SOON: Scenario Oriented Object Navigation with Graph-based Exploration

Fengda Zhu[1]   Xiwen Liang[2]   Yi Zhu[3]   Qizhi Yu[4]   Xiaojun Chang[1*]   Xiaodan Liang[2]

[1]Monash University   [2] Sun Yat-sen University

[3] University of Chinese Academy of Sciences   [4]Zhijiang Laboratory

zhufengda@yahoo.com   liangcici5@gmail.com   zhu.yee@outlook.com

qyu@ieee.org   cxj273@gmail.com   xdliang328@gmail.com

Abstract

The ability to navigate like a human towards a language-guided target from anywhere in a 3D embodied environment is one of the 'holy grail' goals of intelligent robots. Most visual navigation benchmarks, however, focus on navigating toward a target from a fixed starting point, guided by an elaborate set of instructions that depicts step-by-step. This approach deviates from real-world problems in which human-only describes what the object and its surrounding look like and asks the robot to start navigation from anywhere. Accordingly, in this paper, we introduce a Scenario Oriented Object Navigation (SOON) task. In this task, an agent is required to navigate from an arbitrary position in a 3D embodied environment to localize a target following a scene description. To give a promising direction to solve this task, we propose a novel graph-based exploration (GBE) method, which models the navigation state as a graph and introduces a novel graph-based exploration approach to learn knowledge from the graph and stabilize training by learning sub-optimal trajectories. We also propose a new large-scale benchmark named From Anywhere to Object (FAO) dataset. To avoid target ambiguity, the descriptions in FAO provide rich semantic scene information includes: object attribute, object relationship, region description, and nearby region description. Our experiments reveal that the proposed GBE outperforms various state-of-the-arts on both FAO and R2R datasets. And the ablation studies on FAO validates the quality of the dataset.

1. Introduction

Recent research efforts [49, 19, 17, 47, 33, 45, 32] have achieved great success in embodied navigation tasks. The agent is able to reach the target by following a variety of instructions, such as a word (*e.g.* object name or room name) [49, 40], a question-answer pair [11, 18], a natural language sentence [3] or a dialogue consisting of multiple sentences [45, 55].

However, these navigation approaches are still far from real-world navigation activities. Current vision language based navigation tasks such as Vision-language Navigation (VLN) [3], Navigation from Dialog History (NDH) [45] focus on navigating to a target by a fixed trajectory, guided by an elaborate set of instructions that outlines every step. These approaches fail to consider the case in which the complex instruction provided only target description while the starting point is not fixed. In real-world applications, people often do not provide detailed step-by-step instructions and expect the robot to be capable of self-exploration and autonomous decision-making. We claim that the ability to navigate towards a language-guided target from anywhere in a 3D embodied environment like human would be of great importance to an intelligent robot.

To address these problems, we propose a new task, named Vision Situated Object Navigation (SOON), where an agent is instructed to find a thoroughly described target object inside a house. The navigation instructions in SOON are target-oriented rather than step-by-step babysitter as in previous benchmarks. There are two major features that makes our task unique: target orienting and starting independence. A brief example of a navigation process in SOON is illustrated in Fig. 1. Firstly, different from conventional object navigation tasks defined in [49, 40], instructions in SOON play a guidance role in addition to distinguish a target object class. An instruction contains thorough descriptions to guide the agent to find a unique object from anywhere in the house. After receiving an instruction in SOON, the agent first searches a larger-scale area according to the region descriptions in the instruction, and then gradually narrows the search space to the target area. Compared with step-by-step navigation settings [3] or object-goal navigation settings [49], this kind of coarse-to-fine navigation process is more closely resembles a real-world situation. Moreover, the SOON task is starting-independent. Since the language instructions contain geographic region descriptions rather than trajectory
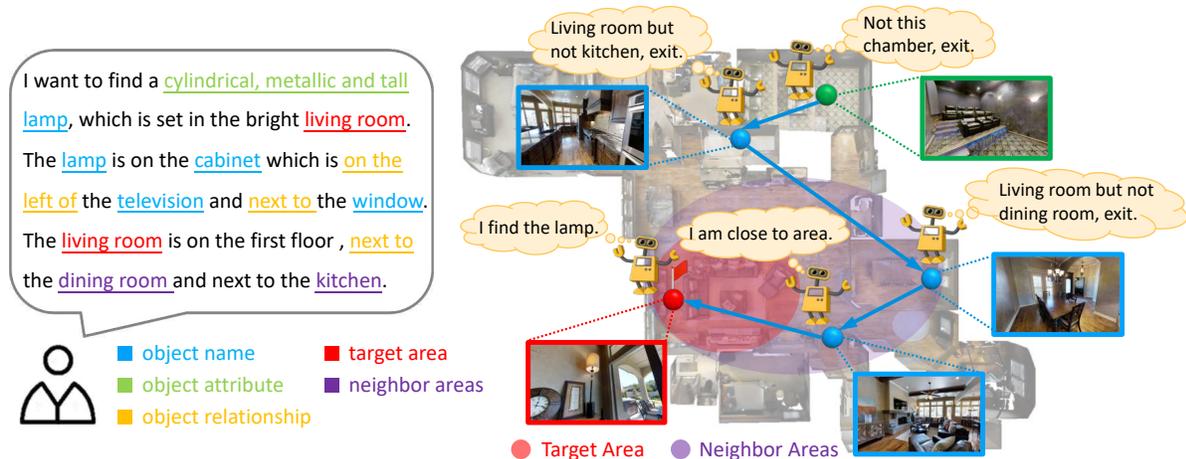
---

*Corresponding author.

