An Interactive Navigation Method with Effect-oriented Affordance

Xiaohan Wang[1,2], Yuehu Liu[1], Xinhang Song[2,3], Yuyi Liu[2,3], Sixian Zhang[2,3], Shuqiang Jiang[2,3]
[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an
[2]Key Lab of Intelligent Information Processing Laboratory of the Chinese Academy of Sciences (CAS),
Institute of Computing Technology, Beijing [3]University of Chinese Academy of Sciences, Beijing
wuhanwxh2016@stu.xjtu.edu.cn; liuyh@mail.xjtu.edu.cn
{xinhang.song, yuyi.liu, sixian.zhang}@vipl.ict.ac.cn; sqjiang@ict.ac.cn

Abstract

Visual navigation is to let the agent reach the target according to the continuous visual input. In most previous works, visual navigation is usually assumed to be done in a static and ideal environment: the target is always reachable with no need to alter the environment. However, the "messy" environments are more general and practical in our daily lives, where the agent may get blocked by obstacles. Thus Interactive Navigation (InterNav) is introduced to navigate to the objects in more realistic "messy" environments according to the object interaction. Prior work on InterNav learns short-term interaction through extensive trials with reinforcement learning. However, interaction does not guarantee efficient navigation, that is, planning obstacle interactions that make shorter paths and consume less effort is also crucial. In this paper, we introduce an effect-oriented affordance map to enable long-term interactive navigation, extending the existing map-based navigation framework to the domain of dynamic environment. We train a set of affordance functions predicting available interactions and the time cost of removing obstacles, which informatively support an interactive modular system to address interaction and long-term planning. Experiments on the ProcTHOR simulator demonstrate the capability of our affordance-driven system in long-term navigation in complex dynamic environments.

1. Introduction

Autonomously navigating to a target in complex environments is a core challenge for Embodied AI. Interactive Navigation (InterNav) [40, 45] aims to navigate more efficiently to a target point with object interaction in cluttered, dynamic environments, where the agent may get blocked by multiple obstacles during the long-term navigation. Compared to non-interactive navigation (e.g. ObjectNav, PointNav), the agent is challenged to plan not only long-term navigation but also interactions that benefit the navigation.

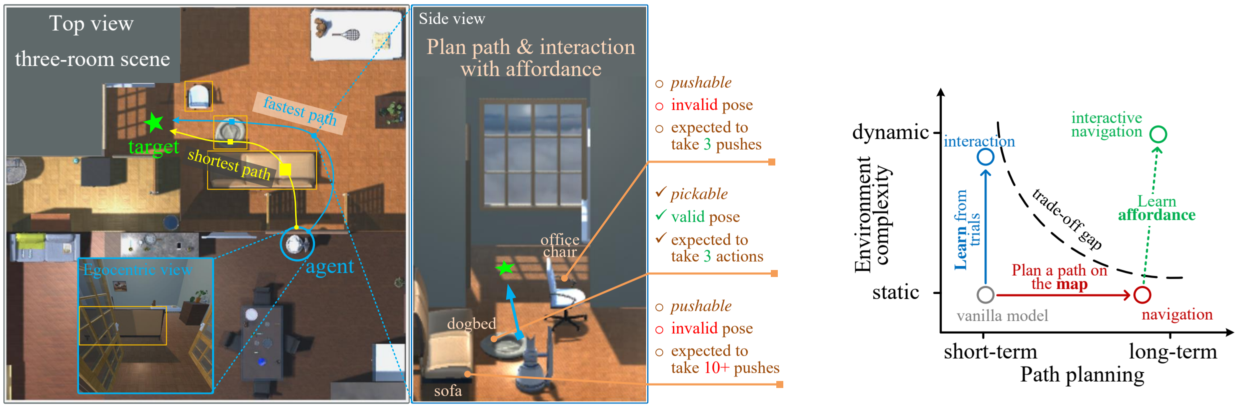
Prior work of InterNav [45] trains an RL-based policy to output the executable actions based on sensory egocentric observations and implicit episodic memories (from recurrent neural networks). This approach enables short-term interaction with extensive trials (corresponding to the red arrow in Figure 1(b)). However, effective interaction does not guarantee an efficient tour. For example (see Figure 1(a)), clearing the shortest path may cost more effort (or even failure when the obstacle gets stuck) than taking a detour at some point. Unlike other interactive tasks (e.g. grasp, rearrange), the purpose is not interaction itself but the effect that facilitates long-term navigation.

Overall, research on visual navigation can be categorized into learning-based methods [8, 27, 38] (prior InterNav works belong to) and map-based methods [4, 5, 25]. The latter builds an explicit map of spatial or semantic information and plans long-term paths on the map, which can be translated into action sequences. The stable map representation of the environment enables long-term navigation in unexplored scenes (corresponding to the red arrow in Figure 1(b)). However, the prerequisite for that to work is the determined reachability of a location that the agent can not reach it when it's occupied, otherwise it can. Therefore, the shortest path can be planned in the reachable area. On the contrary, the reachability in InterNav is uncertain, since the location occupied by or behind the obstacle may become reachable through interaction. To plan long-term paths in dynamic environments (towards the green node in Figure 1(b)), the agent needs to estimate whether an obstacle area can be reached, which depends on the interaction outcome.

As illustrated in Figure 1(b), there is a gap in front of

*Project page: https://github.com/polkalian/ADIN

This work is supported by National Natural Science Foundation of China (NSFC) under Grant 62350083. This work is in part by Beijing Natural Science Foundation under Grant JQ22012, and in part by the National Natural Science Foundation of China under Grant 62032022, 62125207, 62272443 and U23B2012.



