKUP), and Fetch Object (FETCH) – SPOC achieves an impressive average success rate of 49.9% in unseen simulation environments at test time. These numbers easily outperform carefully tuned agents trained with reinforcement learning by a large margin of close to 30 points across the task suite. The high success rates achieved by SPOC translate to the real world where it achieves 56% success across these tasks. We further train SPOC to follow open vocabulary instructions via a suite of seven navigation tasks where it achieves a high success rate of 51%. This benchmark is designed to evaluate a wide range of capabilities such as recognizing objects, discerning affordances, identifying scene elements, recognizing relative attributes of objects (e.g. *bigger*, *lower*), and understanding local references (e.g. *near*, *on*).

We identify four key factors that enable effective imitation learning from heuristic experts. First, diversity of simulated worlds plays an important role – we used almost 3 orders of magnitude more unique houses to train with IL than past work [58, 59, 75, 79]. Second, using powerful visual encoders is critical – we moved from the defacto ResNet-50 CLIP encoder [57] employed in the literature [38] to DINOV2 [50] and SIGLIP [85] and found huge gains. Third, moving to transformer architectures with long context windows of up to 100 frames outperforms previously employed recurrent architectures [73]. Finally, scaling up the size of the training data matters.

This work shows the promise of imitating heuristic experts in simulators as a means to develop capable robots for the real world. Our experiments show that scale and diversity play an important role in enabling this behavior, and we posit that further scaling up this paradigm has huge merits and can lead to large improvements on challenging tasks.

2. Related Work

Simulators, tasks, and benchmarks. Rapid progress in Embodied AI has led to an explosion of simulators, tasks, and benchmarks. Early simulators were built for navigation, often using 3D scans of the real world and supported only basic, if any, object interaction [1, 16, 39, 53, 61, 77]. Recent simulators model realistic robotic agents but often trade off physical fidelity to increase simulation speed [21, 22, 24, 41, 42, 54, 63, 68, 78]. We use the AI2-THOR environment [39] and ProcTHOR [18] to produce unbounded numbers of procedurally generated households which support object manipulation with a Stretch RE-1 [37].

Many embodied AI benchmarks focus on navigation [1, 2, 10, 11, 16, 40, 56, 76, 89]. Beyond navigation, many tasks (*e.g.*, ALFRED [64], Visual Room Rearrangement [74], ARMPOINTNAV / OBJDIS [21, 22], BEHAVIOR [42, 67], and others [25, 26]) require the agent to interact directly with objects in the environment at various levels of abstraction. Most similar to the tasks used in this work is *Open-Vocabulary Mobile Manipulation* (OVMM) [82] task in which an RE-2 Stretch robotic agent must transport an object of a given type from an initial receptacle of a given type to a goal receptacle of a different type. While OVMM focuses on the core task of mobile manipulation, we benchmark our agent across a wide variety of tasks related to navigation, manipulation, and language understanding.

Embodied architectures and training methods. A com-

mon architecture for Embodied AI is based on an observation encoder implemented by means of a CNN, and an RNN providing episode memory and producing the hidden state upon which to condition the policy at each step. For the visual backbone, common choices include *ResNets* [30] or CLIP [38, 57], or trained from scratch [76]. [45] have comprehensively studied the impact of visual backbones on embodied performance.

Embodied agents are frequently trained with on-policy actor-critic RL methods, *e.g.* A3C [49], A2C [62], or DD-PPO [76]. Auxiliary losses co-trained with these RL have also been proposed to improve sample efficiency [65, 81]. Transformer-based architectures have been proposed in combination with IL and BC bypassing the need to use RL for training. Decision transformer [12] and trajectory transformer [35] cast the sequential decision problem as a sequence modeling problem, thus enabling BC to replace RL. *Gato* [60] trains an autoregressive transformer across tasks and embodiments, towards extended generalization.

Imitation learning has recently gained popularity in robotics, significantly impacting areas like autonomous driving [14, 36, 43, 48, 52, 55]. With the effectiveness of in-context learning, using LLMs as robotic planners has become increasingly prominent [15, 33, 34]. RT-1 [4] scales up model capacity and multi-task data focusing on manipulation. RT-2 [3] encodes actions as text tokens to enable large joint VL and decision-taking pretraining for further generalization improvements in manipulation. *DualMind* [72] is trained with self-supervised learning on state-action interactions and imitation learning with prompts. The main bottleneck of these approaches is the tremendously costly human generated trajectories, frequently in the real-world. Importantly, these works usually do not train navigation and manipulation jointly, frequently assuming ground-truth navigation knowledge or that the navigation can be solved by SLAM-based systems. We train an end-to-end system using cheap, shortest-path-planner generated data, and show its effectiveness in the real-world.

Sim-to-Real. One way to reduce the sim-to-real gap is to train in high-fidelity simulators. However, accurately modeling the real world, including camera miscalibration, or out-of-distribution lighting changes, is hard. An alternative is domain randomization [13, 69], where camera poses, lighting, textures, or visual degradations can be randomly sampled during training time, thus making the trained agent resilient to such fluctuations. Another type of augmentation is *Phone2Proc* [19], where a scanned layout of the real-world house is used to generate many simulated variations for agent fine-tuning. Finally, domain transfer as in CycleGAN [88] allows adapting visual appearances during training (sim-to-real) or inference (real-to-sim). RetinaGAN [31] additionally enforces object-detection consistency and is employed in [34].

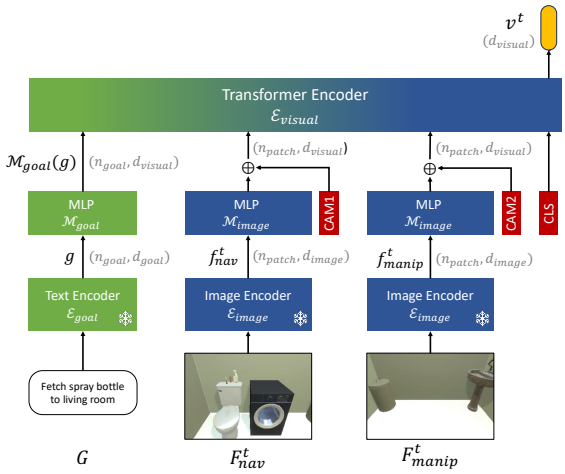

Figure 2. **Goal-conditioned Visual Encoder** for extracting goal-relevant visual information from the two cameras.