Figure 1: An example of the navigation process in SOON. An agent receives a complex natural language instruction consisting of multiple kinds of descriptions (left-hand side). During the agent navigates among different rooms, it searches a larger-scale area first, then gradually narrows down the search scope according to the visual scene and the instructions.

specific descriptions, they do not limit how the agent finds the target. By contrast, in step-by-step navigation tasks such as Vision Language Navigation [3] or Cooperative Vision-and-Dialog Navigation [45], any deviation from the directed path may be considered as an error [25]. We present an overall comparison between the SOON task and existing embodied navigation tasks in Tab. 1.

In this work, We propose a novel Graph-based Semantic Exploration (GBE) method to suggest a promising direction in approaching SOON. The proposed GBE has two advantages compared with previous navigation works [3, 17, 47]. Firstly, GBE models the navigation process as a graph, which enables the navigation agent to obtain a comprehensive and structured understanding of observed information. It adopts graph action space to significantly merge the multiple actions in conventional sequence-to-sequence models [3, 17, 47] into one-step decision. Merging actions reduces the number of predictions in a navigation process, which makes the model training more stable. Secondly, different from other graph-based navigation models [14, 9] that use either imitation learning or reinforcement to learn navigation policy, the proposed GBE combines the two learning approaches and proposes a novel exploration approach to stabilize training by learning from sub-optimal trajectories. In imitation learning, the agent learns to navigate step by step under the supervision of ground truth label. It causes severe overfitting problem since labeled trajectories occupy only a small proportion of the large trajectory space. In reinforcement learning, the navigation agent explores large trajectory space, and learn to maximize the discounted reward. Reinforcement learning leverages sub-optimal trajectories to improve the generalizability. However, the reinforcement learning is not an end-to-end optimization method, which is difficult for the agent to converge and learn a robust policy. We propose

to learn the optimal actions in trajectories sampled from imperfect GBE policy to stabilize training while exploration. Different from other RL exploration methods, the proposed exploration method is based on the semantic graph, which is dynamically built during the navigation. Thus it helps the agent to learn a robust policy while navigating based on a graph.

To investigate the SOON task, we propose a large-scale From Anywhere to Object (FAO) benchmark. This benchmark is built on the Matterport3D simulator, which comprises 90 different housing environments with real image panoramas. FAO provides 4K sets of annotated instructions with 40K trajectories. As Fig. 1 (left) shows, one set of the instruction contains three sentences, including four levels of description: i) the color and shape of the object; ii) the surrounding objects along with the relationships between these objects and the target object; iii) the area in which the target object is located and the neighbour areas. Then, the average word number of the instructions is 38 (R2R is 26), and the average hop of the labeled trajectories is 9.6 (R2R is 6.0). Thus our dataset is more challenging than other tasks.

We present experimental analyses on both R2R and FAO datasets to validate the performance of the proposed GBE and the quality of FAO dataset. The proposed GBE significantly outperforms previous previous VLN methods without pretraining or auxiliary tasks on R2R and SOON tasks. We further provide human performance on the test set of FAO to quantify the human-machine gap. Moreover, by ablating vision and language modals with different granularity, we validate that our FAO dataset contains rich information that enables the agent to successfully locate the target.

Dataset	Instruction Context			Visual Context		Starting	Target
	Human	Content	Unamb.	Real-world	Temporal	Independent	Oriented
House3D [49]	✗	Room Name	✓	✗	Dynamic	✓	✗
MINOS [40]	✗	Ojbect Name	✓	✓	Dynamic	✓	✗
EQA [11], IQA [18]	✗	QA	✓	✗	Dynamic	✓	✗
MARCO [31], DRIF [5]	✓	Instruction	✓	✗	Dynamic	✗	✓
R2R [3]	✓	Instruction	✓	✓	Dynamic	✗	✓
TouchDown [10]	✓	Instruction	✓	✓	Dynamic	✗	✓
VLNA [37], HANNA [36]	✗	Dialog	✗	✓	Dynamic	✗	✓
TtW [13]	✓	Dialog	✓	✓	Dynamic	✗	✓
CVDN [45]	✓	Dialog	✗	✓	Dynamic	✗	✓
REVERIE [39]	✓	Instruction	✓	✓	Dynamic	✗	✓
FAO (Ours)	✓	Instruction	✓	✓	Dynamic	✓	✓

Table 1: Compared with existing datasets involving embodied vision and language tasks.

2. Related Work

Vision Language Navigation Navigation with vision-language information has attracted widespread attention, since it is both widely applicable and challenging. Anderson *et al.* [3] propose Room-to-Room (R2R) dataset, which is the first Vision-Language Navigation (VLN) benchmark combining real imagery [7] and natural language navigation instructions. In addition, the TOUCHDOWN dataset [10] with natural language instructions is proposed for street navigation. To address the VLN task, Fried *et al.* propose a speaker-follower framework [17] for data augmentation and reasoning in supervised learning, along with a concept named "panoramic action space" proposed to facilitate optimization. Wang *et al.* [47] demonstrate the benefit to combine imitation learning [6, 22] and reinforcement learning [34, 42]. Other methods [48, 29, 30, 44, 26, 24] have been proposed to solve the VLN tasks from various angles. Inspired by the success of VLN, many datasets based on natural language instructions or dialogues have been proposed. VLNA [37] and HANNA [36] are environments in which an agent receives assistance when it gets lost. TtW [13] and CVDN [45] provide dialogues created by communication between two people to reach the target position. Unlike the above methods, REVERIE [39] introduces a remote object localization task; in this task, an agent is required to find an object in another room that is unable to see at the beginning. The proposed SOON task is a coarse-to-fine navigation process, which navigates towards a target from anywhere following a complex scene description. An overall comparison between the SOON task and existing embodied navigation tasks is shown in Tab. 1.