(a) Illustration of InterNav driven by affordance

(b) Insight of introducing affordance

Figure 1. (a) In the task of InterNav, the agent marked with the blue circle aims to reach the target point marked with the green star and is blocked by multiple obstacles marked by orange boxes. Squares on the path denote the object interactions. Aware of the affordance of obstacles, a more efficient path can be planned by choosing the appropriate obstacle to interact with. (b) Prior methods for InterNav and map-based methods for visual navigation are limited to learning either short-term interaction or long-term navigation. To plan long-term paths in dynamic environments, the agent needs to learn affordance.

existing methods that they can't learn both long-term navigation and short-term interaction. We believe the "missing puzzle" is *affordance*, which indicates the potential of agent-environment interaction[12]. First of all, the affordance of objects usually refers to the available interactions on them, which indicates *how* the environment can be changed unlike the static spatial or semantic information. However, knowing *how* does not indicate the potential effect on navigation. We want to further know whether interaction with the obstacle can lead to reachable paths given the agent's capability of interaction. Then the long-term paths can be planned on the expected reachable area following the map-based framework.

To help embodied agents plan and conduct interactions, we introduce three levels of affordance: (1) **Object affordance**: the object attributes (e.g. *pushable*, *pickable*) determined by its shape, mass, etc. that are invariant to external factors. (2) **Pose affordance**: whether the object is currently interactive given the agent pose, which is dynamic during navigation. (3) **Effect affordance**: Whether the obstacle can be removed given agent's capability and the situation. As shown in Figure 1(a), the dogbed can be picked away within 3 actions while the sofa can hardly be removed with 10 pushes given its mass and the crowded space. In this paper, we propose a modular approach to address InterNav based on a multi-level affordance map. Overall, the model consists of a set of affordance functions, a mapping module, and an interactive policy. They are interfaced similarly to the map-based system that is widely applied in visual navigation research [4, 5, 25]. First the egocentric segmentations of multiple affordances are predicted with affordance functions from the RGB observation. Then the mapping module geometrically projects the affordance on the 3D voxel from the depth image and updates the top-down affordance map. The interactive policy produces a distance map that covers interaction costs according to the effect affordance and plans the most efficient path on the map. In the end, the policy outputs the action sequence of navigation or interaction given the agent location and the target position. The affordance functions are trained with the outcome of interactions that are conducted by the agent itself in the simulator.

We perform experiments on ProcTHOR simulator [9], which provides various multi-room scenes and supports object interactions. The proposed model outperforms existing end-to-end RL methods and map-based baselines. We systematically study the effective boundary of our affordance map on InterNav, verifying that our interactive modular system maintains effectiveness facing different interaction complexity.

2. Related Work

2.1. Interactive Navigation

Visual navigation, the cornerstone task of Embodied AI, has been extensively studied in the past decade. Most approaches address PointGoal [4, 28, 38] and ObjGoal [2, 5, 39, 48, 49] navigation that work under the assumption of static environment. Interactive navigation (InterNav) [34, 40, 45] considers dynamic environments and can be regarded as the downstream task of PointGoal navigation with the additional component of object interaction. Zeng et al. [45] focus on learning interaction with an additional perception module predicting the change of object keypoints with end-to-end RL in iTHOR environment [17]. However, implicit memory of egocentric perception only enables the short-term ability of interaction. Other interactive tasks such as manipulation [10, 13] and rearrangement [37, 44] also require the understanding of environmental dynamics. InterNav is distinctive in that the agent aims to ”interact for navigation” rather than to change the environment. Thus it’s crucial to learn the interaction effect on navigable areas and plan long-term navigation.

2.2. Map-based Navigation Methods

Classical approaches have formed the navigation paradigm of geometrical mapping and path planning [3, 16, 29]. Motivated by the drawbacks of learning-based methods, Chaplot et al. [4] proposed a hierarchical modular system that involves the learning procedure inside the modular to maintain efficiency and robustness. On the foundation of that, semantic information predicted from egocentric observations has been incorporated to construct semantic map for ObjectGoal navigation [5, 19, 23, 35, 47, 50]. The additional semantic memory supports the goal-oriented understanding of the environment. Recent modular approaches [11, 25, 46] further reason and predict the uncertain information of the environment like the target location and unexplored area. Our method learns effect-oriented affordance to obtain knowledge of interaction uncertainty and to foresee the interaction effect on navigation planning.

2.3. Affordance

The concept of Affordance has been widely applied in the field of psychology, computer vision [7, 18, 20, 22, 32, 42, 43], and robotics concerning interaction pose[1, 13, 15, 33], agent abilities[6], action sequence[31], effect relations[26, 41]. In the domain of embodied AI, several works have introduced affordance to perform effective interactive tasks. Nagarajan and Grauman [21] develop an embodied agent seeking new affordances actively through exploratory interactions. However, the learned affordances are still symbolized (denoted as attributes like *pickupable* and *toggleable*), which makes them less informative for execution. The affordance of navigation has been explored as interactive areas in language-guided interaction [14] and navigable areas in visual navigation[24]. In this work, we explore composite affordances that address the interactions in InterNav from actions to poses and their effect on navigation.

3. Method

3.1. InterNav Formulation

The agent aims to navigate to a reachable target point in an unseen environment with multiple obstacles (e.g. chair, box) spawned. At each step t, the agent receives egocentric RGB i_t, depth d_t images, and the relative target coordinate p_t. The agent then decides and executes an action $a_t \in A$, where A consists of 5 navigation actions (`MoveAhead`, `RotateRight`, `RotatLeft`, `LookUp`, `LookDown`), 6 interaction actions (`DirectionalPush` in 4 directions, `PickUp`, `Drop`), and `Done`. The agent is required to navigate within the distance of a step (0.25m) of the target and execute `Done` to claim a successful task episode. The episode ends when `Done` is taken or the number of steps exceeds the maximum budget of $T = 500$.

