Scene-Intuitive Agent for Remote Embodied Visual Grounding

Xiangru Lin[1] Guanbin Li[2*] Yizhou Yu[1,3]

[1]The University of Hong Kong [2]Sun Yat-sen University [3]Deepwise AI Lab

xrlin2@cs.hku.hk, liguanbin@mail.sysu.edu.cn, yizhouy@acm.org

Abstract

Humans learn from life events to form intuitions towards the understanding of visual environments and languages. Envision that you are instructed by a high-level instruction, "Go to the bathroom in the master bedroom and replace the blue towel on the left wall"*, what would you possibly do to carry out the task? Intuitively, we comprehend the semantics of the instruction to form an overview of* **where** *a bathroom is and* **what** *a blue towel is in mind; then, we navigate to the target location by consistently matching the bathroom appearance in mind with the current scene. In this paper, we present an agent that mimics such human behaviors. Specifically, we focus on the Remote Embodied Visual Referring Expression in Real Indoor Environments task, called REVERIE, where an agent is asked to correctly localize a remote target object specified by a concise high-level natural language instruction, and propose a two-stage training pipeline. In the first stage, we pretrain the agent with two cross-modal alignment sub-tasks, namely the Scene Grounding task and the Object Grounding task. The agent learns* **where** *to stop in the Scene Grounding task and* **what** *to attend to in the Object Grounding task respectively. Then, to generate action sequences, we propose a memory-augmented attentive action decoder to smoothly fuse the pre-trained vision and language representations with the agent's past memory experiences. Without bells and whistles, experimental results show that our method outperforms previous state-of-the-art(SOTA) significantly, demonstrating the effectiveness of our method.*

1. Introduction

Vision and Language tasks, such as Vision-and-Language Navigation (VLN) [2], Visual Question Answering (VQA) [3, 4] and Referring Expression Comprehension (REF) [17, 40, 39] etc., have been extensively studied in the wave of deep neural networks. In particular, VLN [2, 5] is a challenging task that combines both natural language understanding and visual navigation. Recent works have shown promising performance and progress. They mainly focus on designing agents capable of grounding fine-grained natural language instructions, where detailed information is provided, to find ***where*** to stop, for example *"Leave the bedroom and take a left. Take a left down the hallway and walk straight into the bathroom at the end of the hall. Stop in front of the sink"* [10, 23, 37, 36, 34, 18]. However, a practical issue is that fine-grained natural language instructions are not always available in real life and human-machine interactions are mostly based on high-level instructions such as *"Go to the bathroom at the end of the hallway"*. In other words, designing an agent that could perform high-level natural language interpretation and infer the probable target location using knowledge of the environments is of more practical use.

In this paper, we focus on the REVERIE task [30] which is an example of the above mentioned high-level instruction task. Here, we briefly introduce the settings. Given a high-level instruction that refers to a remote target object at a target location within a building, a robot agent spawns at a starting location in the same building and tries to navigate closer to the object. The output of the task is a bounding box encompassing the target object. The success of the task is evaluated based on explicit object grounding at the correct target location. A straightforward solution is to integrate SOTA navigation model with SOTA object grounding model. This strategy has proven to be inefficient in [30] and instead, they proposed an interactive module to enable the navigation model to work together with the object grounding model. Although the performance is improved, we observe that such method has a key weakness: it is unreasonable to discern high-level instruction by directly borrowing the fine-grained instruction navigation model that consists of simple trainable language attention mechanism based on the fact that the perception of high-level instruction primarily depends on commonsense knowledge prior as well as past experiences in memory. Therefore, the overall design

*Corresponding author is Guanbin Li. This work was supported in part by the National Key Research and Development Program of China (No.2020YFC2003902), in part by the Guangdong Basic and Applied Basic Research Foundation under Grant No.2020B1515020048, in part by the National Natural Science Foundation of China under Grant No.61976250 and No.U1811463. This work was also sponsored by CCF-Tencent Open Research Fund.

is not in line with human intuitions in high-level instruction navigation.

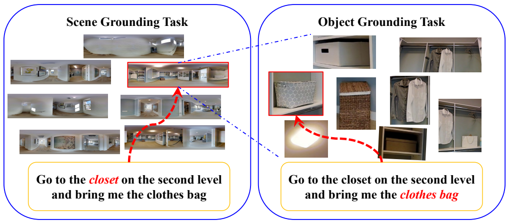

Figure 1. The overview of two pre-training tasks, the Scene Grounding task and the Object Grounding task. The Scene Grounding task empowers the agent the ability to reason where the target location is and the Object Grounding task learns what to attend to.

Designing an agent to solve the problem like the REVERIE task is still under explored and there are still no systematic ways to design such an agent. Then, how does human wisdom solve this task? Human beings have instincts to understand surrounding visual environments and languages. Intuitively, given a high-level instruction, we would first extract high-level ***what*** and ***where*** information and then form an overview of the appearance of the target location in mind based on common sense knowledge. During navigation, we would consistently match current scene and objects in the scene to the instruction semantics and decide where to navigate next. According to such intuitions, we approach this problem from a new perspective and present an agent that imitates such human behaviors. Concretely, we define our problem as designing an agent that is able to solve ***where*** and ***what*** problem in the REVERIE task. We propose a two-stage training pipeline. In the first stage, we design two pre-training tasks, mimicking the aforementioned two human intuitions. The second stage is training the agent with a memory-augmented attentive action decoder, further increasing the agent's navigation capability under high-level instructions.

Pre-training Stage. As is shown in Fig. 1, we introduce a new subtask called the Scene Grounding task that is trained to recognize which viewpoint in a set of viewpoints is best aligned with the high-level instruction and another subtask called the Object Grounding task that helps the agent identify the best object that matches to the instruction among a set of candidate objects located at a target viewpoint. Experimental results show that the Scene Grounding model recognizes the target viewpoint with a high accuracy and the Object Grounding model outperforms the previous best model used in [41, 30] by more than 10%.

Action Decoding Stage. In this stage, with the pre-trained models serving as scene and language encoders, we propose a memory-augmented attentive action decoder that leverages a scene memory structure as the agent's internal past state memory. This design is based on the fact that the computation of action at a specific time step could depend on any provided information in the past. Experimental results indicate that the proposed structure is effective and achieves new state-of-the-art performance.