Mapping and Planning Classical SLAM-based methods [46, 12, 19, 16, 21, 4] build a 3D map with LIDAR, depth or structure, and then plan navigation routes based on this map. Due to the development of photo-realistic environments [3, 10, 50] and efficient simulators [15, 40, 41], deep learning-based methods [35, 28, 53] have become feasible ways of training a navigation agent. Since deep learning methods have revealed their ability in feature engineering, end-to-end agents are becoming popular. Later works [16, 51, 33] adopt the idea of SLAM and introduce a memory mechanism, a method combining classical mapping methods and deep learning methods for generalization and long-trajectory navigation purposes. Recent works [9, 14, 8] model the navigation semantics in graphs and achieve great success in embodied navigation tasks. Different from previous work [14] that only trains the agent using labeled trajectories by imitation learning, our works introduce reinforcement learning in policy learning and propose a novel exploration method to learn a robust policy.

3. Scenario Oriented Object Navigation

Task Definition of SOON We propose a new Scenario Oriented Object Navigation (SOON) task, in which an agent navigates from an arbitrary position in a 3D embodied environment to localize a target object following an instruction. The task includes two sub-tasks: navigation and localization. We consider a navigation to be a *success* if the agent navigates to a position close to the target (<3m); and we consider the localization to be a *success* if the agent correctly locates the target object in the panoramic view based on the success of navigation. To ensure that the target object can be found regardless of the agent's starting point, the instruction consists of several parts: i) object attribute, ii) object relationship, iii) area description, and vi) neighbor area descriptions. An example to demonstrate different parts of description is shown in Fig. 2. In step t in navigation, the agent observes a panoramic view v_t, containing RGB and depth information. Meanwhile, the agent receives neighbour node observations $U_t = \{u_t^1, ..., u_t^k\}$, which are the observations of k reachable positions from the current position. All reachable positions in a house scan are discretized into a navigation graph, and the agent navigates between nodes in the graph. For each step, the agent takes an action a to move from the current position to a neighbor node or stop. In addition to RGB-D sensor, the simulator provides a GPS sensor to inform the

Step	Annotation Content
■ object name	stools
■ object attribute	The two black stools, a bar table with a white desktop and brown body.
■ object relationship	They are set under blue lights, behind a yellow sofa and in front of a table with a white desktop.
■ target area	The stools are in the barroom.
■ neighbor areas	The stools are set by the stairs and near a living room.
■ full instruction	In the barroom, there are two black stools set by the stairs. They are set in in front of a brown table with white desktop by the stairs and near a living room.

Figure 2: An example of annotating instructions in 6 steps.

Figure 3: Converting a 2D bounding box into Polar coordinate.

agent of its x, y coordinates. Also the simulator provides the indexes of the current node and candidate nodes.

Polar Representation REVERIE [39] annotates 2D bounding boxes in 2D views to represent the location of objects. The 2D views are separated from the panoramic views of the embodied simulator. This way of labeling has two disadvantages: 1) some object separated by 2D views is not labeled; 2) 2D image distortion introduces labeling noise. We adopt the idea of *Point Detection* [38, 54] and represent the location by polar coordinates, as shown in Fig. 3. First, we annotate the object bounding box with four vertices $\{p_1, p_2, p_3, p_4\}$. Then, we calculate the center point by p_c. After that, we convert the 2D coordinates into an angle difference between the original camera ray α and the adjusted camera ray α'.

4. Graph-based Semantic Exploration

We present the Graph-based Semantic Exploration (GBE) method in this section. The pipeline of the GBE is shown in Fig. 4. Our vision encoder g and language encoder h are built on a common practice of vision language navigation [47, 44, 52]. Subsequently, we introduce the graph planner in GBE, which models the structured semantics of visited places. Finally, we introduce our exploration method based on the graph planner.

Graph-based Navigation Memorizing viewed scenes and explicitly model the navigation environment are helpful for long-term navigation. Thus, we introduce a graph planner to memorize the observed features and model the explored areas as a feature graph. The graph planner maintains a node feature set $\mathcal{V}$, an edge set $\mathcal{E}$ and a node embedding set $\mathcal{M}$. The node feature set $\mathcal{V}$ is used to store node features and candidate features generated from visual encoder g. The edge set $\mathcal{E}$ dynamically updated to represent the explored navigation graph. The embedding set $\mathcal{M}$ stores the intermediate node embeddings, which are updated by GCN [27]. The node features in $\mathcal{M}$, noted as $f_{n_i}^{\mathcal{M}}$, are initialized by the feature of the same position in $\mathcal{V}$. At step t, the agent navigates to a position whose index is d_0, and receives a visual observation v_t and the observations of neighbor nodes are
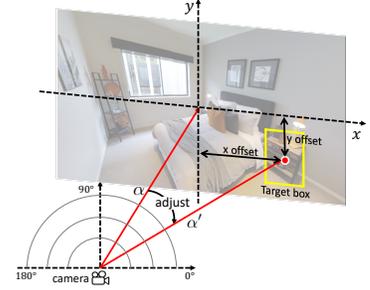
$U_t = \{u_t^1, ..., u_t^k\}$, where k is the number of the neighbors and $N_t = \{n_1, ..., n_k\}$ are node indexes of the neighbors. The visual observation and neighbor observations are embedded by the visual encoder g:

$$
\begin{aligned}
f_{n_0}^v &= g(v_t) \\
f_{n_i}^c &= g(u_t^i),
\end{aligned}
\tag{1}
$$

where n_0 stands for the current node, and $n_i (1 \le i \le n)$ are the node it connects with. The graph planners add the f_t^v and $f_t^{u,i}$ into $\mathcal{V}$:

$$
\mathcal{V} \leftarrow \mathcal{V} \cup \{f_{n_0}^v, f_{n_1}^u, ..., f_{n_k}^u\}.
\tag{2}
$$

For an arbitrary node n_i in the navigation graph, its node feature is represented by $\mathcal{V}$ following two rules: 1) if a node n_i is visited, its feature f_{n_i} is represented by $f_{n_i}^v$; 2) if a node n_i is not visited but only observed, its feature is represented by $f_{n_i}^u$; 3) since a navigable position is able to be observed from multiple different views, the unvisited node feature is represented by the average value of all observed features. The graph planner also updates the edge set $\mathcal{E}$ by:

$$
\mathcal{E} \leftarrow \mathcal{E} \cup \{(n_0, n_1), (n_0, n_2), ..., (n_0, n_k)\}.
\tag{3}
$$

An edge is represented by a tuple consists of two node indexes, indicating that two nodes are connected. Then, $\mathcal{M}$ is updated by GCN based on $\mathcal{V}$ and $\mathcal{E}$:

$$
\mathcal{M} \leftarrow \text{GCN}(\mathcal{M}, \mathcal{E}).
\tag{4}
$$

To obtain comprehensive understanding of the current position and nearby scene, we define the output of the graph planner as:

$$
f_t^g = \frac{1}{k+1} \sum_{i=0}^k f_{n_i}^{\mathcal{M}},
\tag{5}
$$

f_t^g and language feature f_t^l perform cross-modal matching [47] and output $\tilde{f}_t$. GBE uses the $\tilde{f}_t$ for two tasks: navigation action prediction and target object localization. The candidates to navigate are all observed but not visited nodes
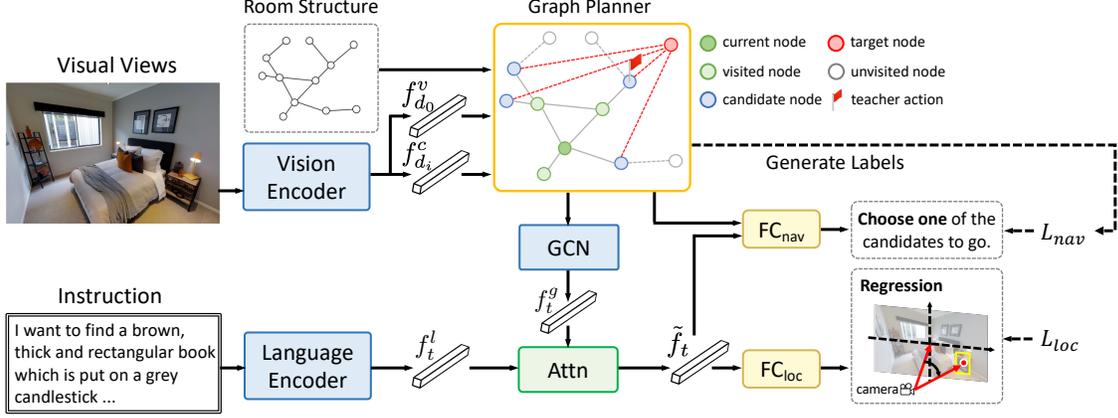
Figure 4: An overview of Graph-Based Semantic Exploration (GBE) model. Visual views are encoded by vision encoder and instructions are encoded by language encoder. The graph planner models the room semantics based on vision embeddings and the room structure information. GBE employs a GCN to embed graph nodes and output a graph embedding. Then, GBE outputs a cross-modal feature based on the graph embedding feature and language features. After that, GBE uses the cross-modal feature to predict the navigation action and regress the target location.

whose indexes are $C = \{c_1, ..., c_{|C|}\}$, where $|C|$ is the number of candidates. The candidate feature are extracted from $\mathcal{V}$, denoted as $\{f_{c_1}, ..., f_{c_{|C|}}\}$. The agent generates a probability distribution p_t over candidates for action prediction, and outputs regression results $\hat{l}_i^h$ and $\hat{l}_i^e$ standing for heading and elevation values for localization:

$$z_i = W_{nav}[\tilde{f}_t, f_{c_i}],$$
$$p_t(a_{c_i}) = \exp(z_i)/\sum_j \exp(z_j), \qquad (6)$$
$$[\hat{l}_i^h, \hat{l}_i^e] = W_{loc}\tilde{f}_t.$$

$0 \leq i \leq |C|$. z_i are logits generated by a fully connected layer whose parameter is W_{nav}. a_{c_0} indicates the stop action. Thus the action space $|\mathcal{A}| = |C| + 1$ is varied depending on the dynamically built graph.