3.2. Method Overview

We propose Affordance Driven Interactive Navigation (ADIN), a modular system addressing long-term interactive navigation (see Figure 2). The system consists of three components: The affordance functions predict multiple affordances of the objects appearing in the current egocentric RGB observation. The mapping module builds and updates a global affordance map m_t storing the spatial and affordance information across the scene. The interactive policy plans a long-term path toward the target on the map and decides the actions of navigation or interaction according to the current agent location. Each module is introduced in the following parts.

3.3. Affordance Functions

The proposed affordance functions address interactive navigation by answering the questions of ”how to interact”, ”when to interact”, and ”whether to interact”. We first define three levels of affordances:

Object affordance. We consider two categories of object affordances corresponding to the two ways of interactions in InterNav: *pushable* and *pickable* indicate the attributes of a large object available to be pushed aside or a small object available to be picked up. **Pose affordance.** An interaction can be executed only if the agent is at a feasible pose where the object is in sight and within a certain distance. Thus we define an affordance of *visible* which indicates the interaction with the object is feasible given the current agent pose. Since *visible* is relative to the agent pose rather than a constant attribute of the object itself, it provides more guidance for the agent to decide *when* to interact.

Effect affordance. To decide *whether* to interact with an obstacle during navigation, the agent needs to estimate the potential effect of interaction including the benefit and the cost of effort. Considering the objective of InterNav, the desired effect of obstacle interaction is clearing the path efficiently. Therefore we quantize such criterion by measuring ”the time cost of removing the obstacle from its current position”. A high time cost means a low probability of reachability which suggests the agent should give up interacting with the obstacle. The limiting case is that it can not be moved away and it’s a dead end (a heavy table stuck in the corner). On the other side, the agent should consider a path

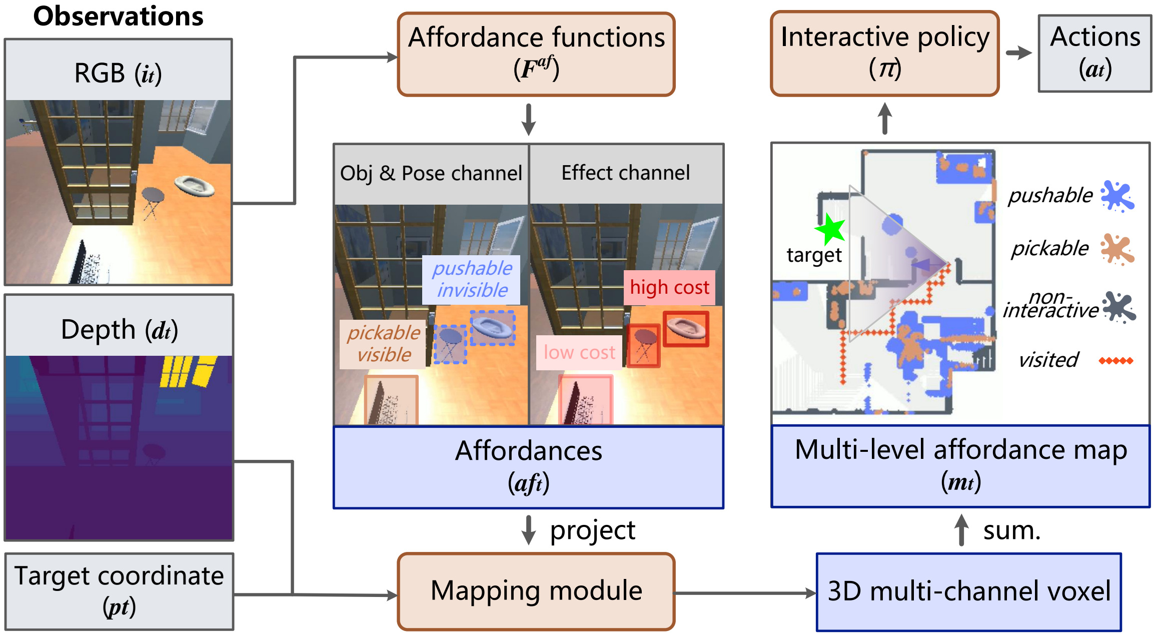


Figure 2. **Model overview.** The proposed model ADIN consists of a mapping module, four affordance functions, and an interactive policy. The affordance functions predict four levels of affordance af_t of objects from RGB observation. The mapper builds 3D voxel from the depth observation and obtains the top-down affordance map m_t. The interactive policy plans paths on the map toward the target coordinates and outputs actions accordingly.

interacting with a low time-cost obstacle when it's more efficient.

Formally, receiving an RGB observation i_t, we first obtain the region of interest (RoI) as bounding box b_t^n of object n and its local feature f_t^n with RoI pooling on the global feature map f_t^*. Then four affordance functions predict the multi-level affordances based on the features:

$$af_t^n = \{af_t^{pu}, af_t^{pi}, af_t^v, af_t^e\}^n = F^{af}(f_t^n, f_t^*) \qquad (1)$$