To sum up, this paper has the following contributions:

- We propose a new framework that borrows human intuitions for designing agent capable of understanding high-level instructions, which closely integrate navigation and visual grounding in both training and inference. Specifically, the visual grounding models are pre-trained and serve as vision and language encoders for training navigation action decoder in the training phase. In inference, the action is predicted by considering logits from both the visual grounding models and the navigation decoder.
- We introduce two novel pre-training tasks, called Scene Grounding task and Object Grounding task, and a new Memory-augmented attentive action decoder in our framework. The pre-training tasks attempt to help the agent learn ***where*** to stop and ***what*** to attend to, and the action decoder effectively exploits past observations to fuse visual and textual modalities.
- Without bells and whistles, our method outperforms all previous methods, achieving new state-of-the-art performance on both seen and unseen environments on the REVERIE task.

2. Related Work

Vision-and-Language Navigation and REVERIE. In VLN, an agent is required to navigate to a goal location in a 3D simulator based on fine-grained instructions. [2] proposed the Matterport3D Simulator and designed the Room-to-Room task. Then, a lot of methods have been proposed to solve this task [10, 37, 36, 34, 18]. On the other hand, the recently proposed REVERIE task [30] is different from traditional VLN in that it requires an agent to navigate and localize target object simultaneously under the guidance of high-level instruction. The model they proposed trains the navigation model with the interactive module that works together with the object grounding model [41], in the hope that the model could learn to understand high-level instruction in a data-driven manner. However, our motivation is essentially different in that we inject commonsense knowledge prior and past memory experiences into the action policy taking into consideration the human perception in dealing with such high-level instruction navigation problems. Specifically, we introduce two pre-training tasks and a memory based action policy to make the agent become scene-intuitive. Moreover, our pre-training tasks differ from the ones proposed in [10, 42, 24] in that their motivation is based on the fact that the ground truth navigation path is actually hidden in the fine-grained instruction, which is not the case in high-level instruction navigation.

Memory-based policy for navigation tasks. Various memory models have been extensively studied for navigation agents, including unstructured memory [14, 26, 38, 15, 25, 7], addressable memory [27, 28], topological memory [31], and metric grid-based maps [11, 1], etc. Unstructured memory representations, such as LSTM memory, have been used extensively in both 2D and 3D environments. However, the issue of RNN based memory is that it does not contain context-dependent state feature storage or retrieval and does not have long time memory [1, 16, 9]. To address these limitations, more advaneced memory structures, such as addressable, topological, and metric based memory are proposed. In this paper, we adopt a simple adressable memory structure. The aim of using such a simple design is 1) to intentionally make it lightweight, thus reducing computational overhead, since the computational cost is important in REVERIE and our pipeline already contains heavy models; 2) to improve the performance of the overall pipeline rather than designing a more advanced memory superior to others. Besides, in VLN, the metric map memory construction requires finegrained language instruction as guidance, which is not available in our task, and building the topological memory requires pre-exploration of the environment, a technique that is certainly helpful to our agent but is beyond the discussion of this paper.

Vision-and-Language BERT based referring expression comprehension. Recent years have witnessed a resurgence of active research in transferrable image-text representation learning. BERT-based models [8, 33, 32, 22, 6, 21] have achieved superior performance over multiple vision-and-language tasks by transferring the pre-trained model on large aligned image-text pairs to other downstream tasks. In BERT-based VLN, the most related agents to ours are [12] and [24]. [12] treats VLN as a vision-and-language alignment task and utilizes a pre-trained vision-and-language BERT model to predict action sequence while [24] formulates VLN as an instruction and path alignment task and adopts a pre-trained vision-and-language BERT model to find the best candidate path that matches to the instruction given. However, our work differs from others in that we propose a generalized pipeline that mimics human intuitions to solve the high-level instruction navigation task where vision-and-language BERT model is a building block which can be customized to other vision-language alignment block. Experimental results show that the main performance gain comes from our proposed pipeline.

3. Method

In the REVERIE task, an agent placed at a starting location navigates to the target location to localize an object specified by a high-level instruction. To carry out this difficult task, we propose a novel pipeline that contains a scene grounding model, an object grounding model, and a memory-based action decoder. We make two claims of our design choice: first, to better grasp the semantics of high-level instructions, we choose ViLBERT model as our basic building block to serve as vision-and-language encoder; second, since scene grounding task and object grounding task are two essentially different tasks, we do not share the basic building blocks for these two tasks. In general, we decompose our method into two stages, as shown in Fig. 2, namely the pre-training stage and the action decoding stage. In the following sections, we first introduce the pre-training tasks; then we illustrate the memory-based attentive action decoder and finally, the loss function used to train the agent.

3.1. ViLBERT introduction

In this section, we briefly introduce the input and output arguments of a ViLBERT model [21] as shown in Fig. 3. A ViLBERT model is a BERT-based model that consists of two input streams, vision encoding stream and language encoding stream, followed by a cross-modal alignment Transformer block. The inputs to ViLBERT model are sequence of words and visual features respectively and the outputs are corresponding encoded word sequence features as well as visual sequence features. We use ViLBERT as our base model (basic building block) for the Scene Grounding task and the Object Grounding task. In Scene Grounding task, a panorama viewpoint image is discretized into 36 view images and the inputs are sequence of words in the instruction and 36 mean-pooled features extracted from 36 view images by a ResNet-152 CNN pre-trained on ImageNet [19]. In Object Grounding task, the inputs are sequence of words in the instruction and all annotated bounding boxes features extracted by Mask R-CNN [13] in a target viewpoint.

3.2. Overview of the proposed method

Settings. To formalize the task, we denote a given high-level instruction as $L = \{l_k\}_{k=1}^{N_l}$ where N_l is the number of words in the instruction L and a set of viewpoints as $\nu = \{V_k\}_{k=1}^{N_v}$ where N_v is the number of viewpoints in the environment. At each time step t, the agent observes a panoramic view V_t, a few navigable views O_t and a set of annotated bounding boxes B_t. The panoramic view is discretized into 36 single views by perspective projections, each of which is a 640×480 size image with field of view set to 60 degrees, and is denoted by $V_t = \{v_{t,i}\}_{i=1}^{36}$. $O_t = \{v_{t,i}\}_{i=1}^{N_o} \subseteq V_t$ where N_o is the maximum navigable directions at a viewpoint V_t. Each $v_{t,i}$ is represented as $\boldsymbol{v}_{t,i} = ResNet(v_{t,i})$. Thus, $\boldsymbol{V}_t = \{\boldsymbol{v}_{t,i}\}_{i=1}^{36}$. Besides, the set of annotated bounding boxes at viewpoint V_t is denoted by $B_t = \{b_{t,i}\}_{i=1}^{N_b}$ where N_b is the number of bounding boxes. Mask R-CNN [13] is used to extract bounding boxes features $\boldsymbol{B}_t = \{\boldsymbol{b}_{t,i}\}_{i=1}^{N_b}$, where $\boldsymbol{b}_{t,i} = MRCNN(b_{t,i})$.

