REVERIE: Remote Embodied Visual Referring Expression in Real Indoor Environments

Yuankai Qi[1,2] Qi Wu[1*] Peter Anderson[3†] Xin Wang[4] William Yang Wang[4] Chunhua Shen[1] Anton van den Hengel[1]
[1]Australia Centre for Robotic Vision, The University of Adelaide [2]Harbin Institute of Technology, Weihai
[3]Georgia Institute of Technology [4]University of California, Santa Barbara
qykshr@gmail.com {qi.wu01, chunhua.shen, anton.vandenhengel}@adelaide.edu.au
peter.anderson@gatech.edu {xwang, william}@cs.ucsb.edu

Abstract

One of the long-term challenges of robotics is to enable robots to interact with humans in the visual world via natural language, as humans are visual animals that communicate through language. Overcoming this challenge requires the ability to perform a wide variety of complex tasks in response to multifarious instructions from humans. In the hope that it might drive progress towards more flexible and powerful human interactions with robots, we propose a dataset of varied and complex robot tasks, described in natural language, in terms of objects visible in a large set of real images. Given an instruction, success requires navigating through a previously-unseen environment to identify an object. This represents a practical challenge, but one that closely reflects one of the core visual problems in robotics. Several state-of-the-art vision-and-language navigation, and referring-expression models are tested to verify the difficulty of this new task, but none of them show promising results because there are many fundamental differences between our task and previous ones. A novel Interactive Navigator-Pointer model is also proposed that provides a strong baseline on the task. The proposed model especially achieves the best performance on the unseen test split, but still leaves substantial room for improvement compared to the human performance. *Repository:* https://github.com/YuankaiQi/REVERIE

1. Introduction

You can ask a 10-year-old child to bring you a cushion, and there is a good chance that they will succeed (even in an unfamiliar environment), while the probability that a robot will achieve the same task is significantly lower. Children have a wealth of knowledge learned from similar environments that they can easily apply to such tasks in an unfamiliar environment, including the facts that cushions generally inhabit couches, that couches inhabit lounge rooms, and that lounge rooms are often connected to the rest of a building through hallways. Children are also able to interpret natural language instructions and associate them with the visual world. However, the fact that robots currently lack these capabilities dramatically limits their domain of application.

Therefore, to equip robots with such abilities and to advance real-world vision-and-language research, we intro-

Instruction: Bring me the bottom picture that is next to the top of stairs on level one.

Figure 1. REVERIE task: an agent is given a natural language instruction referring to a remote object (here in the red bounding box) in a photo-realistic 3D environment. The agent must navigate to an appropriate location and identify the object from multiple distracting candidates. The blue discs indicate nearby navigable viewpoints provided by the simulator.

*Corresponding author
†Now at Google

1. Fold the towel in the bathroom with the fishing theme.
2. Enter the bedroom with the letter E over the bed and turn the light switch off.
3. Go to the blue family room and bring the framed picture of a person on a horse at the top left corner above the TV.
4. Push in the bar chair, in the kitchen, by the oven.
5. Windex the mirror above the sink, in the bedroom with the large, stone fireplace.
6. Could you please dust the light above the toilet in the bathroom that is near the entry way?
7. At the top of the stairs, the first set of potted flowers in front of the stairs need to be dusted off.
8. To the right at the end of the hall, where the large blue table foot stool is, there is a mirror that needs to be wiped.
9. Go to the hallway area where there are three pictures side by side and get me the one on the right.
10. There is a bottle in the office alcove next to the piano. It is on the shelf above the sink on the extreme right. Please bring it here.

Table 1. Indicative instruction examples from the REVERIE dataset illustrating various interesting linguistic phenomena such as dangling modifiers (*e.g*. 1), spatial relations (*e.g*. 3), imperatives (*e.g*. 9), co-references (*e.g*. 10), *etc*. Note that the agent in our task is required to identify the referent object, but is not required to complete any manipulation tasks (such as folding the towel).

duce a new problem, which we refer to as *Remote Embodied Visual referring Expression in Real Indoor Environments — REVERIE*. An example of the REVERIE task is illustrated in Fig. 1. A robot spawns at a starting location and is given a natural language instruction that refers to a remote target object at another location within the same building. To carry out the task, the agent is required to navigate closer to the object and return a bounding box encompassing the target object specified by the instruction. It demands the robot to infer the probable location of the object using knowledge of the environments, and explicitly identify the object according to the language instruction.

In distinction to other embodied tasks such as Vision-and-Language Navigation (VLN) [1] and Embodied Question Answering (EQA) [6], REVERIE evaluates the success based on explicit object grounding rather than the point navigation in VLN or the question answering in EQA. This more clearly reflects the necessity of robots' capability of natural language understanding, visual navigation, and object grounding. More importantly, the concise instructions in REVERIE represent more practical tasks that humans would ask a robot to perform (see Tab. 1). Those high-level instructions fundamentally differ from the fine-grained visuomotor instructions in VLN, and would empower high-level reasoning and real-world applications. Moreover, compared to the task of Referring Expression (RefExp) [8, 13, 22, 27] that selects the desired object from a single image, REVERIE is far more challenging in the sense that the target object is not visible in the initial view and needs to be discovered by actively navigating in the environment. Hence, in REVERIE, there are at least an order of magnitude more object candidates to choose from.