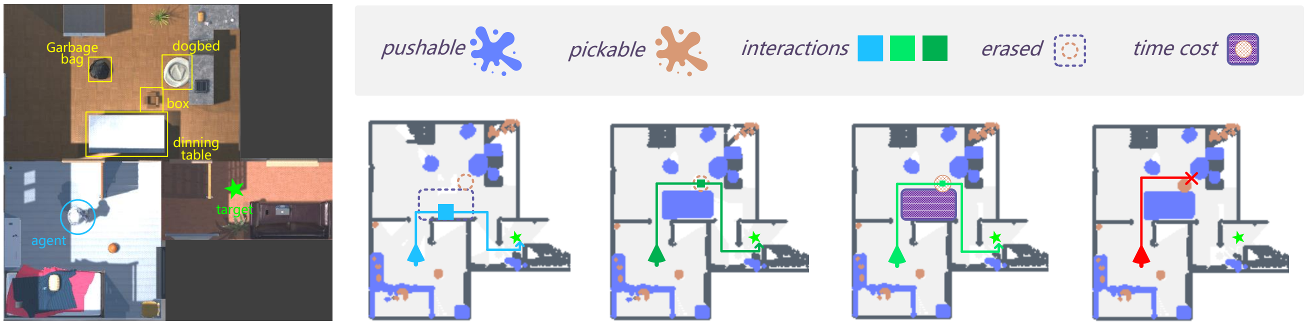
,where affordance af_t^n contains four affordances (respective to the ones we define) of object n. Four affordance functions $F^{af} = \{F^{pu}, F^{pi}, F^v, F^e\}$ take the concatenation of local and global features as input. We implement each function as the sequence of a two-layer MLP. While the values of $af_t^{pu}, af_t^{pi}, af_t^v$ are set as binary, the time cost af_t^e is normalized as a continuous value between 0 and 1 with min-max scaling.

Learning. To train functions F^{af}, we collect a static dataset in various scenes containing the data of features (f_t^n, f_t^*) and labels. In particular, the agent is randomly spawned around the obstacles and attempts to interact with `DirectionalPush` or `Pick` for multiple times. The labels of *pushable*, *pickable*, *visible* are annotated according to the success or failure of the interaction. The ground truth time cost of removing an object is obtained by the agent running trials of interaction and calculating the time steps it spends. We let the agent learn from the outcome of its own experience to maintain a robust understanding of the dynamic environment. The interaction strategy is identical to the interactive policy applied in our model (introduced in section 3.5). Affordance functions are trained through supervised learning with a binary cross entropy loss for discrete output and a mean squared error (MSE) loss for continuous output.

Essentially, the affordance functions provide the agent with a gateway to estimate the interaction uncertainty. The reliability of our approach lies in the proper understanding of agent's capability relative to the interaction complexity. Effective InterNav can be reached within our framework with both the mastery of interaction and matched estimation.

3.4. Mapping Module

Overall, the affordance mapper is responsible for aggregating the affordance and reachability information from egocentric depth frame d_t and affordances af_t^n up to time t into an allocentric map m_t. We follow the standard mapping procedure from visual navigation methods [5, 25]. The depth observation is used to compute a point cloud, where each point is associated with the predicted object affordances. Then the point cloud is projected in 3D space using differentiable geometric computations to get the voxel



Top-down view (a) High *confidence* (b) ADIN-discrete (c) ADIN-continuous (d) Low *confidence*

Figure 3. **Illustration of four map alterations of the interactive policy.** Given the occupancy map m_t^R, the agent plans the path based on its estimation of obstacles and its capability. (a) When the agent is confident, it erases all obstacles on the map and chooses the shortest path regardless of the interaction cost, which may end up less efficient. (d) When the agent is not confident, the target remains unreachable and no valid path can be planned. (b) ADIN-discrete erases obstacles selectively according to the effect affordance prediction, altering the map discretely. (c) ADIN-continuous operates on the continuous distance map, measuring the time cost of different obstacle interactions. The dashed areas denote the erasion of certain objects and the meshed areas denote the time cost calculated on the distance map.

representation. The top-down maps of multi-level affordances m_t^A : $4 \times M \times M$ and reachability (occupancy map) m_t^R : $1 \times M \times M$ are obtained by adding up the voxel representation vertically. Each point in the map corresponds to a 5cm×5cm area in the physical world. Therefore the current affordance map contains 8 channels: $m_t = \{m_t^A, m_t^R, m_t^{exp} p_t^m, l_t^m\}$: $8 \times M \times M$, where m_t^{exp} denotes the map of explored area, p_t^m, l_t^m represent the target and agent location in the map coordinate system, and M is the map size.

3.5. Interactive Policy

Given the affordance map m_t and the target p_t, the interactive policy π plans short-term actions a_t in an analytical manner. It differs from the policy for visual navigation [4, 5] that two modes of navigation and interaction cooperate alternately, since in a short-term view the agent can either interact with obstacles or navigate somewhere. At each step the mode is determined according to the agent location: the agent is in the interactive mode when an interactable (`pushable` or `pickable`) obstacle is `visible` and on the shortest path toward the target point, otherwise it's in the navigation mode. The affordance information stored on the map enables the agent to enter the interaction mode at feasible positions and take feasible interactions, avoiding failed and useless action execution.

For the navigation mode, we modify the distance computation of the Fast Marching Method (FMM) [30] for InterNav which analyzes the occupancy map m_t^R and affordance map m_t^A, and plans actions along the shortest path to the target. Since the reachability of an obstacle area is uncertain, its approximate distance to the target should neither be positive infinity (non-interactive obstacles like walls) nor standard distance (like floor) which ignores the time cost of interaction. Hence, assuming standard FMM obtains a distance map m_t^{dis} given m_t^R, we compute a time-cost measured distance map by adding the equivalent distance of effect affordance map to it:

$$m_t^{dis^*} = m_t^{dis} + \alpha \cdot m_t^e \cdot grid_m \tag{2}$$

, where m_t^e : $1 \times M \times M$ is the effect affordance channel of m_t^A, α is the coefficient of time cost (given af_t^e is normalized), and $grid_m$ is a constant denoting the distance of a step on the map. Thus $\alpha \cdot m_t^e \cdot grid_m$ represents the distance the agent would travel if it spends the time of clearing obstacles on navigation. The value of α therefore is endowed with the meaning of the maximum time steps an interaction is expected to take. The shortest path planned on the new distance map now represents the path that costs the least expected time, namely the most efficient path. For instance, blocked by a `pushable` chair, the agent may choose to bypass it when there's a path aside, since the ”chair area” is assigned with additional distance. When there is no reachable path nearby, the agent will move toward the chair for interaction since it's the only area with the distance value lower than positive infinity. Therefore through the continuous map alteration above, the model (referred to as ADIN-continuous) plans more efficient paths than merely the shortest paths. At each time step, the map is updated according to the new observation and the shortest path is replanned.

