

# March in Chat: Interactive Prompting for Remote Embodied Referring Expression

Yanyuan Qiao<sup>1</sup> Yuankai Qi<sup>1</sup> Zheng Yu<sup>1</sup> Jing Liu<sup>2 3</sup> Qi Wu<sup>1\*</sup>

<sup>1</sup>Australian Institute for Machine Learning, The University of Adelaide

<sup>2</sup>Institute of Automation, Chinese Academy of Sciences

<sup>3</sup>School of Artificial Intelligence, University of Chinese Academy of Sciences

{yanyuan.qiao, zheng.yu, qi.wu01}@adelaide.edu.au, qykshr@gmail.com, jliu@nlpr.ia.ac.cn

## Abstract

Many Vision-and-Language Navigation (VLN) tasks have been proposed in recent years, from room-based to object-based and indoor to outdoor. The REVERIE (Remote Embodied Referring Expression) is interesting since it only provides high-level instructions to the agent, which are closer to human commands in practice. Nevertheless, this poses more challenges than other VLN tasks since it requires agents to infer a navigation plan only based on a short instruction. Large Language Models (LLMs) show great potential in robot action planning by providing proper prompts. Still, this strategy has not been explored under the REVERIE settings. There are several new challenges. For example, the LLM should be environment-aware so that the navigation plan can be adjusted based on the current visual observation. Moreover, the LLM planned actions should be adaptable to the much larger and more complex REVERIE environment. This paper proposes a March-in-Chat (MiC) model that can talk to the LLM on the fly and plan dynamically based on a newly proposed Room-and-Object Aware Scene Perceiver (ROASP). Our MiC model outperforms the previous state-of-the-art by large margins by SPL and RGSPL metrics on the REVERIE benchmark. The source code is available at <https://github.com/YanyuanQiao/MiC>

## 1. Introduction

Vision-and-Language Navigation (VLN), which lies at the intersection of computer vision, natural language processing and robotics, has aroused great attention from research communities in the past few years. Given instructions in natural language, the VLN agent should navigate to the target location based on the dynamic observations in the 3D simulated environments. Since VLN has great

\*Corresponding author

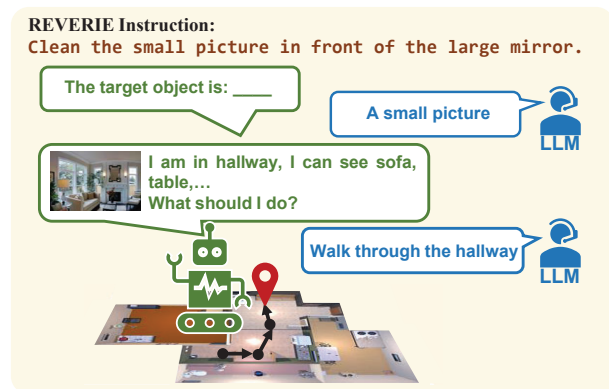


Figure 1: Our March-in-Chat (MiC) model is talking to a Large Language Model (LLM) to generate navigation plans on the fly, with the REVERIE instruction and the dynamic room-and-object information as inputs.

potential in real-world applications such as domestic assistant robots, a large amount of specific VLN tasks have been proposed, including R2R [4] and RxR [21] that ask the agent to navigate from one room to another in a photo-realistic environment according to the fine-grained instruction, NDH [36] provides detailed dialogues which imply the instruction, TouchDown [7] extends the task into an outdoor environment, REVERIE [27] and SOON [39] that additionally require the agent’s ability of remote object grounding and ALFRED [33] that asks the agent to interact with the target object in a single room of the synthetic environment.

Most of these VLN tasks provide detailed step-by-step instructions to the agent, such as “Go up the stairs and then walk the length of the couch. Walk past the dining area and into the kitchen. Stop in front of the refrigerator.” in R2R. Although detailed instructions can help the agent better achieve the navigation goal in the simulated environments, it has a big gap towards real applications where human beings tend to give coarse-grained high-level instructions such as “Go to the refrigerator on the second floor”.