We build the REVERIE dataset upon the Matterport3D Simulator [1, 3], which provides panoramas of all the navigable locations and the connectivity graph in a building. To provide object-level information of the environments, we have extended the simulator to incorporate object annotations, including labels and bounding boxes from Chang *et al*. [3]. The extended simulator can return bounding boxes in images of different viewpoints and angles, thus able to accommodate evaluation on every possible location. The REVERIE dataset comprises 10,318 panoramas within 86 buildings containing 4,140 target objects, and 21,702 crowd-sourced instructions with an average length of 18 words. Tab. 1 shows sample instructions from the dataset, which illustrate various linguistic phenomena, such as spatial relations, dangling modifiers, and coreferences, etc.

We investigate the difficulty of the REVERIE task by directly combining state-of-the-art (SoTA) navigation methods and referring expression methods, and none of them shows promising results. We then propose an Interactive Navigator-Pointer model serving as a strong baseline for the REVERIE task. We also provide the human performance of the REVERIE task to quantify the machine-human gap.

In summary, our main contributions are:

1. A new embodied vision-and-language problem, Remote Embodied Visual referring Expression in Real 3D Indoor Environments (REVERIE), where given a natural language instruction that represents a practical task to perform, an agent must navigate and identify a remote object in real indoor environments.
2. The first benchmark dataset for the REVERIE task, which contains large-scale human-annotated instructions and extends the Matterport3D Simulator [1] with additional object annotations.
3. A novel interactive navigator-pointer model that provides strong baselines for the REVERIE dataset under several evaluation metrics.

2. Related Work

Referring Expression Comprehension. The referring expression comprehension task requires an agent to localise an object in an image given a natural language expression. Recent work casts this task as looking for the object that can generate its paired expressions [12, 17, 31] or jointly embedding the image and expression for matching estimation [5, 11, 15, 20, 30].

Dataset	Language Context				Visual Context			Goal
	Human	Main Content	Unamb	Guidance Level	BBox	Real-world	Temporal	
EQA [6], IQA [10]	✗	QA-pair	✓	–	✗	✗	Dynamic	QA
MARCO [21], DRIF [2]	✓	Nav-Instruction	✓	Detailed	✗	✗	Dynamic	Navigation
R2R [1]	✓	Nav-Instruction	✓	Detailed	✗	✓	Dynamic	Navigation
TouchDown [4]	✓	Nav-Instruction	✓	Detailed	✗	✓	Dynamic	Navigation
VLNA [23], HANNA[24]	✗	Nav-Dialog	✗	High	✗	✓	Dynamic	Find Object
TtW [7]	✓	Nav-Dialog	✓	High	✗	✓	Dynamic	Navigation
CVDN [25]	✓	Nav-Dialog	✗	High	✗	✓	Dynamic	Find Room
ReferCOCO [31]	✓	RefExp	✓	–	✓	✓	Static	Localise Object
REVERIE	✓	Remote RefExp	✓	High	✓	✓	Dynamic	Localise Remote Object

Table 2. Compared to existing datasets involving embodied vision and language tasks. Symbol instruction: 'QA': 'Question-Answer', 'Unamb': 'Unambiguous', 'BBox': 'Bounding Box', 'Dynamic'/'Static': visual context temporally changed or not.

Different from referring expression, REVERIE introduces three new challenges: i) The refereed object is not visible in the initial scene and only can be accessed after navigating to the goal location. ii) In contrast to previous RefExp tasks that select the target object from a single image, object candidates in REVERIE come from panoramas of all possible viewpoints. iii) The objects in RefExp are normally captured from the front view, while in our setting, the visual appearances of objects may vary largely due to different observation angles and viewpoints.

Vision-and-Language Navigation. Vision-and-language navigation (VLN) is the task where an agent is to navigate to a goal location in a 3D simulator given detailed natural language instructions such as 'Turn right and go through the kitchen. Walk past the couches on the right and into the hallway on the left. Go straight until you get to a room that is to the left of the pictures of children on the wall. Turn left and go into the bathroom. Wait near the sink.' [1]. A range of VLN methods [9, 14, 18, 19, 28, 29] have been proposed to address this VLN task.

Although the proposed REVERIE task also requires an agent to navigate to a goal location, it differs from existing VLN tasks in two important aspects: i) The challenge is much more closely related to the overarching objective of enabling natural language robot tasking because the goal is to localise a target object specified in an instruction, not just a location. This removes the artificial constraint that the instruction is restricted solely to navigation, and reflects the reality of the fact that most objects can be seen from multiple viewpoints. ii) Our collected navigation instructions are semantic-level commands which better reflect the way humans communicate. They are thus closer to'the cold tap in the first bedroom on level two' rather than step by step navigation instructions such as 'go to the top of the stairs then turn left and walk along the hallway and stop at the first bedroom on your right'.