Besides modifying the distance map, we also perform a variant (referred to as ADIN-discrete) making alterations directly on the ocupancy map:

$$m_t^{R^*} = m_t^R - round(m_t^e - \beta) \tag{3}$$

where $round(\cdot)$ is the rounding operation and β is the threshold of m_t^e to determine whether the obstacle interaction should be ruled out given agent's estimation. The minus sign denotes the element-wise difference between two binary arrays. Essentially, the map alteration represents the 'confidence' level of agent being able to remove the obstacle. By erasing an obstacle from the map or assigning a low time cost to it, the agent tends to ignore the obstacle during path planning and is confident to handle it. We illustrate four map alterations and the path planned corresponding to them in Figure 3.

For the interaction mode, we let the agent interact with the closest obstacle and pick the actions by analyzing the layout of the surroundings and the expected path to navigate based on the map memory. The agent executes a series of interactions (i.e. series of `DirectionalPush` for `pushable` obstacles, sequence of [`PickUp`, rotations, `Drop`] for `pickable` obstacles) to place the obstacle out of the expected path at a vacant location. We refresh the map with egocentric observation in the interaction mode rather than update it by adding up the point cloud, so that the displacement of obstacles can be captured immediately. If the agent fails to move the obstacle out of the path after several rounds of interaction, we add it to the occupancy map m_t^R so that FMM computes the infinity distance of that obstacle area on m_t^{dis}, knowing it's a dead end and would choose another path toward the target.

4. Experiments

4.1. Experiment Setup

Dataset settings. We evaluate InterNav in the simulator of ProcTHOR [9], which supports various sensory signals and object interactions, and provides 12k flexible multi-room scenes (1∼10+ rooms). In an episode, we randomly set the starting and target points in different rooms and randomly spawn multiple obstacles across rooms. The dataset consists of 450k training episodes (across 9k scenes), 5k validation episodes (across 100 scenes), and 100 testing episodes (across 100 scenes). The static datasets for detection and affordance prediction training are collected within the training set. To evaluate the ability of long-term planning and multiple interactions, we especially report the results on the *hard* split of the test set, which contains episodes collected in the scenes with more than 4 rooms.

Following the environmental setting of [45], we let `MoveAhead` move the agent ahead by $grid = 0.25m$, `RotateRight` and `RotateLeft` change the agent's azimuth angle by ±90 degrees, `LookUp` and `LookDown` rotate the agent's camera elevation angle by ±30 degrees. The `DirectionalPushs` let the agent push (along ±z and ±x axis) the closest visible object with a constant force. The `Pick` puts the object in an invisible pocket and is valid when the pocket is empty. The `Drop` put the object in the pocket in front of the agent at a distance. The interactions are valid when the object has the corresponding attributes such as *pickable* and *movable*. The agent takes the `END` to indicate that it has completed an episode.

Evaluation metrics. We adopt Success Rate (SR), Final Distance to Target (FDT), Success weighted by Path Length (SPL), and Success weighted by Time Steps (STS). SR is the ratio of successful episodes in total episodes. FDT is the average geodesic distance between the agent and the goal when the episode is finished. SPL is calculated as $\frac{1}{N}\sum_{n=1}^{N} Suc_n \frac{L_n}{max(P_n, L_n)}$, where N is the total episodes number, Suc_n is the successful indicator of n-th episode, L_n is the shortest path length, and P_n is the length of the real path. STS is introduced to measure the time efficiency of task completion, given that the time spent by interactions can not be measured by path length: $STS = \frac{1}{N}\sum_{n=1}^{N} Suc_n \frac{L_n/grid}{TS_n}$, where TS_n is the timesteps the agent takes to complete the task, and $grid = 0.25m$ is the unit distance of a step. STS can be regarded as a time-measurement variant of SPL and the score is higher when the agent accomplishes the task with less time.

4.2. Implementation Details

Our method is implemented and evaluated with the AllenAct [36] framework. The egocentric observation is set as 300*300 RGB and depth images. The size of affordance map M is set as 400, the resolution is 0.05, thus a point on the map corresponds to a cell of $25cm^2$ and a step on the map crosses $grid_m = 5$ cells. A cell is determined as occupied with an obstacle when the sum of point clouds number in this area exceeds 10. The hyper-parameters of ADIN α, β are set as 5, 9 according to the ablation results. We obtain the regions of interest b_t^n with a detection model Yolov7, which is COCO-pretrained and finetuned on 150k images from our training set with 20 categories of obstacles (same as prior work [45]). The global feature f_t^* is the feature map extracted with the backbone of Yolov7 and the size of local feature f_t^n is set as 256×7×7. We avoid using the semantic information from the detection results and train the affordance functions with the experimental interaction results. The hidden size of the affordance functions is 256. All affordance functions are trained with the static dataset with 100k training samples for 20 epochs and we pick the model based on the result on 10k validation samples. The model is trained with a batch size of 128 and the Adam optimizer with a start learning rate of 1e-5.