Contrary to other tasks, the Remote Embodied Referring Expression (REVERIE) task is more likely to empower the real-world applications of VLN, of which the instructions are closer to those in practice, such as “Empty the washing machine on level one”. Such high-level instruction is more challenging for VLN agents since it requires them to be more competent in perceiving the surrounding environment and the navigation progress and correspondingly making reasonable plans for the next steps.

Recently, Large Language Models (LLMs) that internalize a wealth of commonsense knowledge show great potential in action planning for some embodied tasks with the help of suitable in-context learning. However, previous works mainly utilize LLMs to plan atomic actions of object manipulation in a very limited space with simple scenes. These predefined atomic actions can be easily planned well by the LLMs planners with a unified template. Different from these embodied tasks, REVERIE requires large-area exploration from one room to another, which is complex in the layout of rooms and scenes with diverse objects.

In this work, to adapt LLMs as the planner for REVERIE with the ability of comprehensive scene perception, we propose a novel model named *March in Chat* (MiC), which enables the LLM as an environment-aware instruction planner through on-the-fly dialogues between the agent and the LLM as Fig. 1 shows. Specifically, the agent is initially situated at the starting position given a high-level coarse-grained REVERIE instruction. First, a Goal-Oriented Static Planning (GOSP) module queries the LLM to point out the target object and infer where the thing may be by using the rich world knowledge internalized in the LLM. Secondly, the agent’s Room-and-Object Aware Scene Perceiver (ROASP) describes the current observation and asks the LLM to generate step-by-step fine-grained planning for the next navigation steps. Then, if the ROASP finds the room has changed, the LLM is queried again by the Scene-Oriented Dynamic Planning (SODP) module to generate a new fine-grained step-by-step planning, which will be concatenated with all previous responses from the LLM. The agent will march under the guidance of such interactive prompting until the task is finished.

To evaluate our proposed MiC, we conduct experiments on the REVERIE benchmark. Our MiC achieves a new state-of-the-art performance in all metrics on REVERIE val unseen set and REVERIE test unseen set. Mainly, MiC obtains 41.97% on the primary navigation metric of SPL and 26.17% on the major object grounding metric of RGSPL on test split, which is at least 3.09% and 3.49% higher than the previous SoTA results. We also conduct ablation studies to validate the contributions of different components in MiC and the effect of scene-aware perception in dynamic planning generation. These promising results demonstrate the effectiveness of our proposed MiC.

In summary, we make the following contributions:

- We propose a novel March-in-Chat (MiC) model, which lets the REVERIE agent talk with an LLM on the fly to make plans for the next few steps.
- Two planning modules, namely Goal-Oriented Static Planning (GOSP) module, and Scene-Oriented Dynamic Planning (SODP) module, and one Room-and-Object Aware Scene Perceiver (ROASP) module, are proposed.
- Extensive quantitative and qualitative experiments are conducted on REVERIE to validate the effectiveness of our method.

## 2. Related work

**Vision-and-Language Navigation** Vision-and-Language Navigation (VLN) has attracted increasing attention in recent years, and many specific VLN tasks have been proposed [7, 19, 24, 27, 29, 36]. Anderson *et al.* [4] proposes the first VLN benchmark, Room-to-Room (R2R), which requires an agent to navigate from one room to another in a house, according to a detailed natural language step-by-step instruction in a photo-realistic environment. Later, Room-across-Room (RxR) [21] was proposed with longer and more detailed multilingual instructions. Both these two tasks give fine-grained instructions, which makes it easier to navigate to the target location. NDH [36] extends the navigation instruction to the dialogue form, and TouchDown [7] extends the environments to outdoor. REVERIE [27] and SOON [39] are proposed for remote object localization, which requires an agent not only to navigate to the target location, but also to specify the object to interact with. The difference between REVERIE and SOON is that REVERIE uses short concise instructions (*e.g.*, “bring me the red cup from the kitchen.”) while SOON employs long detailed instructions (*e.g.*, “I want to find a cylindrical, metallic and tall lamp, which is set in the bright living room. The lamp is on the cabinet which is on the left of the television and next to the window. The living room is on ...” ).