The most closely related challenge to that proposed here is that addressed in [23, 24, 25] whereby an agent must identify an object by requesting and interpreting natural language assistance. The instructions are of the form 'Find a mug', and the assumption is that there is an oracle following the agent around the environment willing to provide natural language assistance. The question is then whether the agent can effectively exploit the assistance provided by the omniscient oracle. REVERIE, in contrast, evaluates whether the agent can carry out a natural language instruction alone. Another closely related work is TOUCHDOWN [4], that requires an agent to find a location in an urban outdoor environment on the basis of detailed navigation instructions.

Embodied Question Answering. Embodied question answering (EQA) [6] requires an agent to answer a question about an object or a room in a synthetic environment. Gordon *et al*. [10] introduce an interactive version of the EQA task, where the agent may need to interact with the environment/objects to correctly answer questions. Our REVERIE task differs from previous works that only output a simple answer or a series of actions, as we ask the agent to output a bounding box around a target object. This is a more challenging but realistic setting because if we want a robot to carry out a task that relates to an object, we need its precise location. Tab. 2 displays the difference between our task and other related embodied vision-language tasks.

3. The REVERIE Dataset

We now describe the REVERIE task and dataset, including the task definition, evaluation metrics, simulator, data collection policy, and analysis of the collected instructions.

3.1. The REVERIE Task

As shown in Fig. 1, our REVERIE task requires an intelligent agent to correctly localise a remote target object (can not be observed at the starting location) specified by a concise high-level natural language instruction. Since the target object is in a different room from the starting one, the agent needs first to navigate to the goal location.

Formally, at the beginning of each episode, the agent is given as input a high-level natural language instruction $\mathcal{X} = \langle \boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_L \rangle$, where L is the length of the instruction and $\boldsymbol{w}_i$ is a single word token. Following the common practice in VLN, the agent has access to surrounding panoramic images $\mathcal{V}_0 = \{\boldsymbol{v}_{0,k}, k \in 1, \ldots, 36\}$ and navigable viewpoints from the current location, where $\boldsymbol{v}_{0,k}$ is

determined by the agent's states comprising a tuple of 3D position, heading and elevation $s_{0,k} = \langle p_0, \phi_{0,k}, \theta_{0,k} \rangle$ (3 elevation and 12 heading angles are used). Then the agent needs to make a sequence of actions $\langle \boldsymbol{a}_0, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_T \rangle$ to reach the goal location, where each action is choosing one of the navigable viewpoints or choosing the current viewpoint which means to stop. The action can also be a 'detecting' action that outputs the target object bounding-box refereed by the instruction. The agent can attempt to localise the target at any step, which is totally up to algorithm design. But we only allow the agent output once in each episode, which means the agent only can guess the answer once in a single run. If the agent 'thinks' it has localised the target object and decides to output it, it is required to output a bounding box or choose from several candidates provided by the simulator. A bounding box is denoted as $\langle b_x, b_y, b_w, b_h \rangle$, where b_x and b_y are the coordinate of the left-top point, b_w and b_h denote the width and height of the bounding box, respectively. The episode ends after the agent outputs the target bounding box.

3.2. Evaluation Metrics

The performance of a model is mainly measured by Remote Grounding Success rate (RGS), which is the number of successful tasks over the total number of tasks. A task is considered successful if it selects the correct bounding box of the target object from a set of candidates (or the IoU between the predicted bounding box and the ground-truth bounding box ≥ 0.5, when candidate objects bounding boxes are not given). Because the target object can be observed at different viewpoints or camera views, we treat it as a success as long as the agent can identify the target within 3 meters, regardless of from different viewpoints or views. We also measure the navigation performance with four kinds of metrics, including success rate, oracle success rate, success rate weighted by path length (SPL), and path length (in meters) [1]. Please note that in our task, a navigation is considered successful only when the agent stops at a location within 3 meters from the target object. More details can be found in supplementary materials.

3.3. The REVERIE Simulator

Our simulator is based on the Matterport3D Simulator [1], a large-scale interactive environment constructed from the Matterport3D dataset [3]. In the simulator, an embodied agent is able to virtually 'move' throughout each building by iteratively selecting adjacent nodes from the graph of panoramic viewpoints and adjusting the camera pose at each viewpoint. It returns a rendered colour image that captures the current view, as shown in Fig. 1.

Adding Object-level Annotations. Object bounding boxes are needed in our proposed task, which are either provided as object hypotheses or used to assess the agent's ability to localise the object that is referred to by a natural language instruction. The main challenge of adding the object bounding boxes into the simulator is that we need to handle the changes in visibility and coordinate of 2D bounding boxes as the camera moves or rotates.

Figure 2. Object bounding boxes (BBox) in our simulator. The BBox size and aspect ratio of the same object may change after the agent moves to another viewpoint or changes its camera view.