Graph-based Exploration Seq2seq navigation models such as speaker-follower [17] only perceives the current observation and an encoding of the historical information. And existing exploration methods focus on data augmentation [44], heuristic-aided approach [30] and auxiliary task [52]. However, with the dynamically built semantic graph, the navigation agent is able to memorize all the nodes that it observes but has not visited. Thus we propose to use the semantic graph to facilitate exploration.

As shown in Fig. 4 (yellow box), the graph planner builds the navigation semantic graph during exploration. In imitation learning, the navigation agent uses the ground truth action a_t^* to sample the trajectory. However, in each step t, in graph-based exploration, the navigation action a_t is sampled from the predicted probability distribution of the candidates in Eq. 6. The graph planner calculate the Dijkstra distance from each candidate to the target. The teacher action $\hat{a}_t$ is to reach the candidate which is the closest to the target. Each trajectory in Room-to-room (R2R) dataset has only one target position. However, in the SOON task, since the target object could be able to be observed from multiple positions, trajectories could have multiple target positions. The teacher action $\hat{a}$ is calculated by:

$$\hat{a}_t = \underset{a_t^{n_i}}{\operatorname{argmin}} \left[\min\left(\mathrm{D}(c_i, n_{T_1}), ..., \mathrm{D}(c_i, n_{T_m})\right)\right], \quad (7)$$

where $n_{T_1}, ..., n_{T_m}$ are indexes of m targets, and the action from current position to node n_i is defined by $a_t^{n_i}$. $\mathrm{D}(n_i, n_j)$ stands for the function that calculates the Dijkstra distance between node n_i and n_j. Note that the target positions are visible in training to calculate the teacher action but not visible in testing. If the current position is one of target nodes, the teacher actions $\hat{a}_t$ is a stop action. Sampling and executing action a from imperfect navigation policy enables the agent to explore in the room. Using the optimal action $\hat{a}_t$ helps to learn a robust policy.

Training Objectives We here introduce two objectives in training: i) the navigation objective L_{nav}; ii) the object localization objective L_{loc}. The GBE model is jointly optimized by these two objectives. In imitation learning, our navigation agent learns from the ground truth action a^*. In reinforcement learning, the agent learns to navigate by maximizing the discounted reward when taking action a_t [43]. In graph-based exploration, we calculate the candidate which is closest to the target by the graph planner and set the action to move to the candidate as $\hat{a}_t$. The L_{nav} is the combination of the above three learning approaches:

$$L_{nav} = -\lambda_1 \sum_{\tau_1} \sum_t a_t^* \log(p_t)$$
$$-\lambda_2 \sum_{\tau_2} \sum_t a_t \log(p_t) A_t - \lambda_3 \sum_{\tau_3} \sum_t \hat{a}_t \log(p_t). \quad (8)$$
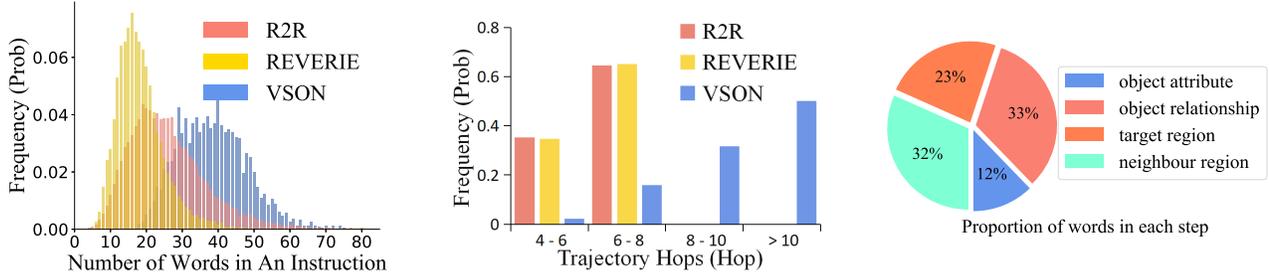
Figure 5: Statistical analysis across FAO

A_t is the advantage defined in A2C [34]. The reward of reinforcement learning is calculated by the Dijkstra distance between the current position and the target. The λ_1, λ_2, λ_3 are loss weights for imitation learning, reinforcement learning and graph-based exploration respectively. Our agent learns a localization branch that is supervised by the center position of the target. Since we map the 2D bounding box position into polar representation, the label consists of two linear values, namely heading l^h and elevation l^e. We use Mean Square Error (MSE) to optimize predictions:

$$L_{loc} = \frac{1}{N} \sum_{i=1}^{N} \left[(\hat{l_i^h} - l_i^h)^2 + (\hat{l_i^e} - l_i^e)^2 \right]. \quad (9)$$

5. Experiments

5.1. From Anywhere to Object (FAO) Dataset

We provide 3,848 sets of natural language instructions, describing the absolute location in a 3D environment. We further collect 6,326 bounding boxes for 3,923 objects across 90 Matterport scenes. Despite the fact that our task does not place limitations on the agent's starting position, we provide over 30K long distance trajectories in our dataset to validate the effectiveness of our task. Each instruction contains attributes, relationships and region descriptions to filter out the unique target object when there are multiple objects. Please refer to the supplementary materials for more details of our FAO dataset and experimental analysis.