4.3. Baselines

We compared our model with two learning-based models (DD-PPO[38] and NIE[45]) and two map-based baselines. Since InterNav has not been addressed with map-based method before, map-based baselines are set as several variants of our system and existing non-interactive model:

Map+RI: This model takes no affordance and tries **random**

Methods	all				hard			
	SR↑	**FDT↓**	**SPL↑**	**STS↑**	**SR**	**FDT**	**SPL**	**STS**
DD-PPO [38]	38.3	5.31	23.0	13.1	21.3	8.68	11.3	6.35
NIE [45]	50.0	4.50	29.1	14.4	37.3	7.38	20.5	8.72
Map+RI	18.5	6.33	12.0	7.93	4.12	9.30	3.02	2.16
ADIN+GP	41.2	5.18	21.8	13.1	25.9	8.22	13.1	6.42
ADIN-discrete	54.3	3.85	27.3	14.2	40.2	6.29	20.4	9.04
ADIN-continuous	**59.0**	**3.46**	**31.3**	**16.6**	**46.1**	**5.52**	**23.6**	**11.3**

Table 1. **Comparison with learning-based models and map-based baselines.** We report the average performance of 3 tests.

Component ablation					all				hard			
OA	PA	TC	EA	GT	**SR**	**FDT**	**SPL**	**STS**	**SR**	**FDT**	**SPL**	**STS**
✓					45.9	3.90	25.9	14.2	36.4	6.39	17.4	8.33
✓	✓				49.0	3.78	27.4	14.9	35.9	6.08	18.9	8.17
✓	✓	✓			55.0	3.40	30.1	15.7	42.0	5.49	20.4	9.19
✓	✓	✓	✓		59.0	3.46	31.3	16.6	46.1	5.52	23.6	11.3
✓	✓	✓	✓	✓	61.2	3.37	32.3	17.2	48.3	5.40	27.7	12.1

Table 2. **Model component ablations.** OA: object affordance, PA: pose affordance, TC: time cost (k=5), EA: effect affordance (continuous variant), GT: ground truth object mask.

interactions when getting stuck during navigation. The policy plans paths seeing all objects on the map as non-interactive and executes actions following the local policy of [5].

ADIN+GP: Instead of directly setting the target point as the goal of the policy, this model learns to plan subgoals with a **global policy** like ObjectGoal navigation methods [5]. The global policy is trained with reinforcement learning (PPO) and the reward is set as the change of target distance. The global policy is trained on the training set for 3 million steps. We add the global policy to study whether the agent can learn to plan which obstacles to interact with by setting subgoals.

4.4. Ablations

We evaluate several variants of our model to study the following questions. First, we study the impact of model components by ablating different affordance components (Table 2) and hyper-parameters (Figure 4(b)(c)). Note that we adjust the strategy of interaction policy accordingly since the agent executes interactions based on the affordances. Second, we alter agent's interactive capability to study the performance boundary of the modular system (PI, NI in Figure 4(a)), since the interaction may not bring the same outcome when transferring to the real world. Third, we also study the impact of agent's "confidence" in its capability (see Figure 4(a)), namely how much obstacles affect path planning. The more *confident* the agent is, the less likely it will take a longer path due to obstacles.

PI (upper bound): This model takes in the ground truth affordances and performs *perfect interactions*, considering the ideal interaction situation under our system. The agent removes the *visible* obstacle from the scene once it takes the interaction. In this case, the planned paths are equivalent to the ones planned in open environments with few obstacles.

NI (lower bound): This model is equipped with no interaction capability that it regards all objects as non-interactive. NI is different from Map+RI by removing the random interactions. In this case, the planned paths are equivalent to the ones planned in cluttered environments with multiple obstacles.

ADIN w/k TC: This line of models quantize agent's "confidence" in interaction with k steps of **time cost** added to the distance map m_t^{dis}. Instead of estimating the time cost on obstacles with effect affordance af_t^e, the higher constant value k uniformly increases the expected cost of interaction and reduces the interaction as a result.

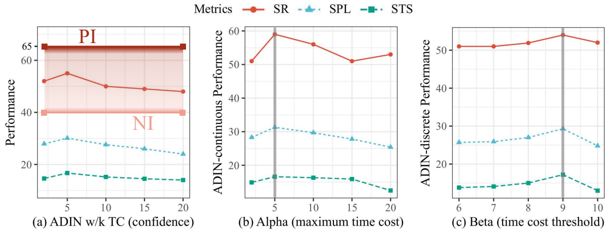


Figure 4. **Parameter ablations of TC, α, β.** The SR performances of PI (upper bound) and NI (upper bound) and the effective range between them are marked with the red gradient area.

4.5. Results Analysis

Comparison with baselines. As shown in Table 1, the proposed models gain the best performance on both splits. ADIN-continuous outperforms the prior learning-based model NIE by 9.0% and 8.8% SR in the *all* and *hard* split, respectively. The advantage on the *hard* split shows that the explicit map memory helps long-term planning better compared to the implicit neural memory. Both variants of ADIN outperform the map-based baseline Map+RI by a large margin (+40.5%/+35.8% SR and +8.67%/+6.27% STS on the *all* split), showing the effectiveness of our modular system that extends the map-based framework to the domain of interactive tasking. The gap between ADIN-discrete and ADIN-continuous suggests that a fine estimation of interaction effect facilitates the path planning better. ADIN+GP gains poorer performance than ADIN, indicating that planning interactions with subgoal is hard to learn through RL and may result in ineffective navigation. ADIN's advantage over RL-based methods on STS is less significant since they end the task actively before consuming too much time.