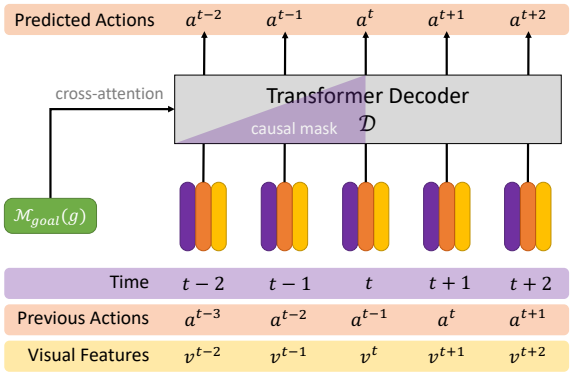

Figure 3. **Action Decoder** for predicting action at the current time step given the goal, current and past observations, and past actions.

3. Imitation in Procedural Houses

Embodied agents are commonly trained in simulation using on-policy RL. The agent's actions result in new observations and rewards from the environment, and the policy is updated using the rewards that guide the agent towards desirable behaviors. Practically, RL in complex visual worlds is sample inefficient, especially when using large action spaces and for long-horizon tasks. Such training is bottlenecked by the simulator's speed; the more physically and visually realistic the simulation, the lower the frame-rate. Finally, RL training, even relatively simple tasks such as navigating to an object or picking it up, requires careful reward shaping, auxiliary losses, and modular architectures.

IL is a compelling alternative to RL since learning from expert trajectories can be cast as a supervised learning problem. However, IL's big success stories have required a lot of data. Data for IL has traditionally been collected from two sources: (1) expert humans [4, 58] and (2) heuristic planners operating from ground-truth information not available

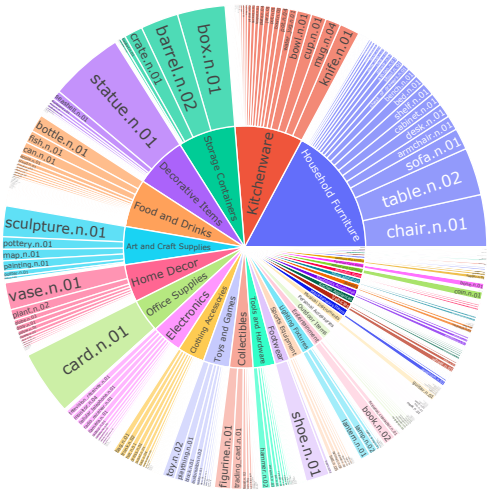

Figure 4. **Diversity of assets** in training environments.

during inference time (*e.g.* shortest paths computed using navigation meshes in simulation). While human-collected data is the gold standard, it is extremely expensive. Planner-based approaches are cheap but have been found in prior work to result in suboptimal learning; for instance, [58] found that navigation agents trained to follow shortest paths achieved success rates of only 4.4% on the ObjectNav task on the MP3D [6] validation dataset versus ≈35% success rates when trained to imitate human trajectories. There is also a healthy skepticism for the generalization ability of shortest-path trained IL agents in novel environments where the agent needs to balance exploration and exploitation to achieve a goal while simultaneously building an implicit map of the environment. Moreover, [75] mathematically proves that learning such sub-optimal behavior is guaranteed in some settings due to an "imitation gap".

In the following sections, we show that using transformer architectures with long context windows to imitate heuristic planners at scale can help unlock the power of IL and produce effective agents in simulation and the real world. Sec. 4 details our agent, SPOC. In Sec. 5 we outline our large-scale data collection, made possible by recent breakthroughs in procedurally generating home environments [18], access to Objaverse 3D assets [17], and efficient heuristic planners that leverage rich ground truth information in the AI2-THOR simulator. We present a new benchmark, CHORES in Sec. 6 and a comprehensive analysis of SPOC in Sec. 7.

4. The Shortest Path Oracle Clone (SPOC)

We present SPOC🖐, an agent embodied in the Stretch RE-1 [37] robot, trained in simulation to follow text instructions and complete long-horizon navigation and manipulation tasks. SPOC takes as input the text instruction and visual observation at each time step t and predicts an action a^t. The Stretch Robot's axis of navigation is perpendicular to the axis of manipulation. This necessitates two RGB cameras, one pointing in the direction of navigation and the other pointing at the arm. We discretize the action space into 20 actions: Move Base (± 20 cm), Rotate Base ($\pm 6°$, $\pm 30°$), Move Arm (x, z) (± 2 cm, ± 10 cm), Rotate Grasper ($\pm 10°$), pickup, dropoff, done with subtask, and terminate.

Our model consists of three main components: (1) a textual goal encoder, which processes open vocabulary language instructions; (2) an instruction-conditioned visual encoder for encoding visual inputs at each time step; and (3) a high-capacity causal transformer action decoder that predicts the action for the current time step given the goal, the current and previous visual inputs, and previous actions.

Goal Encoder. We use a pretrained text encoder $\mathcal{E}_{\text{goal}}$ that maps the goal specification $\mathcal{G}$ into a sequence of contextualized token representations $\mathbf{g} = \mathcal{E}_{\text{goal}}(\mathcal{G}) \in \mathbb{R}^{n_{\text{goal}} \times d_{\text{goal}}}$ where n_{goal} and d_{goal} are the number of sub-word tokens in the goal text and dimension of the token representation, respectively. We experiment with T5 and SIGLIP text encoders.