Data Split The training split contains 3,085 sets of instructions with 28,015 trajectories over 38 houses. We propose a new split named *validation on seen instruction*, which is a validation set containing the same instructions in the same house with different starting positions. The validation seen instruction set contains 245 instructions with 1,225 trajectories. The validation set for seen houses with different instructions contains 195 instructions with 1,950 trajectories. The validation set for the unseen houses contains 205 instructions with 2,040 trajectories.

Data Collection We first label bounding boxes for objects in panoramic views. Then we convert the bounding box labels into polar representations as described in Sec. 3. Note

that the object can be reached from multiple positions. We annotate all these positions to reduce the dataset bias.

To collect diverse instructions with their hierarchical descriptions, we divide the language annotation task into five subtasks as shown in Fig. 2: 1) Describe the attributes, such as the color, size or shape, of the target; 2) Find at least two objects related to the target and describe their relationship; 3) Conduct explorations in the simulator to describe the region in which the target is located; 4) Explore and describe the nearby regions; 5) Rewrite all descriptions within three sentences. The first four steps ensure language complexity and diversity. And the rewriting step makes the language instruction coherent and natural.

Finally, we generate long navigation trajectories using the navigation graph of each scene. To make the task sufficiently challenging, we first set a threshold of 18 meters. For each instruction and object pair, we fix the target viewpoint and sample the starting viewpoint. We determine a trajectory as valid if the Dijkstra distance between the two viewpoints exceeds the threshold. In some houses, long trajectories are often difficult to find or may even not exist. Thus, we discount the threshold by a factor of 0.8 after every five sample failures.

Data Analysis Fig. 5 (left) illustrates the distributions of word numbers in the instructions. The FAO dataset contains 3,848 instructions with a vocabulary of 1,649 words. The average number of the words in an instruction set is 38.6, while which in REVERIE is 26.3 and in R2R is 18.3. Most of the instructions range from 20 words to 60 words, which ensures the power of representation. Moreover, the variance in instruction length makes the description more diverse. The trajectory length ranges from 15 meters to more than 60 meters. Compared with R2R and REVERIE that most of the trajectories are within 8 hops, as shown in Fig. 5 (middle), FAO provides much more long-term trajectories, which makes the dataset more challenging. Fig. 5 (right) illustrates the proportion of word numbers in the four instruction annotating steps. The more words are in the annotation, the richer information it contains. Therefore, we can infer that the object relationship and nearby regions contain the richest information. An agent should consequently pay more attention

Splits	Unseen House (Val)				Unseen House (Test)			
Metrics	NE ↓	OSR ↑	SR ↑	SPL ↑	NE ↓	OSR ↑	SR ↑	SPL ↑
Seq2Seq [3]	7.81	28.4	21.8	-	7.85	26.6	20.4	-
Ghost [2]	7.20	44	35	31	7.83	42	33	30
Speaker-Follower [17]	6.62	43.1	34.5	-	6.62	44.5	35.1	-
RCM [47]	5.88	51.9	42.5	-	6.12	49.5	43.0	38
Monitor* [29]	5.52	56	45	32	5.67	59	48	35
Regretful* [30]	5.32	59	50	41	5.69	56	48	40
EGP [14]	5.34	65	52	41	-	-	-	-
EGP* [14]	**4.83**	64	**56**	**44**	5.34	61	53	42
GBE (Ours)	5.20	**67.0**	53.9	43.4	**5.18**	**64.1**	**53.0**	**43.4**

Table 2: The results of the GMSE and previous state-of-the-art methods on R2R (*: model uses additional synthetic data).

to these two parts in order to achieve good performance.

5.2. Experimental Results

Experiment Setup We evaluate the GBE model on R2R and FAO datasets. We split our dataset into five components: 1) training; 2) validation on seen instructions (on seen houses as well); 3) validation on seen houses but unseen instructions; 4) validation on unseen houses; and 5) testing. Compared with standard VLN benchmark [3], we add a new validation set in FAO, the validation on seen instructions, due to the task starting-independent.

We evaluate the performance from two aspects: navigation performance and localization performance. The navigation performance is evaluated via commonly used VLN metrics, including Navigation Error (NE), Success Rate (SR), Oracle Success Rate (OSR) and the Success Rate weighted by Path Length (SPL) [1]. The localization performance is evaluated by the success rate indicating whether the predicted direction is located in the bounding box. We combine the SPL and localization success to propose a success rate of finding weighted by path length (SFPL):

$$ \text{SFPL} = \frac{1}{N} \sum_{i=1}^{N} S_i^{nav} S_i^{loc} \frac{l_i^{nav}}{\max(l_i^{nav}, l_i^{gt})}, \qquad (10) $$

where S_i^{nav} and S_i^{loc} are indicators of whether the agent has successfully navigated to or localized the target, respectively. l_i^{nav} is the length of the navigation trajectory, while l_i^{gt} is the shortest distance between the ground truth target and the starting position.

Implementation Details We compare the proposed model with several baselines: 1) a random policy; 2) Speaker-Follower [17], an imitation learning method; 3) RCM [47], an imitation learning and reinforcement learning; 4) AuxRN [52], a model with auxiliary tasks; 5) the Hierarchical Memory Network. All five models employ the same vision language navigation backbone introduced in Sec. 4. The visual encoder g is implemented by a Resnet-101 [20] and the language encoder h is a combination of a word em-

bedding layer and an LSTM [23] layer. We train all models on the training split for 10K interactions to ensure that all models are sufficiently trained. The optimizer we use is RMSProp and the learning rate is 10^{-4}.