Stage 1(a): Scene Grounding Task. We formulate the task as finding a viewpoint that best matches to a high-level instruction L in a set of candidate viewpoints ν_s.

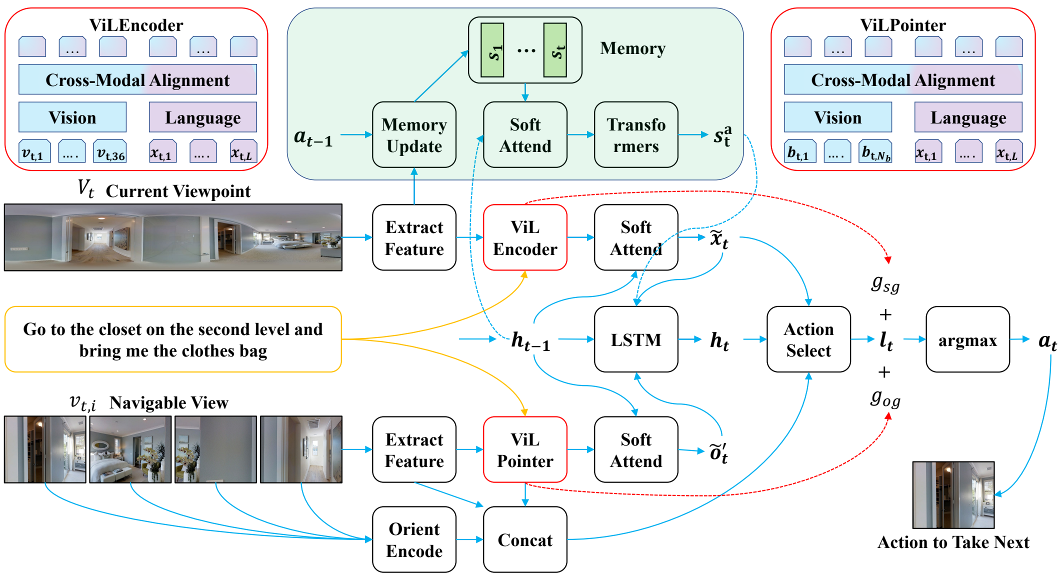

Figure 2. The overall pipeline of our method. The green part of the figure denotes the memory module where current viewpoint feature $\boldsymbol{V}_t$ and previous action feature $\boldsymbol{a}_{t-1}$ are embedded and stored in the Memory. $Transformer$ blocks are used to generate $\boldsymbol{s}_t^a$. The red rectangles represent two pre-trained models, namely Scene Grounding model and Object Grounding model. $ViLEncoder$ consists of $ViLBERT$ and $BiLSTM$ and $ViLPointer$ is $ViLBERT$ trained on viewpoint-based object grounding task. At each time step t, the agent perceives the instruction with viewpoint features and object features simultaneously. Action prediction is made by the Action Select part where an attentive structure is applied. The final action is generated by considering scene grounding score g_{sg}, object grounding score g_{og} and action logit $\boldsymbol{l}_t$. The dashed dot lines are used only for illustration purposes.

$\nu_s = \{V_k | V_k \in \nu\} \subseteq \nu$. Concretely, we define a mapping function $g_{sg}(,)$ that maps (L, V_k) to a matching score. The formula is defined as follows,

$$V_k^\star = \arg\max_{V_k \in \nu_s} g_{sg}(L, ResNet(V_k)) \tag{1}$$

Stage 1(b): Object Grounding Task. The goal of this task is to identify the best matching object among a set of candidate objects located at a target viewpoint. We denote V_T as a target viewpoint and its corresponding annotated bounding boxes set is B_T. We define another compatibility matching function $g_{og}(,)$ that produce matching scores for all objects with a high-level instruction L. Thus, the problem is defined as follows,

$$b_{T,i}^\star = \arg\max_{b_{T,i} \in B_T} g_{og}(L, MRCNN(b_{T,i})) \tag{2}$$

Stage 2: Memory-augmented action decoder. To mitigate the memory problem presented in previous section, a scene memory structure $\boldsymbol{M}_t$ is implemented to store the embedded observation and previous action at each time step t. The memory is updated by,

$$\begin{aligned} \tilde{\boldsymbol{v}}_t &= softmax(\boldsymbol{V}_t(\boldsymbol{W}_1\boldsymbol{h}_{t-1}))^T\boldsymbol{V}_t \\ \boldsymbol{s}_t &= FC([\boldsymbol{a}_{t-1}, \tilde{\boldsymbol{v}}_t]), \\ \boldsymbol{M}_t &= Update(\boldsymbol{M}_{t-1}, \boldsymbol{s}_t) \end{aligned} \tag{3}$$

where $\boldsymbol{s}_t$ is current state representation; $\tilde{\boldsymbol{v}}_t$ is attentive visual feature;$\boldsymbol{h}_{t-1}$ and $\boldsymbol{a}_{t-1}$ are last time step hidden state and action embedding respectively;$\boldsymbol{W}_1 \in \mathbb{R}^{2048 \times D_h}$ is a trainable parameter. FC stands for fully connected layer. The $Update$ operation appends $\boldsymbol{s}_t$ to $\boldsymbol{M}_t$. $\boldsymbol{V}_t \in \mathbb{R}^{36 \times 2048}, \tilde{\boldsymbol{v}}_t \in \mathbb{R}^{1 \times 2048}, \boldsymbol{a}_{t-1} \in \mathbb{R}^{1 \times 3200}, \boldsymbol{s}_t \in \mathbb{R}^{1 \times D_h}, \boldsymbol{h}_t \in \mathbb{R}^{D_h \times 1}, \boldsymbol{M}_t \in \mathbb{R}^{t \times D_h}$.