Among the aforementioned VLN tasks, the instructions of REVERIE are closer to what we would say to an intelligent domestic robot in daily life in terms of the instruction length and logic, which is usually short and concise. However, most existing methods [2, 11, 14, 16, 26, 28] are usually designed for the VLN tasks where detailed step-by-step instructions are used, thus they do not perform well on REVERIE. In this work, we specifically develop a method for REVERIE. Inspired by the fact of LLMs that implicitly internalize rich knowledge in action planning, we propose to exploit LLM as a fine-grained planner to generate detailed navigation plans from the concise instructions of REVERIE to improve navigation success.

**LLMs as Embodied Planner** Benefiting from the rise of LLMs, recent works [17, 34] have explored the use of LLMs

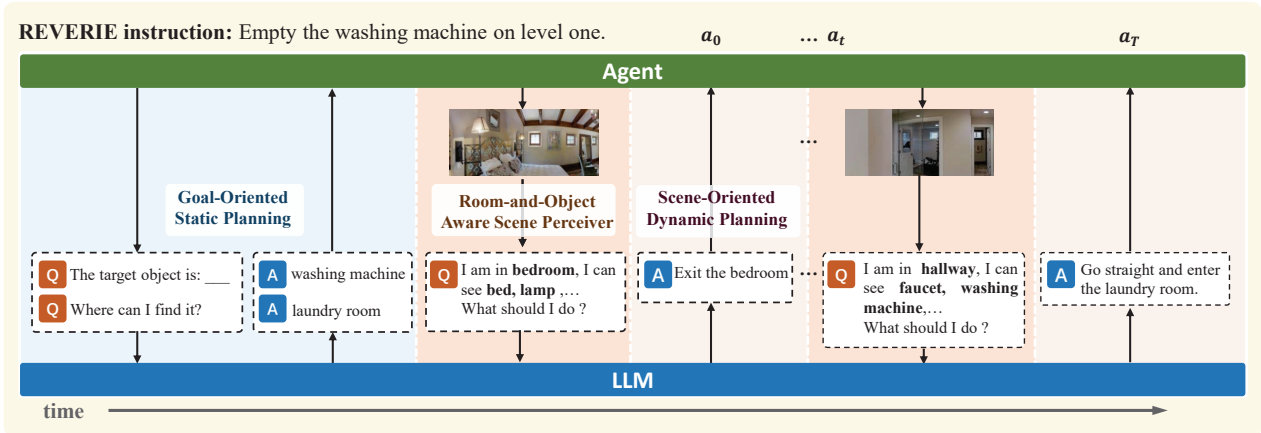


Figure 2: Overview of our March-in-Chat model. The program runs along the vertical arrows from left to right, progressing with the time flow. Our model first performs Goal-Oriented Static Planning (GOSP, Sec. 4.1.1) to reason the target object and its possible lying room; then the Room-and-Object Aware Scene Perceiver (ROASP, Sec. 4.2) perceives what room type the agent currently stands in and what prominent objects can be seen; these information are used by the Scene-Oriented Dynamic Planning module (SODP, Sec. 4.1.2) to generate a detailed instruction to execute. GOSP just runs once, and we repeat ROASP and SODP until the agent chooses to stop or reaches the maximum steps.

in task planning for various embodied tasks. Huang *et al.* [17] propose to utilize the frozen LLMs (*e.g.*, GPT-2 [31], GPT-3 [5] and Codex [8]) to plan actions for the embodied agent with in-context learning [5]. SayCan [1] translates a high-level instruction into a list of candidate low-level actions with a probability, which is then multiplied by a value function for action prediction. These two LLM planners are static, which only generate action plans at the beginning of a task. By contrast, Huang *et al.* [18] propose to introduce the feedback of action progress, detected objects and human assistance into the LLM planner to re-plan atomic actions. One concurrent work by Song *et al.* [35] injects the detected objects to re-generate high-level plans with a fixed program pattern for the ALFRED [33] task.