Goal-Conditioned Visual Encoder. SPOC accepts visual inputs from two RGB cameras pointing in perpendicular directions. At any time step t, the goal-conditioned visual encoder $\mathcal{E}_{\text{visual}}$ extracts and integrates visual information from the two RGB frames $\mathcal{F}^t_{\text{nav}}$ and $\mathcal{F}^t_{\text{manip}} \in \mathbb{R}^{h \times w \times 3}$ and represent it as a single vector $\mathbf{v}^t = \mathcal{E}_{\text{visual}}(\mathcal{F}^t_{\text{nav}}, \mathcal{F}^t_{\text{manip}}, \mathcal{G})$. We use a Transformer encoder to achieve this, shown in Fig. 2. The frames are encoded into sequences of contextualized patch embeddings $\mathbf{f}^t_{\text{nav}}$ and $\mathbf{f}^t_{\text{manip}} \in \mathbb{R}^{n_{\text{patch}} \times d_{\text{image}}}$ using a pretrained image encoder $\mathcal{E}_{\text{image}}$ where n_{patch} and d_{image} are the number of image patches and output feature dimension of the image encoder. These features are mapped to the transformer input dimension d_{visual} using an MLP $\mathcal{M}_{\text{image}}$ with ReLU and LayerNorm. The goal representation $\mathbf{g}$ is also mapped to the d_{visual} dimension using another MLP $\mathcal{M}_{\text{goal}}$. Next, we add learnable camera-type embeddings to differentiate features from the two cameras. Finally, we concatenate the patch features, goal features, and a learnable [CLS] token embedding along the patch dimension and input this $(2n_{\text{patch}} + n_{\text{goal}} + 1) \times d_{\text{visual}}$ tensor through the transformer encoder. The output at the position of the [CLS] token serves as the goal-conditioned visual representation $\mathbf{v}^t$.

Action Decoder. We use an autoregressive Transformer decoder $\mathcal{D}$ with causal masking to predict actions, see Fig. 3. The input to the decoder is the sequence of previous and current visual representations $\{\mathbf{v}^0, \cdots, \mathbf{v}^t\}$ additively combined with sinusoidal temporal position encodings and learned previous time step action embeddings. The decoder conditions on goal encoding $\mathcal{M}_{\text{goal}}(\mathbf{g})$ using cross-attention. At each time step, the output embedding from the transformer decoder is fed through linear and softmax layers to predict an action distribution for that time step $\pi^t = \text{Softmax}(\text{Linear}(\mathcal{D}(\mathbf{v}^{0:t}, a^{0:t-1}; \mathcal{M}_{\text{goal}}(\mathbf{g}))[t]))$. Causal masking during training ensures the decoder only

Task	Description & Example
ObjNav	Locate an object category: "find a mug"
PickUp	Pick up a specified object in agent line of sight: "pick up a mug"
Fetch	Find and pick up an object: "locate a mug and pick up that mug"
RoomVisit	Traverse the house. "Visit every room in this 5-room house. Indicate when you have seen a new room and when you are done."

Table 1. Chores tasks.

Task	Target Description & Example
ObjNav	Object's category: "vase"
ObjNavAfford	Object's possible uses: "a container that can best be used for holding fresh flowers decoratively"
ObjNavLocalRef	Object's nearby objects: "a vase near a tennis racket and a basketball"
ObjNavRelAttr	Object category comparative attribute: "the smallest vase in the bedroom"
ObjNavRoom	Object's room type: "vase in the living room"
ObjNavDesc	Open vocab instance description: "the brown vase painted orange with a bird on the side"
RoomNav	Type of room: "bedroom"

Table 2. ChoresNav tasks. The full task specification also includes a navigation verb, such as "Search for a vase".

attends over current and past inputs to predict the current action. The model is optimized using the cross-entropy loss in a teacher-forcing manner, *i.e*. we minimize the cross entropy between π^t and the one-hot encoding of the expert action for the current timestep. During inference, the agent acts in the environment at time t by sampling an action a^t from π^t; a^t is then fed as an input to the model on the following timestep. For compute-efficient mini-batch training, we train with a limited temporal context window (*e.g*. 100), but the model uses all past observations during inference. To enable using a larger context window during deployment, we randomly shift the time indices fed to the agent during training; in particular, if we sample temporal context window $[s, s+99]$ from an expert trajectory during training, then we input the corresponding actions and visual features to the model as-is, but pair them with the shifted time indices $[s+\ell, s+99+\ell]$ where $\ell \sim \text{Unif}\{0, \ldots, 900\}$.

5. Procedural Data

We now describe our large-scale dataset of diverse household environments and the planners we use to generate expert trajectories within these environments.

5.1. Environments

To overcome the challenges of scale and diversity, we leverage recent advances in the AI2-THOR simulated environment [39] which allow for importing any of the 800k 3D assets from the Objaverse dataset [17] into AI2-THOR scenes. Of these 800k assets, we use a subset of ≈40k objects that have received additional annotations certifying their relevance to household environments and providing additional metadata (*e.g*. object types grounded in the WordNet 2022 hierarchy [47], instance descriptions, and size in meters). When paired with object instances already existing in AI2-THOR (≈2k instances), we are left with 41,133 unique 3D assets corresponding to 863 unique object types (henceforth synonymous with Wordnet synsets). The composition of the resulting collection of assets, in terms of their assigned synsets, is illustrated in Fig. 4.[1]

However, importing many new 3D assets into the 120 AI2-THOR scenes is insufficient for required diversity in scene layouts and may not result in meaningful object configurations. Instead, we use ProcTHOR [18], a procedural house generation framework built within AI2-THOR which can, in principle, generate an unbounded number of unique houses. We use ProcTHOR to generate a total of ≈200k houses (with between 1 and 8 rooms each), all containing Objaverse assets. Assets are partitioned into train and eval instances, resulting in some object categories evaluating zero-shot performance due to the long tail of small-instance-count categories. For more details see the supp.

5.2. Expert Trajectories

In order to produce our expert trajectories for imitation learning, we need planners capable of a range of skills essential for navigating and manipulating within complex multi-room settings. These planners must recognize and interact with objects, navigate through cluttered environments, and adapt to various obstacles. Below, we offer a high-level overview of our planners. We are able to write these planners because of the wealth of ground truth information available within the AI2-THOR environment (*e.g*. 3D coordinates of all objects); we stress that this ground-truth information is *not* available to the agent at inference time and is simply used to produce the expert trajectories used for training. For further information, please refer to the supplementary materials.