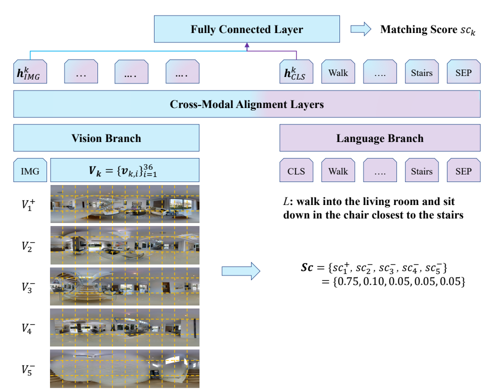

Figure 3. The pipeline of the Scene Grounding Task. We formulate this task as a 5-way multiple choice problem. Each $(L, ResNet(V_k))$ pair is sent to the $ViLBERT$ model separately to generate alignment score sc_k. The panorama viewpoint image here denotes the discretized 36 view images in a viewpoint. We mark the beginning of the image sequence with a special token IMG and the language with CLS.

3.3. Scene Grounding Task

The goal of this task it to help the agent infer where the target location is. Given a high-level instruction, *"Bring me* *the **jeans** that are hanging up in the **closet** to the right"*, humans first locate the ***where*** information, the key word ***closet***, by capturing the semantics of the instruction according to the language context and commonsense knowledge and then form an overview of the appearance of the ***closet*** in mind; then, humans navigate to the target location by consistently matching the ***closet*** appearance in mind with current scene. In fact, humans have gradually formed intuitions towards the understanding of scenes, instructions and tasks in life. For language instructions in relatively simple life scenes that do not involve complex reasoning, they usually directly merge the above two processes for direct perception and understanding. We call this process as context-driven scene perception. In this section, we propose Scene Grounding task to imitate such human behavior.

Based on the observation, we believe that a model that could evaluate the alignment between an instruction and a viewpoint is able to localize the target viewpoint. Therefore, to implement this idea, we create a dataset from the REVERIE training set and fine-tune a ViLBERT model on the dataset. Specifically, we adopt a 5-way multiple choice setting. We eliminate subscript for simplicity concern. Given an instruction L, we sample 5 viewpoints $\{V_1^+, V_2^-, V_3^-, V_4^-, V_5^-\}$, out of which only one is aligned to the instruction (or in other words, positive). In detail, we choose the ending viewpoint in the ground-truth training path as V_1^+, the second last viewpoint along the ground-truth path as V_2^- which is a hard negative sample and random sample V_3^-, V_4^- from the rest of the viewpoints along the path, and V_5^- from other path . Then, we run the ViLBERT model on each of the (L, V_k) pair. As is shown in Fig. 3, the output tokens CLS and IMG encode instruction representation $\boldsymbol{h}_{CLS}$ as well as viewpoint representation $\boldsymbol{h}_{IMG}$ respectively. We define the matching scores as $\boldsymbol{Sc}$ and train the model with cross entropy loss $\mathcal{L}_{sr}$.

$$\begin{aligned}
&\boldsymbol{Sc} = \{sc_1, sc_2, sc_3, sc_4, sc_5\} \\
&sc_k = g_{sg}(L, ResNet(V_k)) = \boldsymbol{W}_2(\boldsymbol{h}_{CLS}^k \odot \boldsymbol{h}_{IMG}^k) \\
&\mathcal{L}_{sr} = CELoss(softmax(\boldsymbol{Sc}), \mathbb{I}(V_1^+))
\end{aligned} \tag{4}$$

where $\boldsymbol{W}_2 \in \mathbb{R}^{1\times1024}$ is a trainable parameter and $\mathbb{I}(.)$ is indicator function. $\boldsymbol{h}_{CLS}^k \in \mathbb{R}^{1024\times1}, \boldsymbol{h}_{IMG}^k \in \mathbb{R}^{1024\times1}$ are the encoded language and visual representations of the language and vision encoding streams from our pre-trained ViLBERT model for kth (L, V_k) pair respectively.

3.4. Object Grounding Task

The aim of this task is to help the agent learn what to attend to. For each ground-truth target viewpoint V_T, we formulate this task as finding the best bounding box $b_{T,i}^\star$ in bounding boxes set B_T given (L, B_T) pair. A straightforward method to implement this idea is to construct a single image based grounding task, where each training sample consists of instruction L and a subset of bounding boxes in B_T that belong to view $v_{T,i}$. However, according to our experiment, this strategy produces moderate performance since objects in 3D space could span multiple views in corresponding projected 2D image space. The cross-image objects relationships in each viewpoint are not well captured by the model. Therefore, we propose a two-stage training strategy, namely a single image based grounding and a viewpoint based object grounding. In single image grounding, we fine-tune the ViLBERT model from [22, 21] on the aforementioned single image grounding dataset where each training sample is $(L, B_{v_{T,i}})$ (all annotated bounding boxes in $v_{T,i}$ are collected) and $B_{v_{T,i}} \subset B_T$; then, we further fine-tune trained model on a new viewpoint based object grounding dataset. Concretely, each training sample in the viewpoint based dataset is a (L, B_T) pair (all annotated bounding boxes in v_T are collected) and the corresponding label is a vector containing 0s and 1s where 1 indicates the IoU of a bounding box with the target bounding box is higher than 0.5. In inference, we represent an object score as the averaged scores from all bounding boxes that share the same object id at a viewpoint that the agent stops.

3.5. Action Decoder

With the pre-trained grounding models, the action decoder generally adopts Encoder-Decoder structure to produce action prediction. Specifically, the Scene Grounding model is accompanied by a $BiLSTM$ network to construct a vision and language grounding encoder $ViLEncoder$ and the Object Grounding model is formulated as an object level grounding encoder $ViLPointer$. The inputs to action decoder are L, $\boldsymbol{B}_t$ and $\boldsymbol{V}_t$ and it outputs predicted action distribution $\boldsymbol{l}_t$.