However, these above-mentioned methods mainly concentrate on planning atomic actions for object manipulation in a very limited space with simple scenes. By contrast, REVERIE has plenty of much larger and more complicated environments: 90 multi-layer buildings of various styles (*e.g.*, office, home, gym, to name a few). To handle the complex scenarios of REVERIE, we propose a Room-and-Object Aware Scene Perceiver module that helps the LLM planner dynamically interact with the environment in the form of a natural language dialogue.

### 3. Problem Setup

In the REVERIE task, given a concise and high-level instruction referring to a remote object, the agent is expected to navigate to the goal location and identify the target object in previously unseen environments. The environment is defined as an undirected graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ , where

$\mathcal{V} = \{V_i\}_{i=1}^K$  denotes  $K$  navigable nodes, and  $\mathcal{E}$  denotes connectivity edges. The agent is first placed in a starting node with the initial state  $s_0$  and perceives a panorama  $\mathcal{R}_t$  as the visual observation at each time step  $t$ . The panorama  $\mathcal{R}_t$  is split into  $n$  single view images as  $\mathcal{R}_t = \{r_i\}_{i=1}^n$ . Each single view image  $r_i$  is represented by an image feature vector and an orientation feature vector. In addition, the object features  $\mathcal{O}_t = \{o_i\}_{i=1}^m$  of  $m$  objects are extracted from the panorama view using the annotated object bounding boxes or object detectors. Then, the agent makes a sequence of actions  $\langle a_0, \dots, a_T \rangle$  to reach the target location, where each action is achieved by choosing a navigable node from the candidate list. The agent navigates in the environment until the target object is grounded or the agent reaches the pre-defined maximum trajectory length.

### 4. Method

As illustrated in Fig. 2, when initially situated at the starting position and given a concise high-level instruction such as “Empty the washing machine on level one”, the agent first queries an LLM with the **GOSP** module (Sec. 4.1.1) to find out the target object “washing machine” in the instruction and reason out the potential location “laundry room” by using the world knowledge implied in an LLM. Then the **ROASP** module (Sec. 4.2) extracts the room type and visible objects from the current visual observation to obtain the environmental feedback. With the description of the scene perception, the LLM is queried again by the **SODP** (Sec. 4.1.2) to generate the next step instruction, which is used to guide the agent to navigate to the target object in the room. The baseline agent is based on

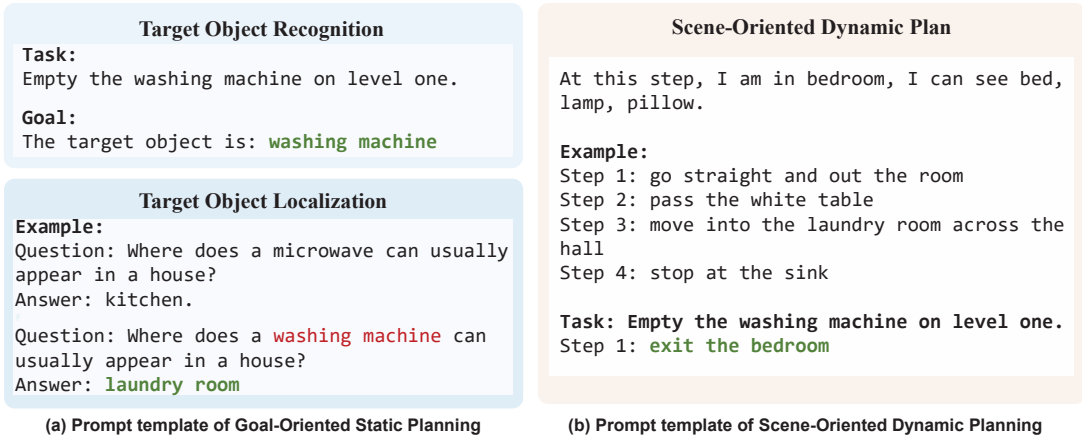


Figure 3: Examples of prompting templates for Goal-Oriented Static Planning (a) and Scene-Oriented Dynamic Planning (b). Outputs are marked with **green** color. And the **red** denotes the predicted object from Target Object Recognition.

HM3D-DUET [10], more details can be found in the supplementary material.