To address these issues, we calculate the overlap between bounding boxes and object depth in each view. If a bounding box is fully covered by another one and it has a larger depth, we treat it as an occluded case. Specifically, for each building the Matterport3D dataset [3] provides all the objects appearing in it with centre point position $\mathbf{c} = \langle c_x, c_y, c_z \rangle$, three axis directions $\mathbf{d}_i = \langle d_i^x, d_i^y, d_i^z \rangle, i \in \{1, 2, 3\}$, and three radii r_i, one for each axis direction. To correctly render objects in the web simulator, we first calculate the eight vertexes using $\mathbf{c}, \mathbf{d}_i$ and r_i. Then these vertexes are projected into the camera space by the camera pose provided by Matterport3D dataset. Both C++ and web simulators will be released with the code. Fig. 2 presents an example of projected bounding boxes. Note that the target object may be observed at multiple viewpoints in one room, but we expect a robot can reach the target in a short distance. Thus, we only preserve objects within three meters to a viewpoint. For each object, a class label and a bounding box are associated and we adjust the size and aspect-ratio accordingly as the viewpoint and camera angle change. In total, we obtain ∼28k object annotations.

3.4. Data Collection

Our goal is to collect high-level human daily commands that may be assigned to a home robot in future, such as 'Open the left window in the kitchen' or 'Go to my bedroom and bring me a pillow'. We develop an interactive 3D WebGL simulator to collect such instructions on Amazon Mechanical Turk (AMT). The web simulator first shows a path animation and then randomly highlights one object at the goal location for workers to provide instructions to find or operate with. There is no style limitation of the command as long as it can lead the robot to the target object. Assistant

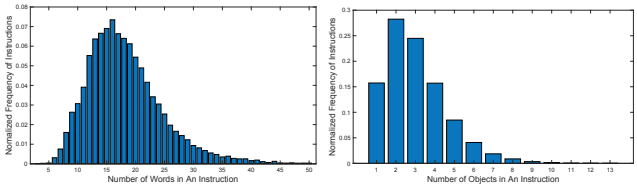

Figure 3. The distribution of the number of words (left) and objects (right) in each instruction.

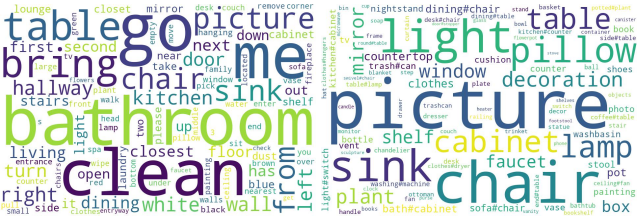

Figure 4. Word cloud of instructions (left) and target objects (right) in the REVERIE dataset. The bigger the font, the more percentage it occupies.

room and object information are provided to workers facilitating them to provide unambiguous instructions if there are similar rooms or objects. The workers can look around at the goal location to learn the environment. For each target object, we collect three referring expressions. The full collection interface (see in supplementary) is the result of several rounds of experimentation. Over 1,000 workers took part in the data collection, totally contributing around 2,648 hours of annotation time. Examples of the collected data can be found in Tab. 1, and more are in supplementary.

3.5. Dataset Analysis

The REVERIE dataset contains totally 21,702 instructions and a vocabulary of over 1,600 words. The average length of collected instructions is 18 words involving both navigation and referring expression information. Considering the detailed navigation instructions provided in the R2R [1] with an average length of 29 words and the previous largest dataset RefCOCOg [31] contains an average of 8 words, our instructions command is much more concise and natural, and thus more challenging.

Fig. 3 (left) displays the length distribution of the collected instructions, which shows that most instructions have $10 \sim 22$ words while the shortest annotation could be only 3 words, such as 'Flush the toilet'. Fig. 4 (left) presents the relative amount of words used in instructions in the form of word cloud. It shows that people prefer 'go' for navigation, and most instructions involve 'bathroom'. We also compute the number of mentioned objects in instructions and its distribution is presented in Fig. 3 (right). It shows that 56% instructions mention 3 or more objects, 28% instructions mention 2 objects, and the remaining 15% instructions mention 1 object. On average, there are 7 objects with 50 bounding boxes at each target viewpoint. There are 4,140 target objects in the dataset, falling into 489 categories, which are 6 times more than the 80 categories in ReferCOCO [31], a most popular referring expression dataset at present. Fig. 4 (right) shows the relative amount of target objects in different categories.

Data Splits We follow the same train/val/test split strategy as the R2R [1] datasets. The training set consists of 60 scenes and 10,466 instructions over 2,353 objects. The validation set including seen and unseen splits totally contains 56 scenes, 953 objects, and 4,944 instructions, of which 10 scenes and 3,521 instructions over 513 objects are reserved for val unseen split. For the test set, we collect 6,292 instructions involving 834 objects randomly scattered in 16 scenes. All the test data are unseen during training and validation procedures. The ground truth for the test set will not be released, and we will host an evaluation server where agent trajectories and detected bounding boxes can be uploaded for scoring.

4. The Interactive Navigator-Pointer Model

As our REVERIE task requires an agent to navigate to the goal location and point out the target object, a naive solution is to employ SoTA navigation (as a Navigator) and referring expression comprehension (as a Pointer) methods jointly. However, it is of great importance how the navigator and pointer should work together. Ideally, we want the navigator and pointer to benefit each other. Here, we propose a simple yet effective interactive manner that achieves the best performance in unseen environments as a strong baseline. As shown in Fig. 5, the proposed method takes the panoramic images observed at the current viewpoint and the instruction as input for the Pointer module. The Pointer computes matched objects in each view to the current attended instruction and outputs the visual features and labels of top-3 matched objects. Then, the Interaction module encodes these labels and outputs the concatenation of label embeddings and visual features of both object regions and the whole image. Finally, the Navigator module takes the combined feature and the instruction as input to determine the next action.