First. At each time step t, to perceive current scene and instruction, we obtain $\tilde{\boldsymbol{x}}_t$ by grounding L with $\boldsymbol{V}_t$ through $ViLEncoder$ and then selecting the fused language sequence as output. The formula is defined as follows,

$$\begin{aligned}
\boldsymbol{X}_t &= ViLEncoder(L, \boldsymbol{V}_t) \\
&= BiLSTM(ViLBERT(L, \boldsymbol{V}_t)) \\
\tilde{\boldsymbol{x}}_t &= softmax(\boldsymbol{X}_t(\boldsymbol{W}_3\boldsymbol{h}_{t-1}))^T \boldsymbol{X}_t
\end{aligned} \tag{5}$$

where $\boldsymbol{W}_3 \in \mathbb{R}^{1024\times D_h}$ is a trainable parameter and $\boldsymbol{X}_t$ is encoded language feature taking current scene $\boldsymbol{V}_t$ into consideration. $\boldsymbol{X}_t \in \mathbb{R}^{N_l\times1024}, \tilde{\boldsymbol{x}}_t \in \mathbb{R}^{1\times1024}, \boldsymbol{h}_{t-1} \in \mathbb{R}^{D_h\times1}$.

Second. To decide which navigable direction to go next, we perform object level referring expression comprehension. The object level referring comprehension helps the agent infer whether a navigable view $v_{t,i}$ contains possible target object. In particular, the set of bounding boxes in view $v_{t,i}$ is denoted by $\hat{B}_{t,i} = \{b_{t,k} | b_{t,k} \in B_t, Inside(b_{t,k}, v_{t,i}) = 1\}$ where

$Inside(,)$ function decides whether $b_{t,k}$ is inside view $v_{t,i}$. $ViLPointer$ is $ViLBERT$ pre-trained on the Object Grounding task and we select the fused bounding boxes features as the output. Then,

$$\begin{aligned} \boldsymbol{F}_{t,i} &= ViLPointer(L, MRCNN(\hat{B}_{t,i})) \\ \tilde{\boldsymbol{v}}_{t,i} &= g_{top-k}(\boldsymbol{F}_{t,i}) \end{aligned} \tag{6}$$

where $\boldsymbol{F}_{t,i}$ is the set of aligned bounding boxes features at view $v_{t,i}$ and $g_{top-k}(,)$ selects top-k aligned bounding boxes and averages the corresponding aligned bounding boxes features from $\boldsymbol{F}_{t,i}$ to produce view comprehension $\tilde{\boldsymbol{v}}_{t,i} \in \mathbb{R}^{1\times1024}$.

Third. We define the representation of each navigable view as $\boldsymbol{v}'_{t,i}$:

$$\boldsymbol{v}'_{t,i} = [\boldsymbol{v}_{t,i}, (\cos\theta_{t,i}, \sin\theta_{t,i}, \cos\phi_{t,i}, \sin\phi_{t,i}), \tilde{\boldsymbol{v}}_{t,i}] \tag{7}$$

where the agent's current orientation $(\theta_{t,i}, \phi_{t,i})$ represents the angles of heading and elevation and is tiled 32 times according to [10]. $(\cos\theta_{t,i}, \sin\theta_{t,i}, \cos\phi_{t,i}, \sin\phi_{t,i}) \in \mathbb{R}^{1\times128}$ and $\boldsymbol{v}'_{t,i} \in \mathbb{R}^{1\times3200}$. The set of navigable view representation is denoted as $\boldsymbol{O}'_t = \{\boldsymbol{v}'_{t,i}\}_{i=1}^{N_o}$. The grounded navigable visual representation $\tilde{\boldsymbol{o}}'_t$ is represented as follows:

$$\tilde{\boldsymbol{o}}'_t = softmax(g(\boldsymbol{O}'_t)(\boldsymbol{W}_4\boldsymbol{h}_{t-1}))^T g(\boldsymbol{O}'_t) \tag{8}$$

where $\boldsymbol{W}_4 \in \mathbb{R}^{1024\times D_h}$ is a trainable parameter and $g(,)$ is a number of Fully Connected layers accompanied by ReLU nonlinearities. $\tilde{\boldsymbol{o}}'_t \in \mathbb{R}^{1\times1024}, \boldsymbol{O}'_t \in \mathbb{R}^{N_o\times3200}$.

Fourth. The new context hidden state $\boldsymbol{h}_t$ is updated by a LSTM layer taking as input the grounded text $\tilde{\boldsymbol{x}}_t$ and navigable view features $\tilde{\boldsymbol{o}}'_t$ as well as the current state representation feature $\boldsymbol{s}^a_t$.

$$(\boldsymbol{h}_t, \boldsymbol{c}_t) = LSTM([\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{o}}'_t, \boldsymbol{s}^a_t], (\boldsymbol{h}_{t-1}, \boldsymbol{c}_{t-1})) \tag{9}$$

where $\boldsymbol{s}^a_t$ is memory augmented current state representation and is defined as,

$$\begin{aligned} \boldsymbol{M}^a_t &= [Transformer(\boldsymbol{M}_t, \boldsymbol{M}_t)]_{\times N_{mem}} \\ \boldsymbol{s}^a_t &= [Transformer(\boldsymbol{s}_t, \boldsymbol{M}^a_t)]_{\times N_{state}} \end{aligned} \tag{10}$$

where N_{mem} and N_{state} are number of memory transformer blocks used and number of state transformer blocks used respectively. $\boldsymbol{s}^a_t \in \mathbb{R}^{1\times D_h}, \boldsymbol{M}^a_t \in \mathbb{R}^{t\times D_h}$. $Transformer$ is the standard version Transformer block from [35].

Finally. The action logit $\boldsymbol{l}_t$ is computed in an attentive manner.

$$l_{t,i} = g(\boldsymbol{O}'_{t,i})(\boldsymbol{W}_5[\boldsymbol{h}_t, \tilde{\boldsymbol{x}}_t]) \tag{11}$$

where $\boldsymbol{W}_5 \in \mathbb{R}^{1024\times(1024+D_h)}$ is a trainable parameter and $\boldsymbol{l}_t \in \mathbb{R}^{N_o\times1}$. In training stage, $a_t = Categorical(\boldsymbol{l}_t)$ is selected based on categorical policy and in inference stage, it is selected by $a_t = \arg max(\boldsymbol{l}_t)$. Action embedding is selected based on $\boldsymbol{a}_t = \boldsymbol{O}'_t[a_t]$.

3.6. Inference

We propose to use a combined logit $\sum_{\tau=0}^{t} \boldsymbol{l}_\tau + g^\tau_{og} + g^\tau_{sg}$ that sums action logits, object grounding logits and scene grounding logits to perform navigation, where g^τ_{og} and g^τ_{sg} denote object grounding score and scene grounding score at time step τ respectively. Experimental results indicate that our strategy shortens the search trajectories while maintaining a good success rate. The final output bounding box is obtained by running $ViLPointer$ at the stop viewpoint that the agent predicts.