Navigation. Given a target object or GPS coordinate within an environment, navigate to that target by following a shortest path computed via a navigation mesh. If the target was an object, rotate so that the object is approximately centered in the agent's navigation (or manipulation) camera. As our agent takes discrete actions (*e.g*. "move ahead by 0.2m"), the shortest path is followed approximately.

Manipulation. Given a target object instance, the agent first uses the privileged information from the simulation to navigate to a location from which the object is reachable by the arm. Then, as the poses of the object and the agent are known, we use iterative distance minimization to bring the arm close to the target object and then grasp that object.

Room Visitation. For our RoomVisit task, the agent must visit every room in the house and issue sub-task completion signal. Since the layout of the house is known during trajectory generation in simulation, we can obtain the center of

[1] For this visualization, synsets have been further classified by GPT-3.5 [5, 51] as belonging to one of 36 possible semantic clusters, in turn selected based on the whole collection of synsets.

the rooms. For this task, we define a shortest-path planner that navigates to each room center via depth-first-search.

6. Benchmark

We evaluate SPOC on a new benchmark, CHORES (**C**ore **HO**usehold **R**obot **E**valuation**S**). CHORES consists of 4 task types (Tab. 1) and is designed to evaluate how well the model can handle multiple tasks at once, which require skills including navigation, object recognition, object manipulation, and environment exploration.

We also evaluate models on CHORESNAV (Tab. 2), an extension of our benchmark which assesses the agent's ability to interpret and follow object navigation instructions that specify target objects in different ways. In addition to evaluating navigation and object recognition capabilities, CHORESNAV evaluates open vocabulary instruction following, object affordance understanding, scene understanding (*e.g.*, "on top of", "in the kitchen"), and relative object-attributes comparison (*e.g.* "largest container"). For tasks like OBJNAVRELATTR where comparison is needed, each environment has at least one other object of the same type that doesn't meet the condition. For instance, if the task is to find the "smallest bowl", there will be at least two bowls of different sizes in the same room. OBJNAVAFFORD, OBJNAVRELATTR, and OBJNAVLOCALREF tasks may also sample WordNet hypernyms of scene synsets, *e.g.* "container" that would be satisfied by "vase" and "mug" or "sports equipment" for "basketball".

To analyze models we first present results on a subset of 15 object categories from the full 863 categories, called $\mathbb{S}$. Evaluations on the full category set are named $\mathbb{L}$. Our training data contains an average of 90k episodes per task and on average each task contains 195 episodes in the evaluation benchmark. Please see supplementary material.

7. Experiments

Implementation details. SPOC uses SIGLIP image and text encoders. We use 3-layer transformer encoder and decoder and a context window of 100. All models are trained with batch size=224, AdamW and LR=0.0002. Single-task models and multi-task models are trained for 20k and 50k iterations, respectively. Using 16-bit mixed precision training SPOC trains at an FPS of ≈3500, compared to an FPS of ≈175 for RL implemented using AllenAct [73]. We find that data augmentation both during training and testing is critical for model performance, both in simulation and real. In simulation, the PickUp action succeeds if the object is within 6cm of the gripper. In the real world, we leverage a heuristic object grasping model which is called when SPOC invokes PickUp. See supplementary for more details.

7.1. Quantitative Analysis

We compare single and multitask versions of SPOC against single-task RL baselines based on EmbCLIP [38] on CHORES-$\mathbb{S}$ and SPOC's ability to handle large object vocabulary on CHORES-$\mathbb{L}$ (Tab. 3). We thoroughly investigate design decisions like architecture choices (Tab. 4), image encoders (Tab. 5), context window size (Tab. 6), choice of experts (Tab. 7c), and demonstrate the importance of scale (Tab. 7a) and diversity of environments (Tab. 7b). To assess the instruction following capabilities of SPOC, we evaluate on CHORESNAV (Tab. 8). We report Success rate, Episode-length weighted Success (SEL[2] [20]), and percentage rooms visited (%Rooms) for each task and the average Success across all tasks. We now discuss our findings.

IL on shortest-path episodes at scale produces highly capable agents. Comparing rows 1 and 2 of Tab. 3, we see that our IL-trained SPOC dramatically outperforms the popular RL-trained EmbCLIP architecture [38] across all CHORES tasks. Note that the RL baselines were upgraded to use the SIGLIP visual backbone to be comparable to SPOC, required extensive reward shaping, were run on the same hardware as SPOC, but for 2x the number of hours. Indeed, despite extensive efforts, we were unable to obtain non-zero performance on the FETCH task using RL.

SPOC can multitask. Comparing Tab. 3 rows 2-3, CHORES-$\mathbb{S}$ multitask IL performance (49.9%) matches single-task IL (50%). This suggests an absence of performance degradation due to task-competition traditionally seen during multitask training.

Detection brings huge gains. Our RGB-only SPOC agent learns to navigate well, and an error analysis reveals that the majority of failures arise from perception problems. As seen in Tab. 3, compare rows 3-4 and 5-6, SPOC with ground truth detection (provided by the simulator) shows 15% absolute average success rate gain across all tasks on CHORES-$\mathbb{S}$ and an even larger 24.5% absolute gain on CHORES-$\mathbb{L}$ where the detection problem is harder due to the larger object vocabulary. The gains are more prominent for OBJNAV, which obtains an impressive 85%, and FETCH which both require localizing the target object. Tasks like PICKUP (where the agent begins facing the object) and ROOMVISIT (no target object) show little gains as expected. These results indicate that IL trained agents can continue to improve significantly with better object perception.

Transformers provide gains at encoding and decoding. In Tab. 4, we compare SPOC with other commonly used architectural choices for Embodied agents. First, we adapted EmbCLIP's goal-conditioned visual encoder [38] to input 2 RGB frames and upgraded its image encoder to SigLIP to create a non-Transformer visual encoder (nonTxEnc). Re-

[2]We report SEL instead of the prevalent SPL metric due to the known limitations of SPL, see [20], and because weighting by episode length is more informative than path length for tasks that include manipulation.