4.1. The Pointer Module

We use MAttNet [30] as our pointer because of its good performance. It decomposes an expression into three modular components related to subject appearance, location, and relationship to other objects via the attention mechanism $q^m = \sum_{j=1}^{L} a_{m,j} e_j$, where $m \in \{\text{subj}, \text{loc}, \text{rel}\}$, e_j is the embedding of each word in the instruction $\mathcal{X}$. $a_{m,j}$ is the attention on each word for each module.

Then three kinds of matching scores $S(o_i|q^m)$ are computed for each object o_i conditioned on each modular phrase embedding q^m. Specifically, $S(o_i|q^{\text{subj}}) = F(\tilde{v}_i^{\text{subj}}, q^{\text{subj}})$, $S(o_i|q^{\text{loc}}) = F(\tilde{l}_i^{\text{loc}}, q^{\text{loc}})$,

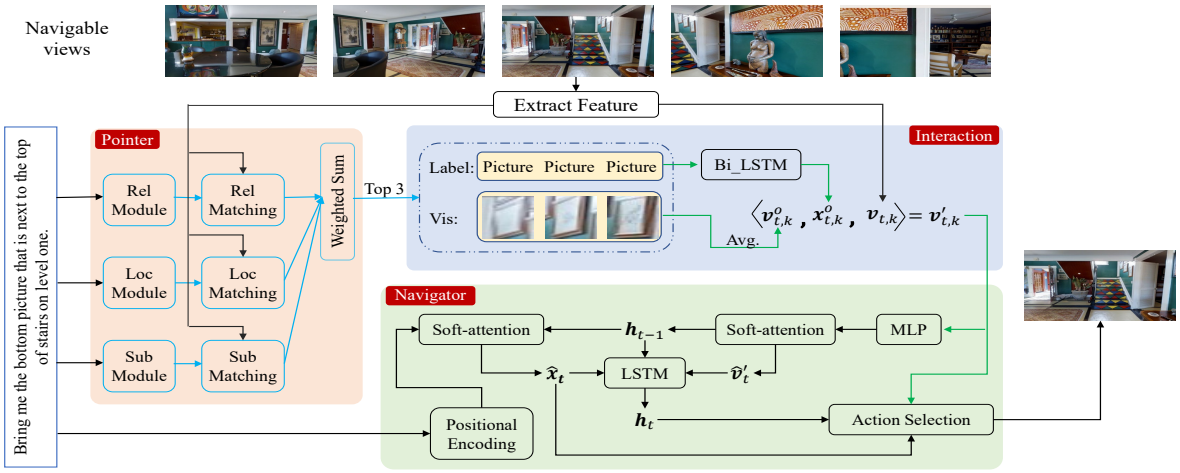

Figure 5. Our Interactive Navigator-Pointer Model

and $S(o_i|q^{\text{rel}}) = \max_{j \neq i} F(\tilde{v}_{ij}^{\text{rel}}, q^{\text{rel}})$, where $F(\cdot)$ is a two-layer MLP, $\tilde{v}_i^{\text{subj}}$ is a 'in-box' attended feature for each object using a 14×14 grid. $\tilde{l}_i^{\text{loc}}$ is the location representation of object o_i obtained by a fully-connected layer taking as input the relative position offset and area ratio to its up to five surrounding objects of the same category. $\tilde{v}_{ij}^{\text{rel}}$ is the visual representation of the surrounding object o_j regardless of categories.

The final matching score of object o_i and the instruction $\mathcal{X}$ is a weighted sum:

$$S = \sum S(o_i|q^m) w_m, \tag{1}$$

where $w_m = \text{softmax}(W_m^L [h_0, h_L] + b_m)$.

4.2. The Interaction Module

Intuitively, we want the Navigator and Pointer to interact with each other so that both navigation and referring expression accuracy can be improved. For example, the Navigator can use the visual grounding information to decide when and where to stop, and the Pointer accuracy can be improved if the navigator can reach the correct target location. To this end, we propose an interaction module that can plug the Pointer's output into the Navigator. Specifically, we first perform referring expression comprehension using the above Pointer module to select the top-3 matching objects in each candidate view. Then we use a trainable bi-direction LSTM

$$\boldsymbol{x}_{t,k}^o = \text{bi\_LSTM}(\mathcal{X}_O) \tag{2}$$

to encode the category labels $\mathcal{X}_O = \{\text{Label}_{i \in top3}\}$ of these selected objects as the textual representation for the k-th candidate viewpoint. In addition, the averaged output of ResNet FC7 layer of these object regions is used as the visual representation $\boldsymbol{v}_{t,k}^o$. Finally, we update the candidate viewpoint feature using concatenation

$$\boldsymbol{v}'_{t,k} = [\boldsymbol{v}_{t,k}, \boldsymbol{x}_{t,k}^o, \boldsymbol{v}_{t,k}^o] \tag{3}$$

which is send to the navigator (see Equ. 5 and 7). The pointer in such an interaction serves as hard attention for each candidate viewpoint, which highlights the most target-related objects for the navigator to take into account.