3.7. Loss Functions

To train the agent, we use a mixture of Imitation Learning (IL) and Reinforcement Learning (RL) to supervise the training. Specifically, In IL, at each time step, we allow the agent to learn to imitate the teacher action by using a cross entropy loss $\mathcal{L}_{ce}$ and a mean squared error loss $\mathcal{L}_{pm}$ for progress monitor [23]. In RL, we follow the idea of [34] and allow the agent to learn from rewards. If the agent stops within 3 meters near the target viewpoint, a positive reward $+3$ is assigned at the final step; otherwise a negative reward -3 is given.

$$\begin{aligned} \mathcal{L}_{final} &= \alpha\mathcal{L}_{ce} + \beta\mathcal{L}_{pm} + \gamma\mathcal{L}_{RL} \\ \mathcal{L}_{ce} &= -\sum_{t=1}^{T} y^\star_t \log(l_{t,\star}) \\ \mathcal{L}_{pm} &= -\sum_{t=1}^{T} (y^{pm}_t - p^{pm}_t)^2 \end{aligned} \tag{12}$$

where $y^\star_t$ is the teacher action at step t; $y^{pm}_t \in [0,1]$ is the shortest normalized distance from current viewpoint to the target viewpoint; p^{pm}_t is the predicted progress; α, β and γ are all set to 1.

4. Experiments

In the REVERIE dataset, the training set contains 59 scenes and 10466 instructions over 2353 objects; the val seen split consists of 53 scenes and 1371 instructions over 428 objects and the val unseen split include 10 scenes and 3573 instructions over 525 objects. The test set contains 16 scenes and 6292 instructions over 834 objects. In this section, we conduct extensive evaluation and analysis of the effectiveness of our proposed components. For the **Implementation Details** and the **Qualitative Examples**, please refer to the supplementary document.

4.1. Evaluation Metrics

Following [30], we evaluate the performance of the model based on REVERIE Success Rate (RGS) and REVERIE Success Rate weighted by Path Length (RG SPL). We also

Table 1. Ablation Study experiments performed to verify the effectiveness of the proposed method. In different ablation study block, the best performing result is marked in **bold**.

Experiments	ID	Methods								Val Seen						Val Unseen					
		Encoder				Pointer		Policy		Nav. Acc.				RGS↑	RG SPL↑	Nav. Acc.				RGS↑	RG SPL↑
		L_{enc}	$Bert_{enc}$	$ViLRaw_{enc}$	ViL_{enc}	MN_{ptr}	ViL_{ptr}	C_{pol}	MA_{pol}	Succ.↑	OSucc.↑	SPL↑	Length↓			Succ.↑	OSucc.↑	SPL↑	Length↓		
Component Effectiveness	1	√				√		√		50.53	55.17	45.50	16.35	31.97	29.66	14.40	28.20	7.19	45.28	7.84	4.67
	2		√			√		√		54.18	58.68	48.99	**12.46**	33.87	21.23	18.66	29.51	10.44	**32.95**	11.13	6.32
	3			√		√		√		33.73	39.14	30.72	14.56	23.82	21.94	15.22	31.64	8.44	42.62	8.89	4.84
	4			√			√	√		39.00	43.85	35.00	13.71	28.95	25.98	13.80	31.33	8.21	37.31	9.17	5.54
	5			√		√			√	37.32	43.08	31.71	18.29	24.88	21.70	19.06	44.39	7.10	79.88	11.08	4.17
	6				√	√		√		56.36	60.93	52.24	13.21	36.33	33.92	21.61	31.98	12.21	36.05	13.21	7.31
	7				√		√	√		54.25	56.08	50.49	13.56	39.56	37.16	26.98	37.86	13.70	42.50	17.32	8.71
	8				√		√		√	**59.52**	**64.23**	**55.30**	14.00	**43.57**	**40.42**	**28.17**	**40.41**	**14.77**	43.12	**19.60**	**10.27**
Memory Blocks (N_{mem}, N_{state})	9	(1, 1)								55.24	58.61	52.29	**12.42**	40.90	38.76	28.97	39.56	13.28	44.10	20.51	9.19
	10	(3, 3)								**61.91**	**65.85**	**57.08**	13.61	**45.96**	**42.65**	**31.53**	**44.67**	**16.28**	**41.53**	**22.41**	**11.56**
	11	(5, 5)								60.01	63.38	54.99	17.44	44.69	41.10	25.84	38.20	13.09	44.00	18.23	9.19
	12	(7, 7)								57.27	62.26	52.78	13.96	42.66	39.38	23.66	35.61	11.67	45.73	16.79	8.43
	13	(9, 9)								57.06	60.15	53.35	14.16	42.38	39.67	28.15	39.45	14.92	41.53	19.54	10.13
Logit Fusion	14	l_t								60.92	65.78	56.14	15.28	45.61	42.19	**32.35**	**49.08**	14.74	60.89	22.35	10.54
	15	l_t+g_{sg}								61.49	65.78	56.72	13.67	45.47	42.31	31.20	47.80	15.90	45.82	21.68	11.08
	16	l_t+g_{og}								61.14	65.77	55.21	16.82	44.48	40.04	32.12	46.54	15.73	52.14	21.98	11.02
	17	l_t+g_{sg}+g_{og}								**61.91**	**65.85**	**57.08**	**13.61**	**45.96**	**42.65**	31.53	44.67	**16.28**	**41.53**	**22.41**	**11.56**

Table 2. Comparison with state-of-the-art methods on the REVERIE task. The best performing result is marked in **bold**.