Benchmark	Model	Training	ObjNav			PickUp			Fetch			RoomVisit			Avg
			Success	SEL	%Rooms	Success	SEL	%Rooms	Success	SEL	%Rooms	Success	SEL	%Rooms	Success
Chores-S	EmbSigLIP* [38]	Single-task RL	36.5	24.5	42.2	71.9	52.9	30.3	0.0	0.0	50.5	16.5	11.9	44.6	31.2
	Spoc-1-task	Single-task IL	57.0	46.2	51.5	84.2	81.0	30.3	15.1	12.6	48.1	43.7	40.4	81.2	50.0
	Spoc	Multi-task IL	55.0	42.2	56.3	90.1	86.9	30.3	14.0	10.5	49.3	40.5	35.7	81.1	49.9
	Spoc w/ GT Det	Multi-task IL	85.0	61.4	58.7	91.2	87.9	30.3	47.3	35.6	61.6	36.7	33.7	79.3	65.0
Chores-L	Spoc	Multi-task IL	33.7	25.1	53.7	75.1	69.1	31.5	10.6	8.1	42.9	35.0	33.2	77.8	38.6
	Spoc w/ GT Det	Multi-task IL	83.9	58.0	64.0	78.0	75.7	31.5	48.6	38.3	60.0	42.0	39.1	83.1	63.1

Table 3. **Training on single tasks, IL outperforms RL even with meticulous reward shaping.** EmbSigLIP refers to using the EmbCLIP [38] model with an upgrade to use the SigLIP backbone since that hugely outperforms the ResNet-50 CLIP backbone (See Tab 5). Further, IL easily extends to multitask training without any performance degradation. Equipping the agent with detection massively boosts the success rate across all tasks except RoomVisit which does not require navigating to or manipulating objects.

Models	ObjNav			PickUp			Fetch			RoomVisit			Avg
	Success	SEL	%Rooms	Success	SEL	%Rooms	Success	SEL	%Rooms	Success	SEL	%Rooms	Success
TxEnc + GRU	44.7	33.8	47.7	84.8	81.4	30.3	10.5	9.0	41.8	34.5	31.8	72.6	43.6
nonTxEnc + TxDec	42.5	36.8	49.2	81.9	77.8	30.3	14.5	12.9	46.3	41.5	36.7	82.4	45.1
TxEnc + TxDec (Spoc)	55.0	42.2	56.3	90.1	86.9	30.3	14.0	10.5	49.3	40.5	35.7	81.1	49.9

Table 4. **Swapping transformer encoder and decoder with alternative architectures.** GRU performs much worse than TxDec for long horizon Fetch and RoomVisit tasks. TxEnc also has a clear advantage over EmbodiedCLIP [38]-style goal-conditioned visual feature extraction. All models use SigLIP image and text encoders.

Image Encoder	ObjNav			PickUp			Fetch			RoomVisit			Avg
	Success	SEL	%Rooms	Success	SEL	%Rooms	Success	SEL	%Rooms	Success	SEL	%Rooms	Success
Clip-RN50	19.6	12.1	44.1	64.1	60.2	30.5	1.8	0.8	43.4	21.0	19.5	62.6	26.6
DinoV2-ViT-S/14	47.5	32.7	53.1	87.7	84.2	30.3	9.9	7.8	44.7	34.0	31.3	77.5	44.8
SigLIP-ViT-B/16 (Spoc)	55.0	42.2	56.3	90.1	86.9	30.3	14.0	10.5	49.3	40.5	35.7	81.1	49.9

Table 5. **Comparing different image encoders.** SigLIP [85] significantly outperforms Clip-RN50 [38] and DinoV2 [50].

Window Size	ObjNav			PickUp			Fetch			RoomVisit			Avg
	Success	SEL	%Rooms	Success	SEL	%Rooms	Success	SEL	%Rooms	Success	SEL	%Rooms	Success
10	34.0	27.5	46.3	56.7	53.1	30.3	2.3	2.1	50.6	18.0	16.0	56.1	27.8
50	40.5	30.6	48.4	87.1	83.5	30.3	4.1	3.8	37.7	28.0	25.2	70.3	39.9
100 (Spoc)	55.0	42.2	56.3	90.1	86.9	30.3	14.0	10.5	49.3	40.5	35.7	81.1	49.9

Table 6. **Effect of context window.** Longer context is essential particular for long-horizon tasks like Fetch and RoomVisit.

Training Eps.	ObjNav		
	Success	SEL	%Rooms
1k	19.0	14.3	47.6
10k	39.0	31.1	52.9
100k (Spoc-1-task)	57.0	46.2	51.5

(a)

Houses	ObjNav		
	Success	SEL	%Rooms
100	43.5	35.2	53.6
10k (Spoc-1-task)	57.0	46.2	51.5

(b)

Expert	ObjNav		
	Success	SEL	%Rooms
Explore ObjNav	46.5	27.9	47.7
ObjNav (Spoc-1-task)	57.0	46.2	51.5

(c)

Table 7. **Effect of scale, house diversity, and choice of experts.** (a) Performance increases with more training episodes; (b) With the same number of episodes (100k), increasing diversity of houses boosts performance; (c) Training with experts that explore until the object becomes visible provides no gains.

placing Spoc's Transformer-based visual encoder (TxEnc) with nonTxEnc resulted in a performance drop of 4.8 points showing the superiority of TxEnc architecture for extracting relevant visual information. Swapping the action decoder (TxDec) with GRU showed an even larger drop of 6.3 points. We hypothesize TxDec outperforms GRU because of the ability to attend over observations and actions several 100 steps in the past while GRUs struggle with compressing history in a single history embedding.

Strong image encoders produce strong agents. Recent advances in image representations such as DinoV2 and SigLIP directly translate to gains in Embodied tasks. SigLIP particularly nearly doubles the average success rate of Clip on Chores-S (Tab. 5). Interestingly, while self-supervised DinoV2 significantly outperforms Clip, it lags behind SigLIP which is trained for image-text matching.

Long horizon tasks require long context windows. Transformer based embodied agents in the literature often rely on short context lengths to encode history due to compute constraints (*e.g.* RT-1 [4] uses 6 previous frames). We find that short context windows are detrimental to performance on longer horizon tasks like Fetch and RoomVisit (Tab. 6). Note that we train Spoc with limited context length but use all past observations for inference.