4.3. The Navigator Module

The backbone of our navigator module is a 'short' version of FAST [14], which uses a sequence-to-sequence LSTM architecture with an attention mechanism and a backtracking mechanism to increase the action accuracy. Specifically, let $\boldsymbol{X} \in \mathbb{R}^{L \times 512}$ denote instruction features obtained from $\mathcal{X}$ by an LSTM, and $\boldsymbol{V'} = [\boldsymbol{v}'_{t,1}; \ldots; \boldsymbol{v}'_{t,K}] \in \mathbb{R}^{K \times 4736}$ denote updated visual features obtained by our interactive module (Sec. 4.2) for panoramic images $\mathcal{V}_t$ at step t. FAST-short learns the local logit l_t signal, which is calculated by a visual and textual co-grounding model adopted from [19]. First, grounded text $\hat{\boldsymbol{x}}_t = \boldsymbol{\alpha}_t^\top \boldsymbol{X}$ and grounded visual $\hat{\boldsymbol{v}}'_t = \boldsymbol{\beta}_t^\top \boldsymbol{V'}$ are learned by

$$\boldsymbol{\alpha}_t = \text{softmax}(PE(\boldsymbol{X})(\boldsymbol{W}_x \boldsymbol{h}_{t-1})) \tag{4}$$

$$\boldsymbol{\beta}_t = \text{softmax}(g(\boldsymbol{V'})(\boldsymbol{W}_v \boldsymbol{h}_{t-1})) \tag{5}$$

where $\boldsymbol{\alpha}_t \in \mathbb{R}^{L \times 1}$ is textual attention weight, $\boldsymbol{\beta}_t \in \mathbb{R}^{K \times 1}$ is visual attention weight, $\boldsymbol{W}_x$ and $\boldsymbol{W}_v$ are learnable parameters, $PE(\cdot)$ is the positional encoding [26] that captures the relative position between each word within an instruction, $g(\cdot)$ is a one-layer Multi-Layer Perceptron (MLP), $\boldsymbol{h}_{t-1} \in \mathbb{R}^{512 \times 1}$ is previous encoder context. The new context is updated by an LSTM

$$(\boldsymbol{h}_t, \boldsymbol{c}_t) = \text{LSTM}([\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{v}}'_t, \boldsymbol{a}_{t-1}], (\boldsymbol{h}_{t-1}, \boldsymbol{c}_{t-1})) \tag{6}$$

taking as input the newly grounded text and visual features as well as previous action $\boldsymbol{a}_{t-1}$. Then the logit l_t can be

computed via an inner-product between each candidate's encoded context and instruction by

$$l_{t,k} = (\boldsymbol{W}_a[\boldsymbol{h}_t, \hat{\boldsymbol{x}}_t])^\top g(\boldsymbol{v}'_{t,k}) \tag{7}$$

where $\boldsymbol{W}_a$ is a learnable parameter matrix.

Based on logit l_t, FAST-short maintains one candidate queue and one ending queue. All navigable viewpoints (including the current viewpoint) at the current location are pushed into the candidate queue, but only the viewpoint with the largest accumulated logit $\sum_{\tau=0}^{t} l_\tau$ is popped out as the selected next step. Each passed viewpoint is pushed into the ending queue. One episode ends if the current viewpoint is selected or the candidate queue is empty or the maximum step is reached. Finally, the viewpoint with the largest accumulated logits is chosen as the actual stop location.

4.4. Loss Functions

Our final loss consists of two parts, the navigation loss L_{nav} and referring expression loss L_{exp}. The L_{nav} is a cross-entropy loss for action selection and a mean squared error loss for progress monitor:

$$L_{nav} = -\lambda_1 \sum_{t=1}^{T} y_t^a log(l_{t,k}) - \lambda_2 \sum_{t=1}^{T} (y_t^{pm} - p_t^{pm})^2 \tag{8}$$

where y_t^a is the ground truth action at step t, $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$ are weights balancing the two loss, $y_t^{pm} \in [0, 1]$ is the normalised distance in units of length from the current viewpoint to the goal, $p_t^{pm} = \tanh(\boldsymbol{W}_{pm}([\boldsymbol{\alpha}_t, \boldsymbol{h}_t^{pm}]))$ is the predicted progress and $\boldsymbol{h}_t^{pm} = \text{sigmoid}(\boldsymbol{W}_h([\boldsymbol{h}_{t-1}, \hat{\boldsymbol{v}}'_t]))$.

The referring expression loss L_{exp} is a ranking loss:

$$\begin{aligned} L_{exp} = \sum_i [&\lambda_3 \max(0, \delta + S(o_i|r_j) - S(o_i|r_i)) \\ &+ \lambda_4 \max(0, \delta + S(o_k|r_i) - S(o_i|r_i))] \end{aligned} \tag{9}$$

where $\lambda_3 = 1.0$, $\lambda_4 = 1.0$, (o_i, r_i) is a positive (object, expression) pair, (o_i, r_j) and (o_k, r_i) are negative (object, expression) pairs, δ is the distance margin between positive and negative pairs. All the losses are summarised together:

$$L = L_{nav} + \lambda_5 L_{exp} \tag{10}$$

to train our Interactive Navigator-Pointer model. We set λ_5 to 1.0 by default.