Methods	Val Seen						Val Unseen						Test (Unseen)					
	Nav. Succ.				RGS↑	RG SPL↑	Nav. Succ.				RGS↑	RG SPL↑	Nav. Succ.				RGS↑	RG SPL↑
	Succ.↑	OSucc.↑	SPL↑	Length↓			Succ.↑	OSucc.↑	SPL↑	Length↓			Succ.↑	OSucc.↑	SPL↑	Length↓		
RCM [36] + MattNet	23.33	29.44	21.82	10.70	16.23	15.36	9.29	14.23	6.97	11.98	4.89	3.89	7.84	11.68	6.67	10.60	3.67	3.14
SelfMonitor [23] + MattNet	41.25	43.29	39.61	**7.54**	30.07	28.98	8.15	11.28	6.44	**9.07**	4.54	3.61	5.80	8.39	4.53	**9.23**	3.10	2.39
FAST-short [18] + MattNet	45.12	49.68	40.18	13.22	31.41	28.11	10.08	20.48	6.17	29.70	6.24	3.97	14.18	23.36	8.74	30.69	7.07	4.52
REVERIE [30]	50.53	55.17	45.50	16.35	31.97	29.66	14.40	28.20	7.19	45.28	7.84	4.67	19.88	30.63	11.61	39.05	11.28	6.08
Human	-	-	-	-	-	-	-	-	-	-	-	-	81.51	86.83	53.66	21.18	77.84	51.44
Ours	**61.91**	**65.85**	**57.08**	13.61	**45.96**	**42.65**	**31.53**	**44.67**	**16.28**	41.53	**22.41**	**11.56**	**30.8**	**44.56**	**14.85**	48.61	**19.02**	**9.20**

report the performance of Navigation Success Rate, Navigation Oracle Success Rate, Navigation Success Rate weighted by Path Length (SPL), and Navigation Length. Please refer to the supplementary document for more details.

4.2. Ablation Study

In this section, we aim to answer the following questions: (a) Does the performance gain mainly come from BERT-based structure? (b) How effective is each of the proposed component? (c) Does the memory blocks number matter? (d) Why do we need logit fusion? For simplicity concern, we define the following experiment settings: (1) our proposed $ViLEncoder$ is ViL_{enc}; (2) the $ViLRaw_{enc}$ is $ViLEncoder$ not pre-trained on the Scene Grounding task but pre-trained on the Conceptual Captions dataset [29] as well as the 12 tasks specified in [22]; (3) the $BERT_{enc}$ is a BERT language encoder pre-trained on the BookCorpus [43] and English Wikipedia datasets; (4) our proposed $ViLPointer$ is ViL_{ptr}; (5) previous SOTA MattNet pointer is MN_{ptr}; (6) our action policy is MA_{pol}; (7) previous action policy is C_{pol}; (8) previous simple language encoder is L_{enc} composed of a trainable embedding layer with a Bidirectional LSTM layer.

Performance Gain. To answer question (a), we perform experiments 1, 2, 3 and 6 as is shown in Table 1. All agents are trained under $\mathcal{L}_{ce}$ and $\mathcal{L}_{pm}$ with α and β both set to 0.5. It is clear that the agent's overall performance is incrementally improved by changing the language encoder from the simple L_{enc} to our proposed ViL_{enc}, which proves our analysis that previous language encoder does not well capture the semantics of high-level instructions. The experimental results of 3 and 6 clearly suggests that the BERT-based structure is not the root cause of our performance gain and our proposed Scene Grounding task significantly increase the RG SPL metric to 33.9% on Val Seen and 7.31% on Val Unseen, even higher than the strong baseline in experiment 2.

Component Effectiveness. To answer question (b), based on the statistics from Table 1, we train six models in experiments from 3 to 8 and ablate the proposed component one by one to demonstrate the effectiveness. For fair comparison, we follow the settings of [30]. All agents are trained under $\mathcal{L}_{ce}$ and $\mathcal{L}_{pm}$ with α and β both set to 0.5. We start from the baseline experiment 3 and replace each component by our proposed ones. Specifically, in experiments 3 and 6, the proposed $ViLEncoder$ improves the RG SPL (and SPL) by a large margin, 11.98% (and 21.52%) higher in Val Seen and 2.47% (and 3.77%) higher in Val Unseen than the baseline respectively, which proves that the Scene Grounding task is effective; in experiments 3 and 4, our pointer $ViLPointer$ outperforms the MattNet counterpart by shortening the length of the search trajectory while maintaining a high RG SPL (and SPL), which demonstrates the effectiveness of the Object Grounding task; in experiments 3 and 5, the results show that the overall search trajectory of our action policy is longer than that of the baseline while our action policy achieves higher RGS and Navigation Success Rate, which demonstrates that the memory structure in our policy guides the agent to the correct target location at the cost of long trajectory; in experiments 7 and 8, we demonstrate that by integrating all our proposed methods, our agent improves previous SOTA in terms of RG SPL by 10.76% on Val Seen and 5.6% on Val Unseen.

Memory Blocks. To answer question (c), we train five models with different N_{mem} and N_{state} values. In these experiments, we train the agents with $\mathcal{L}_{ce}$, $\mathcal{L}_{pm}$ and $\mathcal{L}_{RL}$ and

Table 3. Pointer Task: REVERIE Success Rate at the ground truth target viewpoint; Encoder Task: given ground truth path, the success rate of identifying the target viewpoint among a set of candidate viewpoints along the path.

Tasks	Methods	Val Seen	Val Unseen
Pointer	MattNet [41]	68.45	56.63
	CM-Erase [20]	65.21	54.02
	ViLPointer-image-based	65.72	55.53
	ViLPointer-vp-based	**73.26**	**67.45**
Encoder	ViLEncoder	**85.67**	**66.43**

α, β and γ set to 1.0. In general, according to the experiments from 9 to 13 in Table 1, all pairs of (N_{mem}, N_{state}) exhibit superior performance compared to previous SOTA method in experiment 1 and the strong BERT baseline model in experiment 2. Moreover, the best performance model is achieved by setting (N_{mem}, N_{state}) to $(3, 3)$ in these five models, which suggests that using small values of (N_{mem}, N_{state}) limits the agent's memorization ability and using large values of (N_{mem}, N_{state}) enables the agent to achieve good performance on Val Unseen while maintains good performance on Val Seen.

Logit Fusion. To answer question (d), we report two accuracies to verify the effectiveness of g_{og} and g_{sg}. In the Encoder Task of Table 3, given ground-truth path, our proposed $ViLBERT$ model achieves competitive performance on both Val Seen and Val Unseen, demonstrating the strong ability of g_{sg} to identify a target viewpoint. In the Pointer Task of Table 3, the performance of $ViLPointer$-vp-based is significantly higher than previous image-based pointers because it is able to capture cross-image objects relationships, suggesting that g_{og} has the ability to find the target location if the target object exists. According to experiments from 14 to 17, where the agents are trained with $\mathcal{L}_{ce}$, $\mathcal{L}_{pm}$ and $\mathcal{L}_{RL}$ and α, β and γ set to 1.0, summing $\boldsymbol{l}_{\tau}$, g_{og}^{τ}, and g_{sg}^{τ} shortens the search trajectory and maintains a high RGS(Navigation Success Rate) and RG SPL(SPL). The motivation behind the summing strategy is to use model ensemble to reduce bias when searching for target locations considering the fact that the agent has no prior knowledge of the surrounding environments and the guidance of the high-level instructions is weak.