Results on R2R In Tab. 2, we compare the GBE model with state-of-the-art models without pretraining and auxiliary tasks. On the unseen house validation set, the GBE outperforms all models without using additional data. It outperforms EGP, other graph-based navigation method by 2.4% in SPL. On the test set, the GBE outperforms pervious models on all the evaluation metrics. It outperforms RCM, a seq2seq model with imitation learning with reinforcement learning by 5.4% in SPL.

Results on FAO The experimental results are presented in Tab. 3. The performances of the baseline models reveal some unique features of the FAO dataset. Firstly, the human performance largely outperforms all models. The existence of this human-machine gap suggests that current methods are not able to solve this new task. The random policy method performs poorly on all metrics, which reveals that our dataset is not biased. Moreover, Reinforced Cross-Modal Matching (RCM), a method combines imitation learning and reinforcement learning outperforms the pure imitation learning method (Speaker-follower) on the unseen house set. It indicates that reinforcement learning helps avoid overfitting in our dataset. Our experiment of the AuxRN shows that the auxiliary tasks work on R2R are not benefitial on FAO, which indicate the SOON is unique. We test the performance of the GBE and the GBE without graph-based exploration. We observe that with graph-exploration, the model obtain better generalization ability. The final model is 0.7% higher in oracle success rate, 0.5% higher in success rate, 1.5% higher in SPL and 0.6% higher in SFPL than which without graph-based exploration on the test set. We discover that models perform well on the seen instruction set but perform poorly on other two sets. Since the domain of the seen instruction set is close to the training set, it indicates that models fit the training data well but lack of generalizability.

Splits	Val Seen Instruction				Val Seen House				Unseen House (Test)			
Metrics	OSR	SR	SPL	SFPL	OSR	SR	SPL	SFPL	OSR	SR	SPL	SFPL
Human	-	-	-	-	-	-	-	-	91.4	90.4	59.2	51.1
Random	0.1	0.0	1.5	1.4	0.4	0.1	0.0	0.9	2.7	2.1	0.4	0.0
Speaker-Follower [17]	97.8	97.9	97.7	24.5	69.4	61.2	60.4	**9.1**	9.8	7.0	6.1	0.6
RCM [47]	89.1	84.0	82.6	10.9	72.7	62.4	60.9	7.8	12.4	7.4	6.2	0.7
AuxRN [52]	**98.7**	**98.4**	97.4	13.7	**78.5**	68.8	**67.3**	8.3	11.0	8.1	6.7	0.5
GBE w/o GE	91.8	89.5	88.3	24.2	73	62.5	60.8	6.7	18.8	11.4	8.7	0.8
GBE (Ours)	98.6	**98.4**	97.9	**44.2**	64.1	**76.3**	62.5	7.3	**19.5**	**11.9**	**10.2**	**1.4**

Table 3: The results for baselines and our model on two validation set and test set.

Models	vision	language	SR	SPL	SFPL
GBE	✗	✗	0.6	0.4	0.0
GBE	✓	✗	9.8	8.1	0.5
GBE	✗	✓	1.8	1.5	0.2
GBE	✓	✓	**11.9**	**10.2**	**1.4**

Table 4: Ablation of unimodal inputs.

Models	SR	SPL	SFPL
GBE+①	7.3	6.2	0.5
GBE+①+②	6.2	4.9	0.7
GBE+①+②+③	6.6	5.5	0.8
GBE+④	**11.9**	**10.2**	**1.4**

Table 5: Ablation of granularity levels.

Ablation study of FAO We ablate the FAO dataset from two aspects: 1) the effect of vision and language modalities and 2) the effect of different granularity levels. The ablation result of input modal is shown in Tab. 4. We observe that the model without vision and language input performs the worst. Thus it is impossible to finish SOON task without vision-language modalities. And the model with vision only performs better than the model with language only. We infer that the vision is more import than language in SOON task. Finally, we find that the model with vision and language performs the best, indicating that the two modalities are related and both modalities are important. Some objects like 'chair' exist in all houses while other objects like 'flower' do not commonly exist. The model learns prior knowledge to find common object in navigation without language.

The ablation result of granularity levels is shown in Tab. 5. We train the GBE with different annotation granularity levels: ① object names, ② object attributes and relationships, ③ region information, ④ rewritten instructions. Note that the model with object names (GBE+①) is equivalent to the *ObjectGoal* navigation. We find that the model trained in *ObjectGoal* setting performs worse than the models trained with more information. It has two reasons: 1) there are more than one objects belongs to the same class, and navigating with object name cause ambiguity; 2) navigating without scene and region makes the agent harder to find the final location. By comparing the first three experiments, we infer that the object name (①), object attributes and relationships (②) and region descriptions (③) all contribute to the SOON navigation. At last, we find that the model with rewritten instructions performs the best (0.6% higher in SFPL than GBE+①+②+③). We infer that a well developed natural language instruction facilitates the agent to comprehend.

6. Conclusion

In this paper, we have proposed a task named Scenario Oriented Object Navigation (SOON), in which an agent is instructed to find an object in a house from an arbitrary starting position. To accompany this, we have constructed a dataset named From Anywhere to Object (FAO) with 3K descriptive natural language instructions. To suggest a promising direction for approaching this task, we propose GBE, a model that explicitly models the explored areas as a feature graph, and introduces graph-based exploration approach to obtain a robust policy. Our model outperforms all previous state-of-the-art models on R2R and FAO datasets. We hope that the SOON task could help the community approach real-world navigation problems.