Scale and diversity of training data matters. In Tab. 7a, we show that performance on ObjNav steadily increases with number of training episodes. This raises a question, is it sufficient to collect a large number of samples from a limited number of houses? To answer this, we create two training sets with different numbers of houses - 100 houses with 1000 episodes each, and 10k houses with 10 episodes each. Spoc trained on the latter shows an absolute gain of

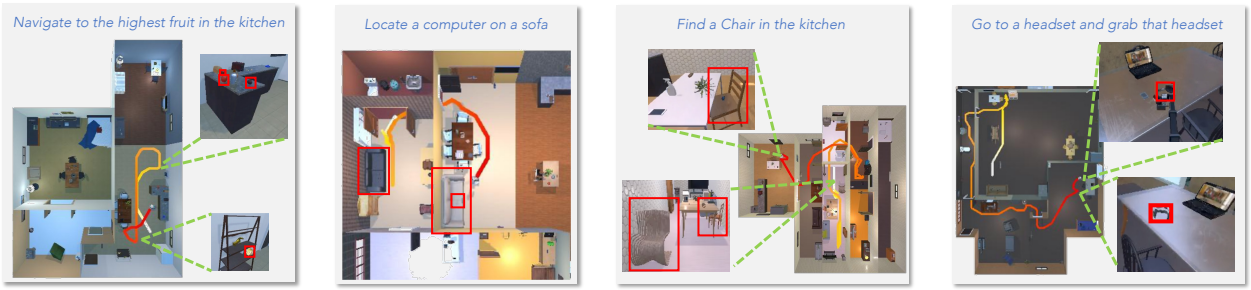

Figure 5. **SPOC's Behavior.** L to R: the first image depicts the agent navigating to all possible fruits in the kitchen and then going back to the highest located fruit; the second shows the agent scanning a sofa, then moving to another sofa and finally ending the episode when it sees the laptop; the third illustrates the agent skipping chairs in the living room to reach one in the kitchen; and the fourth demonstrates the agent visiting the headset, but then repositioning itself around the table to access a location where the headset is reachable.

Benchmark	ObjNav		ObjNavRoom		ObjNavRelAttr		ObjNavAfford		ObjNavLocalRef		ObjNavDesc		RoomNav		Avg
	Success	%Rooms	Success	%Rooms	Success	%Rooms	Success	%Rooms	Success	%Rooms	Success	%Rooms	Success	%Rooms	Success
ChoresNav -𝕊	57.5	55.7	50.3	54.6	54.6	62.2	62.4	53.0	45.1	51.5	30.6	49.9	74.5	48.1	53.6
ChoresNav -𝕃	38.7	53.4	54.2	55.7	38.5	56.0	43.5	48.0	44.5	58.7	30.5	56.8	67.5	49.9	45.3

Table 8. ChoresNav results to evaluate SPOC's ability to handle diverse target specifications for navigation.

13.5% (Tab. 7b). We believe that lack of house diversity in prior studies like PIRLNav [59] (which used 120 scenes) may have contributed to the inefficacy of IL on shortest-path trajectories for tasks like ObjNav.

Exploration based planners provide no gain. Prior work [59] found that IL with frontier exploration trajectories outperformed shortest-path trajectories. In Tab. 7c we compare 1-task SPOC to a variant trained with episodes generated by an exploration-based planner (see supplement for details) and find no gains using the latter. This reinforces the finding in Tab. 7b that the diversity of training environments is critical to the success of IL on shortest path trajectories.

SPOC follows open vocabulary instructions. Tab. 8 shows the performance of SPOC on several navigation tasks that require it to follow long instructions, disambiguate attributes, and understand relative distances. High performance on ObjNavAfford shows the ability of SPOC to understand instructions such as *locate an edible fruit that can best be used as a guacamole ingredient* or *find a tool that can best be used for cutting fruits and vegetables*, when there are multiple tools and edible fruits in the house. A high Success rate on ObjNavRoom for Chores-𝕃 compared to ObjNav show that locating a large vocabulary of objects is indeed easier when the agent is told which room the object lies in, and the high performance on RoomNav confirms that the agent has learned to identify rooms.

SPOC transfers effectively to the real world. To assess real-world generalization with no visual adaptation and no real world finetuning, we evaluate two of our best models in physical environments. These models were tested across 88 trials in two different real-world settings. Table 9, row 1 shows the performance of the RGB only SPOC. Row 2 is the performance of SPOC trained with GT Detection but evaluated in the real world with a DETIC object detector [87]. Comparing Table 9 with rows 3 and 4 from Table 3 shows that the performance drop between simulation and real is small overall and minimal for the navigation tasks. For manipulation tasks, note that the numbers in parentheses measure Soft Success, i.e. the model is rewarded if the gripper is within 6cm of the object, regardless of whether the heuristic grasping is successful. The Soft Success numbers are very similar to the simulation results indicating that the transfer of the learned policy from sim to real is very effective.

Model	ObjNav	PickUp	Fetch	RoomVisit	Average
SPOC	50.0	46.7 (66.7)	11.1 (33.3)	50.0	39.5
SPOC w/ Detic	83.3	46.7 (86.7)	44.4 (44.4)	50.0	56.1

Table 9. **Real world results**. Parenthetical numbers on manip. tasks indicate Soft Success: SPOC called PickUp sufficiently near the target, ignores heuristic grasping success/failure.

7.2. Agent Behavior

Figure 5 presents our qualitative examples, highlighting several intriguing behaviors exhibited by SPOC. Figure 1 illustrates additional qualitative trajectories in both simulated and real-world environments. These examples emphasize the model's capabilities in exploration, backtracking, scene and spatial comprehension, and instruction following. For more examples, please refer to the supplementary material.