Ablation study. As shown in Table 2, each component of the affordance map contributes to the efficacy of ADIN

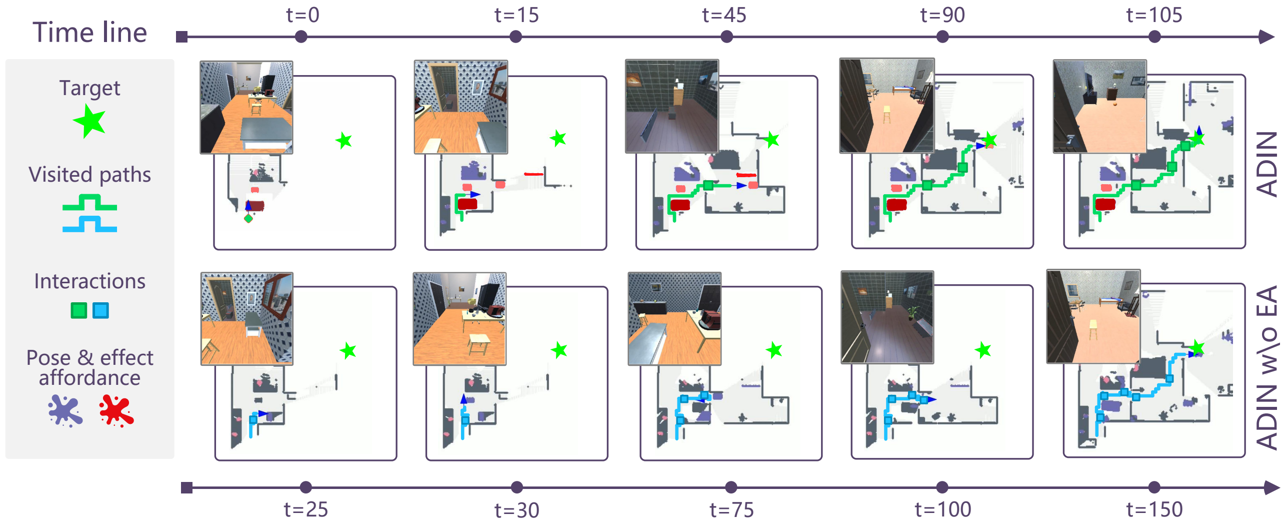


Figure 5. **Qualitative case of ADIN and ADIN without effect affordance.** We present the RGB observation and the output affordance map of two models at several key steps (before or after interaction). For the above line, the pose and effect affordances are marked on the map, which affects agent's path planning. The visited paths of agents are marked with lines on the map.

Methods	Δ BR	ISR	PuSR	PiSR
DD-PPO [38]	10.1	31.8	49.1	27.3
NIE [45]	19.3	47.9	51.1	47.4
Map+RI	6.7	12.6	14.7	6.52
Map+OA	6.8	54.0	58.3	10.3
Map+OA+PA	10.7	64.5	86.8	25.0
ADIN-continuous	21.2	72.9	83.3	31.4

Table 3. **Obstacle interaction results.** ΔBR: decreased blocked ratio, ISR: interaction success rate, PuSR: push success rate, PiSR: pick success rate

progressively, including object affordance (*how*), pose affordance (*when*), time cost on the distance map, and effect affordance (*whether*) concerning the process of interaction navigation. Note that the effect affordance is applied on top of the time cost calculation on the distance map, and further considers the features of different obstacles.

As shown in Figure 4(a), the performance of our interactive modular system falls in the range between PI and NI (40%-65% SR) and it's better when the "confidence" TC matches its real capability. Thus in our case, the agent's capability is around TC=5, meaning it takes an average of 5 steps to remove the obstacle off the path. The results of α, β ablations (see Figure 4(b)(c)) indicates that a proper estimation of interaction effect leads to better path planning.

Study of obstacle interaction. We evaluate the interaction outcome more directly with additional metrics in Table 3, different from SR, SPL that assess the overall completion of the task:(1) Ratio of episodes (ΔBR) that the agent is blocked initially and makes clear paths in the end. This metric measures the effect of obstacle interaction on navigation objective. (2) Overall (ISR) and separate (PuSR, PiSR) success rates of interaction are the ratios of conducting valid interactions, which measure the accuracy of actions. The results indicate that object affordance (line 4) and pose affordance (line 5) effectively increase the interaction accuracy but benefit less for the navigation objective (i.e. clearing the path) without the guidance of effect affordance (line 6). The performances of map-based models on PiSR are relatively low since *pickable* objects are usually small and hard to recognize from a distance.

Case study. We visualize the paths and affordance maps produced by ADIN with and without the help of effect affordance in an episode (see Figure 5). Although ADIN w/o EA may finally complete the task, it plans interactions with over 5 obstacles and costs unnecessary effort, while ADIN achieves a more efficient tour with 45 steps less and only 3 necessary interactions.

5. Conclusion

We propose an effect-oriented affordance map for Interactive Navigation (InterNav). The insight is to model the interaction uncertainty with affordance and extend the existing map-based framework into the domain of dynamic environments. We construct an interactive modular system consisting of a set of affordance functions, a mapping module, and an interactive policy. The system plans long-term paths considering the potential effect and effort of obstacle interaction. Experiments in ProcTHOR verify the effectiveness of our approach in complex dynamic environments.

References

[1] Paola Ardón, Eric Pairet, Ronald PA Petrick, Subramanian Ramamoorthy, and Katrin S Lohan. Learning grasp affordance reasoning through semantic relations. *IEEE Robotics and Automation Letters*, 4(4): 4571–4578, 2019. 3

[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. *arXiv preprint arXiv:2006.13171*, 2020. 2

[3] John Canny. *The complexity of robot motion planning*. MIT press, 1988. 3

[4] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020. 1, 2, 3, 5

[5] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020. 1, 2, 3, 4, 5, 7

[6] Anthony Chemero. An outline of a theory of affordances. In *How Shall Affordances Be Refined?*, pages 181–195. Routledge, 2018. 3

[7] Joya Chen, Difei Gao, Kevin Qinghong Lin, and Mike Zheng Shou. Affordance grounding from demonstration video to target image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6799–6808, 2023. 3

[8] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. *arXiv preprint arXiv:1903.01959*, 2019. 1