4.3. Compared to previous state-of-the-art results

We first show what kind of cases our method improves compared to previous SOTA and our BERT-based strong baseline in experiment 2. Specifically, we divide the shortest distance lengths of all ground-truth paths into three groups, namely short path(5 meters to 9 meters with 462 sample paths on Val Seen and 1400 sample paths on Val Unseen), middle path(9 meters to 14 meters with 703 sample paths on Val Seen and 1869 sample paths on Val Unseen), and long path(14 meters to 18 meters with 247 sample paths on Val Seen and 250 sample paths on Val Unseen). Then, we count the cases that the agent successfully navigates to

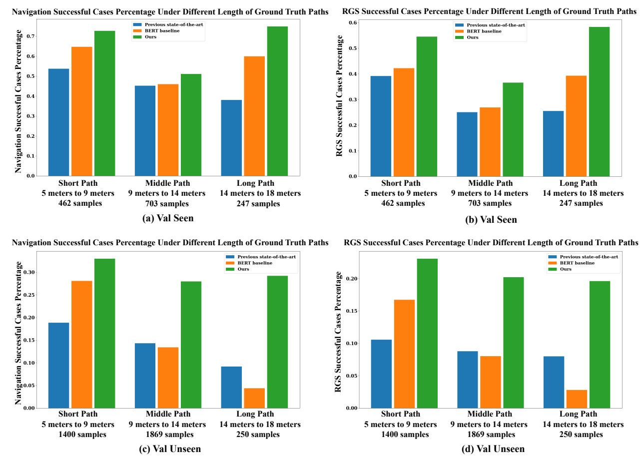

Figure 4. Percentage of successful Navigation and RGS cases under different length of ground-truth paths on Val Seen and Val Unseen datasets for previous state-of-the-art method, BERT baseline in experiment 2, and our method.

the target locations and the cases that the agent successfully navigates and localizes the target objects for the three groups. In Fig. 4, we report the corresponding successful cases percentage. It is obvious that our proposed method improves all kinds of sample paths by a clear margin.

Then, we compare our final model with previous SOTA models in Table 2. As is clearly shown in Table 2, our model outperforms all previous models by a large margin. Specifically, in terms of SPL, our agent increases previous SOTA by 11.58% on Val Seen, 9.09% on Val Unseen and 3.24% on Test respectively; for RG SPL, our agent increase previous SOTA by 12.99% on Val Seen, 6.89% on Val Unseen and 3.12% on Test. The overall improvements indicate that our proposed scene-intuitive agent not only navigates better but also localizes target objects more accurately.

5. Conclusion

In this paper, we present a scene-intuitive agent capable of understanding high-level instructions for the REVERIE task. Different from previous works, we propose two pre-training tasks, Scene Grounding task and Object Grounding task respectively, to help the agent learn ***where*** to navigate and ***what*** object to localize simultaneously. Moreover, the agent is trained with a Memory-augmented action decoder that fuses grounded textual representation and visual representation with memory augmented current state representation to generate action sequence. We extensively verify the effectiveness of our proposed components and experimental results demonstrate that our result outperforms previous methods significantly. Nevertheless, how to bridge the performance gap between seen and unseen environments and how to shorten the navigation length efficiently remains an open problem for further investigation.

References

[1] 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 (NeurIPS)*, 2019. 3

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[4] Qingxing Cao, Xiaodan Liang, Bailing Li, Guanbin Li, and Liang Lin. Visual question reasoning on general dependency tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7249–7257, 2018. 1

[5] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[6] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *The European Conference on Computer Vision (ECCV)*, 2020. 3

[7] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, D. Parikh, and Dhruv Batra. Embodied question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 3

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[9] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. 2019. 3

[10] 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 *Neural Information Processing Systems (NeurIPS)*, 2018. 1, 2, 6

[11] Saurabh Gupta, Varun Tolani, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. *International Journal of Computer Vision*, 128:1311–1330, 2019. 3

[12] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 3

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3

[15] Max Jaderberg, V. Mnih, W. Czarnecki, T. Schaul, Joel Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *ICLR*, 2017. 3

[16] Jia Lv Yanjie Duan Zhen Qin Guodong Li Jingyu Zhao, Feiqing Huang and Guangjian Tian. Do rnn and lstm have long memory? In *International Conference on Machine Learning*, 2020. 3

[17] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 1

[18] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 7

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[20] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1950–1959, 2019. 8

[21] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019. 3, 5

[22] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 5, 7

[23] 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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 1, 6, 7

[24] 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. 2020. 2, 3

[25] P. Mirowski, Razvan Pascanu, F. Viola, Hubert Soyer, Andy Ballard, Andrea Banino, Misha Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and Raia Hadsell. Learning to navigate in complex environments. In *ICLR*, 2017. 3

[26] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J. Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. Learning to navigate in complex environments. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 3

[27] Junhyuk Oh, Valliappa Chockalingam, Satinder Singh, and H. Lee. Control of memory, active perception, and action in minecraft. In *ICML*, 2016. 3

[28] Emilio Parisotto and R. Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. In *ICLR*, 2018. 3

[29] Sharma Piyush, Ding Nan, Goodman Sebastian, and Soricut Radu. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018. 7

[30] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 6, 7

[31] Nikolay Savinov, A. Dosovitskiy, and V. Koltun. Semi-parametric topological memory for navigation. In *ICLR*, 2018. 3

[32] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020. 3

[33] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019. 3

[34] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments:back translation with environmental dropout. In *Proceedings of The North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. 1, 2, 6

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008, 2017. 6

[36] 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, 2019. 1, 2, 7

[37] 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 *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2

[38] Daan Wierstra, A. Förster, Jan Peters, and J. Schmidhuber. Solving deep memory pomdps with recurrent policy gradients. In *ICANN*, 2007. 3

[39] Sibei Yang, Guanbin Li, and Yizhou Yu. Graph-structured referring expression reasoning in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9952–9961, 2020. 1

[40] Sibei Yang, Guanbin Li, and Yizhou Yu. Relationship-embedded representation learning for grounding referring expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[41] 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 *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 8

[42] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[43] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 7