5. Experiments

In this section, we first present the training details of our model, and then provide extensive evaluation and analysis.

5.1. Implementation Details

The simulator image resolution is set to 640×480 pixels with a vertical field of view of 60 degrees. For each instruction in the train split, images and object bounding boxes at the goal viewpoint (for the views where the target object is visible) are organised following the format as in MAttNet for Pointer training. With the trained Pointer, assistant object information is provided as described in Section 4.2 to train the Navigator.

5.2. REVERIE Experimental Results

We first evaluate several baseline models and SoTA navigation models, combined with the our Pointer, *i.e.*, MAttNet. After the navigation models decide to stop, the Pointer is used to predict target object. In addition, we also test human performance (see details in the supplementary). Below is a brief introduction of the evaluated baseline and SoTA models. There are four baseline models:

- **Random** exploits the characteristics of the REVERIE dataset by randomly choosing a path with random steps (maximum 10) and then randomly choose an object as the predicted target.
- **Shortest** always follows the shortest path to the goal.
- **R2R-TF and R2R-SF [1]** are the first batch of navigation baselines. The difference between R2R-TF and R2R-SF is that R2R-TF is trained with the ground truth action at each step (Teacher-Forcing, TF) while R2R-SF adopts an action sampled from the predicted probability over its action space (Student-Forcing, SF).

The evaluated four SoTA navigation models are:

- **SelfMonitor [19]** uses a visual-textual co-grounding module to highlight the instruction for the next action and a progress monitor to reflect the progress.
- **RCM [28]** employs reinforcement learning to encourage global matching between instructions and trajectories, and performs cross-model grounding.
- **FAST-Short [14]** introduces backtracking into Self-Monitor.
- **FAST-Lan-Only** employs above FAST-Short model but we only use the language instruction as input. This model is used to check whether our task/dataset has a bias on language input.

Results. The detailed experimental results are presented in Tab. 3, of which the first four rows are results for baselines, the following four rows are for SoTA methods, and the last two rows are for our model and human performance.

According to the baseline section in Tab. 3, the Random model only achieves a RGS around 1%, which indicates the REVERIE task has a huge solution space. R2R-TF and R2R-SF [1] achieve good results on the Val Seen split but decrease a lot on the unseen splits. Student-Forcing is generally better than Teacher-Forcing. The Shortest model achieves the perfect performance because the ground-truth path to the goal is directly given.

In the second part, the best RGS rate is achieved by the combination of SoTA navigation (FAST) and referring

Methods	Val Seen					Val UnSeen					Test (Unseen)				
	Navigation Acc.				RGS	Navigation Acc.				RGS	Navigation Acc.				RGS
	Succ.	OSucc.	SPL	Length		Succ.	OSucc.	SPL	Length		Succ.	OSucc.	SPL	Length	
Random	2.74	8.92	1.91	11.99	1.97	1.76	11.93	1.01	10.76	0.96	2.30	8.88	1.44	10.34	1.18
Shortest	100	100	100	10.46	68.45	100	100	100	9.47	56.63	100	100	100	9.39	48.98
R2R-TF [1]	7.38	10.75	6.40	11.19	4.22	3.21	4.94	2.80	11.22	2.02	3.94	6.40	3.30	10.07	2.32
R2R-SF [1]	29.59	35.70	24.01	12.88	18.97	4.20	8.07	2.84	11.07	2.16	3.99	6.88	3.09	10.89	2.00
RCM [28]	23.33	29.44	21.82	10.70	16.23	9.29	14.23	6.97	11.98	4.89	7.84	11.68	6.67	10.60	3.67
SelfMonitor [19]	41.25	43.29	39.61	7.54	30.07	8.15	11.28	6.44	9.07	4.54	5.80	8.39	4.53	9.23	3.10
FAST-Short [14]	45.12	49.68	40.18	13.22	31.41	10.08	20.48	6.17	29.70	6.24	14.18	23.36	8.74	30.69	7.07
FAST-Lan-Only	8.36	23.61	3.67	49.43	5.97	9.37	29.76	3.65	45.03	5.00	8.15	28.45	2.88	46.19	4.34
Ours	**50.53**	**55.17**	**45.50**	16.35	**31.97**	**14.40**	**28.20**	**7.19**	45.28	**7.84**	**19.88**	**30.63**	**11.61**	39.05	**11.28**
Human	–	–	–	–	–	–	–	–	–	–	81.51	86.83	53.66	21.18	77.84

Table 3. Remote grounding success rate (RGS) achieved by combining SoTA navigation methods with the RefExp method MAttNet [30].

expression (MAttNet) models. However, the RGS rate is only 7.07% on the test split, falling far behind human performance 77.84%. The navigation-only accuracy of these SoTA navigation models indicates the challenge of our navigation task. Nearly 30% drops on the unseen splits are observed compared to the performance on previous R2R [1] task. For example, the navigation SPL score of FAST-Short [14] on Val UnSeen split drops from 43% on the R2R dataset to 6.17% on REVERIE.