8. Conclusion

In this work, we explore the potential of imitation learning for learning Embodied policies. Using shortest path expert planners in procedurally generated environments, it is now finally possible to generate training data at the scale and diversity needed to make techniques like Behavior Cloning work. More importantly, we show that agents learned by cloning experts in simulation not only generalize to novel environments but also to the real world.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3674–3683. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[2] 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. *CoRR*, abs/2006.13171, 2020. 2

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael S. Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong T. Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: vision-language-action models transfer web knowledge to robotic control. *CoRR*, abs/2307.15818, 2023. 3

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: robotics transformer for real-world control at scale. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. 2, 3, 7

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 5, 3

[6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 4

[7] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, et al. Goat: Go to any thing. *arXiv preprint arXiv:2311.06430*, 2023. 2

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

[9] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[10] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip W. Robinson, and Kristen Grauman. SoundSpaces: Audio-Visual Navigation in 3D Environments. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, pages 17–36. Springer, 2020. 2

[11] Changan Chen, Carl Schissler, Sanchit Garg, Philip Kobernik, Alexander Clegg, Paul Calamia, Dhruv Batra, Philip W. Robinson, and Kristen Grauman. SoundSpaces 2.0: A Simulation Platform for Visual-Acoustic Learning. In *NeurIPS*, 2022. 2

[12] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15084–15097, 2021. 3

[13] Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei Wang. Understanding Domain Randomization for Sim-to-real Transfer. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 3

[14] Felipe Codevilla, Matthias Müller, Alexey Dosovitskiy, Antonio M. López, and Vladlen Koltun. End-to-end driving via conditional imitation learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, 2017. 3

[15] Ishita Dasgupta, Christine Kaeser-Chen, Kenneth Marino, Arun Ahuja, Sheila Babayan, Felix Hill, and Rob Fergus. Collaborating with language models for embodied reasoning. *ArXiv*, abs/2302.00763, 2023. 3

[16] 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. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3161–3171. Computer Vision Foundation / IEEE, 2020. 2, 9

[17] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A Universe of Annotated 3D Objects. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13142–13153, 2022. 4, 5, 1, 2

[18] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied AI using procedural generation. In *NeurIPS*, 2022. 2, 4, 5, 1

[19] Matt Deitke, Rose Hendrix, Ali Farhadi, Kiana Ehsani, and Aniruddha Kembhavi. Phone2Proc: Bringing Robust Robots into Our Chaotic World. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 9665–9675. IEEE, 2023. 3, 9

[20] Ainaz Eftekhar, Kuo-Hao Zeng, Jiafei Duan, Ali Farhadi, Ani Kembhavi, and Ranjay Krishna. Selective visual representations improve convergence and generalization for embodied ai. *arXiv preprint arXiv:2311.04193*, 2023. 6

[21] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4497–4506. Computer Vision Foundation / IEEE, 2021. 2

[22] Kiana Ehsani, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. Object manipulation via visual target localization. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIX*, pages 321–337. Springer, 2022. 2

[23] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998. 2, 4

[24] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, Kuno Kim, Elias Wang, Michael Lingelbach, Aidan Curtis, Kevin T. Feigelis, Daniel Bear, Dan Gutfreund, David D. Cox, Antonio Torralba, James J. DiCarlo, Josh Tenenbaum, Josh H. McDermott, and Dan Yamins. ThreeDWorld: A Platform for Interactive Multi-Modal Physical Simulation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. 2

[25] Chuang Gan, Siyuan Zhou, Jeremy Schwartz, Seth Alter, Abhishek Bhandwaldar, Dan Gutfreund, Daniel L. K. Yamins, James J. DiCarlo, Josh H. McDermott, Antonio Torralba, and Joshua B. Tenenbaum. The ThreeDWorld Transport Challenge: A Visually Guided Task-and-Motion Planning Benchmark for Physically Realistic Embodied AI. *CoRR*, abs/2103.14025, 2021. 2

[26] Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S. Sukhatme. DialFRED: Dialogue-Enabled Agents for Embodied Instruction Following. *IEEE Robotics Autom. Lett.*, 7(4):10049–10056, 2022. 2

[27] Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. Navigating to objects in the real world. *Science Robotics*, 2023. 2

[28] Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo A Pires, and Rémi Munos. Neural predictive belief representations. *ICLR*, 2019. 2

[29] Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *International Conference on Machine Learning*, 2020. 2

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 3

[31] Daniel Ho, Kanishka Rao, Zhuo Xu, Eric Jang, Mohi Khansari, and Yunfei Bai. RetinaGAN: An Object-aware Approach to Sim-to-Real Transfer. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pages 10920–10926. IEEE, 2021. 3

[32] Wenlong Huang, P. Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *ICML*, 2022. 2

[33] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022. 3

[34] Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as I can, not as I say: Grounding language in robotic affordances. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, pages 287–318. PMLR, 2022. 2, 3

[35] Michael Janner, Qiyang Li, and Sergey Levine. Offline Reinforcement Learning as One Big Sequence Modeling Problem. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing*

Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 1273–1286, 2021. 3

[36] Parham Mohsenzadeh Kebria, Abbas Khosravi, Syed Moshfeq Salaken, and Saeid Nahavandi. Deep imitation learning for autonomous vehicles based on convolutional neural networks. *IEEE/CAA Journal of Automatica Sinica*, 7:82–95, 2020. 3

[37] Charles C. Kemp, Aaron Edsinger, Henry M. Clever, and Blaine Matulevich. The Design of Stretch: A Compact, Lightweight Mobile Manipulator for Indoor Human Environments. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pages 3150–3157. IEEE, 2022. 2, 4, 7

[38] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: CLIP embeddings for embodied AI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 14809–14818. IEEE, 2022. 2, 3, 6, 7

[39] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Kumar Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *ArXiv*, abs/1712.05474, 2017. 2, 5, 1

[40] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4392–4412. Association for Computational Linguistics, 2020. 2

[41] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Elliott Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, C. Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In *Conference on Robot Learning, 8-11 November 2021, London, UK*, pages 455–465. PMLR, 2021. 2

[42] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R. Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, Karen Liu, Jiajun Wu, and Li Fei-Fei. BEHAVIOR-1K: A Benchmark for Embodied AI with 1, 000 Everyday Activities and Realistic Simulation. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, pages 80–93. PMLR, 2022. 2