7. Acknowledgements

References

[1] Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir Roshan Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 7

[2] Peter Anderson, Ayush Shrivastava, Devi Parikh, Dhruv Batra, and Stefan Lee. Chasing ghosts: Instruction following as bayesian state tracking. In *Advances in Neural Information Processing Systems*, pages 371–381, 2019. 7

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018. 1, 2, 3, 7

[4] Gil Avraham, Yan Zuo, Thanuja Dharmasiri, and Tom Drummond. Empnet: Neural localisation and mapping using embedded memory points. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8119–8128, 2019. 3

[5] Valts Blukis, Dipendra Misra, Ross A Knepper, and Yoav Artzi. Mapping navigation instructions to continuous control actions with position-visitation prediction. *arXiv preprint arXiv:1811.04179*, 2018. 3

[6] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 3

[7] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676, 2017. 3

[8] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *arXiv preprint arXiv:2007.00643*, 2020. 3

[9] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations*, 2020. 2, 3

[10] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12538–12547, 2019. 3

[11] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–10, 2018. 1, 3

[12] Andrew J. Davison and David W. Murray. Mobile robot localisation using active vision. *european conference on computer vision*, pages 809–825, 1998. 3

[13] Harm de Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. Talk the walk: Navigating new york city through grounded dialogue. *arXiv preprint arXiv:1807.03367*, 2018. 3

[14] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. *arXiv preprint arXiv:2007.05655*, 2020. 2, 3, 7

[15] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *Conference on Robot Learning*, pages 1–16, 2017. 3

[16] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 538–547, 2019. 3

[17] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *NIPS 2018: The 32nd Annual Conference on Neural Information Processing Systems*, pages 3314–3325, 2018. 1, 2, 3, 5, 7, 8

[18] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4089–4098, 2018. 1, 3

[19] Saurabh Gupta, Varun Tolani, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. *International Journal of Computer Vision*, 128(5):1311–1330, 2020. 1, 3

[20] 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)*, pages 770–778, 2016. 7

[21] Joao F. Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8476–8484, 2018. 3

[22] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*, 2016. 3

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 7

[24] Haoshuo Huang, Vihan Jain, Harsh Mehta, Alexander Ku, Gabriel Magalhaes, Jason Baldridge, and Eugene Ie. Transferable representation learning in vision-and-language navigation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7403–7412, 2019. 3

[25] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *ViGIL@NeurIPS*, 2019. 2

[26] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6741–6749, 2019. 3

[27] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR (Poster)*, 2016. 4

[28] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(1):1334–1373, 2016. 3

[29] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *International Conference on Learning Representations (ICLR)*, 2019. 3, 7

[30] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6732–6740, 2019. 3, 5, 7

[31] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: connecting language, knowledge, and action in route instructions. In *AAAI'06 proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1475–1482, 2006. 3

[32] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. *arXiv preprint arXiv:2004.14973*, 2020. 1

[33] Lina Mezghani, Sainbayar Sukhbaatar, Arthur Szlam, Armand Joulin, and Piotr Bojanowski. Learning to visually navigate in photorealistic environments without any supervision. *arXiv preprint arXiv:2004.04954*, 2020. 1, 3

[34] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pages 1928–1937, 2016. 3, 6

[35] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 3

[36] Khanh Nguyen and Hal Daumé. Help, anna! vision-based navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *2019 Conference on Empirical Methods in Natural Language Processing*, pages 684–695, 2019. 3

[37] Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12527–12537, 2019. 3

[38] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 685–694, 2015. 4

[39] 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. *arXiv: Computer Vision and Pattern Recognition*, 2019. 3, 4

[40] Manolis Savva, Angel X. Chang, Alexey Dosovitskiy, Thomas A. Funkhouser, and Vladlen Koltun. Minos: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, 2017. 1, 3

[41] 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 International Conference on Computer Vision*, pages 9339–9347, 2019. 3

[42] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3

[43] Richard S Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, volume 12, pages 1057–1063, 1999. 5

[44] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2610–2621, 2019. 3, 4, 5

[45] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. *arXiv preprint arXiv:1907.04957*, 2019. 1, 2, 3

[46] Sebastian Thrun. *Probabilistic Robotics*. 2005. 3

[47] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2018. 1, 2, 3, 4, 7, 8

[48] Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. *arXiv preprint arXiv:1803.07729*, 2018. 3

[49] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. In *ICLR 2018 : International Conference on Learning Representations 2018*, 2018. 1, 3

[50] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018. 3

[51] Jingwei Zhang, Lei Tai, Joschka Boedecker, Wolfram Burgard, and Ming Liu. Neural slam: Learning to explore with external memory. *arXiv preprint arXiv:1706.09520*, 2017. 3

[52] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *CVPR*, 2020. 4, 5, 7, 8

[53] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3357–3364, 2017. 3

[54] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1859–1868, 2017. 4

[55] Yi Zhu, Fengda Zhu, Zhaohuan Zhan, Bingqian Lin, Jianbin Jiao, Xiaojun Chang, and Xiaodan Liang. Vision-dialog navigation by exploring cross-modal memory. In *CVPR*, 2020. 1