To test whether our dataset has strong language bias, *i.e.*, whether a language-only model can achieve good performance, we implement a FAST-Lan-Only model with only instructions as its input. We observe a big drop on both seen and unseen splits, which suggests jointly considering language and visual information is necessary to our task.

Overall, these results show that a simple combination of SoTA navigation and referring expression methods would not necessarily lead to the promising performance as failures from either the navigator or the pointer would decrease the overall success. In this paper, we make the first attempt to enable the navigator and pointer to work interactively as described in Sec. 4.2. The results in Tab. 3 show that our method achieves consistently better results than non-interactive ones. The FAST-Short can be treated as our ablated model without our proposed interaction module. Our method achieves a gain of $\sim 4\%$ on the test split.

Referring Expression-Only. We also report the Referring Expression-Only performance. In this setting, the agent is placed at the ground-truth target location, and then referring expression comprehension models are tested.

We test the SoTA models such as MattNet [30] and CM-Erase [16] as well as a simple CNN-RNN baseline model with triplet ranking loss. Tab. 4 presents the results with human performance. It shows that the SoTA models achieve around 50% accuracy on the test split[1] which are far more better than the results when jointly considering the navigation and referring expression shown in Tab. 3. Even though, there is still a 40% gap to human performance, suggesting that our proposed REVERIE task is challenging.

[1]These SoTA models achieve 80% accuracy on ReferCOCO [31], a golden benchmark for referring expression.

	Val Seen	Val UnSeen	Test
Baseline	30.69	18.63	16.18
MAttNet [30]	68.45	56.63	48.98
CM-Erase [16]	65.21	54.02	45.25
Human	–	–	90.76

Table 4. Referring expression comprehension success rate ($\%$) at the ground truth goal viewpoint of our REVERIE dataset.

6. Conclusion

Enable human-robots collaboration is a long-term goal. In this paper, we make a step further towards this goal by proposing a Remote Embodied Visual referring Expression in Real Indoor Environments (REVERIE) task and dataset. The REVERIE is the first one to evaluate the capability of an agent to follow high-level natural languages instructions to navigate and identify the target object in previously unseen real images rendered buildings. We investigate several baselines and an interactive Navigator-Pointer agent model, of which the performance consistently demonstrate the significant necessity of further researches in this field.

We reach three main conclusions: First, REVERIE is interesting because existing vision and language methods can be easily plugged in. Second, the challenge of understanding and executing high-level instructions is significant. Finally, the combination of instruction navigation and referring expression comprehension is a challenging task due to the large gap to human performance.

7. Acknowledgements

We thank Philip Roberts, Zheng Liu, Zizheng Pan, and Sam Bahrami for their great help in building the dataset. Yuankai Qi is funded in part by NSFC 61902092 and HIT.NSRIF.2020005. Qi Wu is funded by DE190100539 and NSFC 61877038. The authors from UCSB are not supported by any of the projects above.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018.

[2] Valts Blukis, Dipendra Kumar Misra, Ross A. Knepper, and Yoav Artzi. Mapping navigation instructions to continuous control actions with position-visitation prediction. In *CoRL*, pages 505–518, 2018.

[3] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. In *3DV*, pages 667–676, 2017.

[4] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. TOUCHDOWN: natural language navigation and spatial reasoning in visual street environments. In *CVPR*, pages 12538–12547, 2019.

[5] Kan Chen, Rama Kovvuri, and Ram Nevatia. Query-guided regression network with context policy for phrase grounding. In *ICCV*, pages 824–832, 2017.

[6] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *CVPR*, pages 1–10, 2018.

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

[8] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *CVPR*, pages 7746–7755, 2018.

[9] 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 *NeurIPS*, pages 3318–3329, 2018.

[10] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. IQA: visual question answering in interactive environments. In *CVPR*, pages 4089–4098, 2018.

[11] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, pages 4418–4427, 2017.

[12] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *CVPR*, pages 4555–4564, 2016.

[13] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798, 2014.

[14] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha S. Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *CVPR*, pages 6741–6749, 2019.

[15] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *ICCV*, pages 4866–4874, 2017.

[16] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *CVPR*, pages 1950–1959, 2019.

[17] Ruotian Luo and Gregory Shakhnarovich. Comprehension-guided referring expressions. In *CVPR*, pages 3125–3134, 2017.

[18] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *CVPR*, pages 6732–6740, 2019.

[19] 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 *ICLR*, 2019.

[20] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. Multimodal convolutional neural networks for matching image and sentence. In *ICCV*, pages 2623–2631, 2015.

[21] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *AAAI*, pages 1475–1482, 2006.

[22] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016.

[23] Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *CVPR*, pages 12527–12537, 2019.

[24] Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *EMNLP-IJCNLP*, pages 684–695, 2019.

[25] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. *CoRR*, abs/1907.04957, 2019.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[27] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *CVPR*, pages 1960–1968, 2019.

[28] Xin Wang, Qiuyuan Huang, Asli Çelikyilmaz, 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 *CVPR*, pages 6629–6638, 2019.

[29] 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. In *ECCV*, pages 38–55, 2018.

[30] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L. Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, pages 1307–1315, 2018.

[31] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85, 2016.