[43] G. Li, Matthias Müller, Vincent Casser, Neil G. Smith, Dominik Ludewig Michels, and Bernard Ghanem. Oil: Observational imitation learning. *Robotics: Science and Systems XV*, 2018. 3

[44] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023. 2

[45] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? *CoRR*, abs/2303.18240, 2023. 3

[46] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *ICLR*, 2023. 2

[47] John P. McCrae, Alexandre Rademaker, Francis Bond, Ewa Rudnicka, and Christiane Fellbaum. English WordNet 2019 - An Open-Source WordNet for English. In *Proceedings of the 10th Global Wordnet Conference, GWC 2019, Wroclaw, Poland, July 23-27, 2019*, pages 245–252. Global Wordnet Association, 2019. 5, 2, 4

[48] Luc Le Mero, Dewei Yi, Mehrdad Dianati, and Alexandros Mouzakitis. A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23:14128–14147, 2022. 3

[49] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1928–1937. JMLR.org, 2016. 3

[50] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 2, 7

[51] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. 5

[52] Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos A. Theodorou, and Byron Boots. Imitation learning for agile autonomous driving. *The International Journal of Robotics Research*, 39:286 – 302, 2019. 3

[53] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. VirtualHome: Simulating Household Activities via Programs. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-*

22, 2018, pages 8494–8502. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[54] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, Vladimir Vondrus, Théophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots. *CoRR*, abs/2310.13724, 2023. 2

[55] Henry Pulver, Francisco Eiras, Ludovico Carozza, Majd Hawasly, Stefano V Albrecht, and Subramanian Ramamoorthy. Pilot: Efficient planning by imitation learning and optimisation for safe autonomous driving. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1442–1449. IEEE, 2021. 3

[56] 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. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9979–9988. Computer Vision Foundation / IEEE, 2020. 2

[57] 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. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pages 8748–8763. PMLR, 2021. 2, 3

[58] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5163–5173. IEEE, 2022. 2, 3, 4

[59] Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In *Workshop on IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2, 8, 7

[60] Scott E. Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A Generalist Agent. *Trans. Mach. Learn. Res.*, 2022, 2022. 3

[61] Manolis Savva, Jitendra Malik, Devi Parikh, Dhruv Batra, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, and Vladlen Koltun. Habitat: A Platform for Embodied AI Research. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9338–9346. IEEE, 2019. 2

[62] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347, 2017. 3

[63] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, Micael Tchapmi, Kent Vainio, Josiah Wong, Li Fei-Fei, and Silvio Savarese. iGibson 1.0: A Simulation Environment for Interactive Tasks in Large Realistic Scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pages 7520–7527. IEEE, 2021. 2

[64] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10737–10746. Computer Vision Foundation / IEEE, 2020. 2

[65] Kunal Pratap Singh, Jordi Salvador, Luca Weihs, and Aniruddha Kembhavi. Scene Graph Contrastive Learning for Embodied Navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10884–10894, 2023. 2, 3

[66] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2

[67] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Mart'in-Mart'in, Fei Xia, Kent Vainio, Zheng Lian, Cem Gokmen, S. Buch, C. Karen Liu, Silvio Savarese, Hyowon Gweon, Jiajun Wu, and Li Fei-Fei. BEHAVIOR: Benchmark for Everyday Household Activities in Virtual, Interactive, and Ecological Environments. In *Conference on Robot Learning*, 2021. 2

[68] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John M. Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel X. Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training Home Assistants to Rearrange their Habitat. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 251–266, 2021. 2

[69] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 23–30. IEEE, 2017. 3

[70] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities. Technical Report MSR-TR-2023-8, Microsoft, 2023. 2

[71] Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multitask agents. *arXiv preprint arXiv:2302.01560*, 2023. 2

[72] Yao Wei, Yanchao Sun, Ruijie Zheng, Sai Vemprala, Rogerio Bonatti, Shuhang Chen, Ratnesh Madaan, Zhongjie Ba, Ashish Kapoor, and Shuang Ma. Is Imitation All You Need? Generalized Decision-Making with Dual-Phase Training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16221–16231, 2023. 3

[73] Luca Weihs, Jordi Salvador, Klemen Kotar, Unnat Jain, Kuo-Hao Zeng, Roozbeh Mottaghi, and Aniruddha Kembhavi. Allenact: A framework for embodied ai research. *arXiv preprint arXiv:2008.12760*, 2020. 2, 6, 1

[74] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[75] Luca Weihs, Unnat Jain, Iou-Jen Liu, Jordi Salvador, Svetlana Lazebnik, Aniruddha Kembhavi, and Alexander G. Schwing. Bridging the imitation gap by adaptive insubordination. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 19134–19146, 2021. 2, 4

[76] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: learning near-perfect pointgoal navigators from 2.5 billion frames. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 2, 3

[77] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-World Perception for Embodied Agents. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9068–9079. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[78] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A SimulAted Part-based Interactive ENvironment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[79] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline visual representation learning for embodied navigation. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023. 2

[80] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997. 7

[81] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectnav. *CoRR*, abs/2104.04112, 2021. 3

[82] Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Théophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John M. Turner, Zsolt Kira, Manolis Savva, Angel X. Chang, Devendra Singh Chaplot, Dhruv Batra, Roozbeh Mottaghi, Yonatan Bisk, and Chris Paxton. Homerobot: Open-vocabulary mobile manipulation. *CoRR*, abs/2306.11565, 2023. 2

[83] Naoki Yokoyama, Alexander Clegg, Eric Undersander, Sehoon Ha, Dhruv Batra, and Akshara Rai. Adaptive skill coordination for robotic mobile manipulation. *ArXiv*, abs/2304.00410, 2023. 2

[84] Kuo-Hao Zeng, Luca Weihs, Ali Farhadi, and Roozbeh Mottaghi. Pushing it out of the way: Interactive visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[85] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *ICCV*, abs/2303.15343, 2023. 2, 7

[86] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023. 9

[87] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IX*, pages 350–368. Springer, 2022. 8, 10

[88] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2242–2251. IEEE Computer Society, 2017. 3

[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. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 3357–3364. IEEE, 2017. 2