[9] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, et al. Procthor: Large-scale embodied ai using procedural generation. *arXiv preprint arXiv:2206.06994*, 2022. 2, 6

[10] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4497–4506, 2021. 3

[11] Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. Learning to map for active semantic goal navigation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 3

[12] James J Gibson. The theory of affordances. *Hilldale, USA*, 1(2):67–82, 1977. 2

[13] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 3

[14] Zhiwei Jia, Kaixiang Lin, Yizhou Zhao, Qiaozi Gao, Govind Thattai, and Gaurav S Sukhatme. Learning to act with affordance-aware multimodal neural slam. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5877–5884. IEEE, 2022. 3

[15] Juntao Jian, Xiuping Liu, Manyi Li, Ruizhen Hu, and Jian Liu. Affordpose: A large-scale dataset of hand-object interactions with affordance-driven hand pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14713–14724, 2023. 3

[16] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12 (4):566–580, 1996. 3

[17] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 3

[18] Yuanzhi Liang, Xiaohan Wang, Linchao Zhu, and Yi Yang. Maal: Multimodality-aware autoencoder-based affordance learning for 3d articulated objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 217–227, 2023. 3

[19] Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A strong baseline for indoor object navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2022, Kyoto, Japan, October 23-27, 2022*, pages 3287–3293. IEEE, 2022. 3

[20] Lorenzo Mur-Labadia, Jose J Guerrero, and Ruben Martinez-Cantin. Multi-label affordance mapping from egocentric vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5238–5249, 2023. 3

[21] Tushar Nagarajan and Kristen Grauman. Learning affordance landscapes for interaction exploration in 3d environments. *Advances in Neural Information Processing Systems*, 33:2005–2015, 2020. 3

[22] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction

hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8688–8697, 2019. 3

[23] Medhini Narasimhan, Erik Wijmans, Xinlei Chen, Trevor Darrell, Dhruv Batra, Devi Parikh, and Amanpreet Singh. Seeing the un-scene: Learning amodal semantic maps for room navigation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVIII*, pages 513–529. Springer, 2020. 3

[24] William Qi, Ravi Teja Mullapudi, Saurabh Gupta, and Deva Ramanan. Learning to move with affordance maps. *arXiv preprint arXiv:2001.02364*, 2020. 3

[25] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18890–18900, 2022. 1, 2, 3, 4

[26] Erol Şahin, Maya Cakmak, Mehmet R Doğar, Emre Uğur, and Göktürk Üçoluk. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15(4):447–472, 2007. 3

[27] Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. *arXiv preprint arXiv:1810.02274*, 2018. 1

[28] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 2

[29] James A Sethian. A fast marching level set method for monotonically advancing fronts. *proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. 3

[30] James A Sethian. Fast marching methods. *SIAM review*, 41(2):199–235, 1999. 5

[31] Mark Steedman. Formalizing affordance. In *Proceedings of the Twenty-fourth Annual Conference of the Cognitive Science Society*, pages 834–839. Routledge, 2019. 3

[32] Jiajin Tang, Ge Zheng, Jingyi Yu, and Sibei Yang. Cotdet: Affordance knowledge prompting for task driven object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3068–3078, 2023. 3

[33] Liquan Wang, Nikita Dvornik, Rafael Dubeau, Mayank Mittal, and Animesh Garg. Self-supervised learning of action affordances as interaction modes. *arXiv preprint arXiv:2305.17565*, 2023. 3

[34] Xiaohan Wang, Yuehu Liu, Xinhang Song, Beibei Wang, and Shuqiang Jiang. Camp: Causal multipolicy planning for interactive navigation in multiroom scenes. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[35] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15625–15636, 2023. 3

[36] 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. 6

[37] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5922–5931, 2021. 3

[38] 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. *arXiv preprint arXiv:1911.00357*, 2019. 1, 2, 6, 7, 8

[39] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6750–6759, 2019. 2

[40] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 5(2):713–720, 2020. 1, 2

[41] Danfei Xu, Ajay Mandlekar, Roberto Martín-Martín, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. Deep affordance foresight: Planning through what can be done in the future. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6206–6213. IEEE, 2021. 3

[42] Yuhang Yang, Wei Zhai, Hongchen Luo, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Grounding 3d object affordance from 2d interactions in images. *arXiv preprint arXiv:2303.10437*, 2023. 3

[43] Yufei Ye, Xueting Li, Abhinav Gupta, Shalini De Mello, Stan Birchfield, Jiaming Song, Shubham Tulsiani, and Sifei Liu. Affordance diffusion: Synthesizing hand-object interactions. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 22479–22489, 2023. 3

[44] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021. 3

[45] 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*, pages 9868–9877, 2021. 1, 2, 6, 7, 8

[46] Albert J. Zhai and Shenlong Wang. PEANUT: predicting and navigating to unseen targets. *CoRR*, abs/2212.02497, 2022. 3

[47] Jiazhao Zhang, Liu Dai, Fanpeng Meng, Qingnan Fan, Xuelin Chen, Kai Xu, and He Wang. 3d-aware object goal navigation via simultaneous exploration and identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 6672–6682. IEEE, 2023. 3

[48] Sixian Zhang, Weijie Li, Xinhang Song, Yubing Bai, and Shuqiang Jiang. Generative meta-adversarial network for unseen object navigation. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIX*, pages 301–320. 2

[49] Sixian Zhang, Xinhang Song, Yubing Bai, Weijie Li, Yakui Chu, and Shuqiang Jiang. Hierarchical object-to-zone graph for object navigation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 15110–15120. IEEE, 2021. 2

[50] Sixian Zhang, Xinhang Song, Weijie Li, Yubing Bai, Xinyao Yu, and Shuqiang Jiang. Layout-based causal inference for object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10792–10802, 2023. 3