#### 4.1. Planning with World Knowledge from LLMs

This section illustrates how we utilize in-context learning (ICL) [5] to acquire world knowledge from the LLMs for planning. We first briefly introduce in-context learning. Then, we elaborate the Goal-Oriented Static Planning (GOSP) and Scene-Oriented Dynamic Planning (SODP). Last, we show the demonstration selection process.

**Preliminary: In-context Learning for Planning.** In-context learning (ICL) is a paradigm that lets LLMs directly make predictions based on a natural language context without gradient updates [5]. Specifically, under the setting of in-context learning, an LLM is fed a “prompt” that usually contains a task description and several demonstrations, and then the LLM generates the required outputs. Both the prompt template and the choice of demonstration examples have an impact on how well ICL performs. In this work, we use two different ICL settings to generate different navigation plans, *i.e.* the GOSP and SODP module.

The GOSP aims to identify the target object and infer the target location by arousing the world knowledge contained in an LLM through appropriate prompts. A fixed demonstration example is used for GOSP. While the SODP aims to generate step-by-step planning instructions after observing the dynamic scenes from the environment, which is more complicated than the former. To better generate plannings, we dynamically select the most suitable demonstration examples for SODP and incorporate the environmental feedback as prompts for interactive planning.

##### 4.1.1 Goal-Oriented Static Planning (GOSP)

Given a high-level concise instruction, such as “Empty the washing machine on level one”, an LLM is first asked to generate a goal-oriented static planning instruction: “Goal:

The target object is a washing machine. It is usually in a laundry room”, which emphasizes the target object and points out where the target object may lie. As shown in Fig. 3(a), the planning generation mainly consists of two sub-tasks: target object recognition and target object localization, which can be achieved by providing specifically designed prompts for the LLM. To this end, we design the prompts for the former sub-task in the form: “**Task:** Empty the washing machine on level one. **Goal:** The target object is: ”. Then the LLM will generate a corresponding answer: “washing machine”. By contrast, the latter sub-task, reasoning out the target location, is more complex because it requires more suitable prompts to arouse the internalized world knowledge in the LLM. To address this problem, we utilize a fixed demonstration example for the LLM, and design the prompts for the latter sub-task in the form: “**Example:** Question: Where does a microwave can usually appear in a house? Answer: kitchen. Question: Where does a washing machine can usually appear in a house? Answer: ”. Given such prompts, the LLM will generate the corresponding answer “laundry room”. With these answers, we can easily combine them into the goal-oriented planning format: “Goal: The target object is a washing machine. It is usually in a laundry room”.

##### 4.1.2 Scene-Oriented Dynamic Planning (SODP)

As is shown in Fig 3(b), the prompt for SODP consists of three parts. The first part is based on the scene perception of room type, such as “bedroom”, and visible objects, such as “bed, lamp, pillow”, obtained by the Room-and-Object Aware Scene Perceiver (ROASP, Sec. 4.2). These information are transformed into a natural language description of the current scene in the format of “At this step, I am in bedroom, I can see bed, lamp, pillow”. The second part is a demonstration of the fine-grained step-by-step instruction, which is selected according to the strategy



---

**R2R**  
Go up the stairs and then walk the length of the couch. Walk past the dining area and into the kitchen. Stop in front of the refrigerator.

**FGR2R**  
Step 1: go up the stair  
Step 2: and then walk the length of the couch  
Step 3: walk past the dining area and into the kitchen  
Step 4: stop in front of the refrigerator

---

Figure 4: Example of instructions in R2R and FGR2R.

detailed in the next section. The last part is previous instructions, such as “**Task:** Empty the washing machine on level one. Step 1: ”. All these three parts are concatenated together and then fed into an LLM to generate the fine-grained planning instruction for the next step accordingly, such as “Exit the bedroom”.

#### 4.1.3 Dynamic Demonstration Selection

Recent works show that providing various demonstration examples to LLMs benefits the in-context learning for different tasks [17, 22, 25]. In light of these findings, to direct the LLMs in generating better fine-grained plans, we dynamically select the most suitable demonstration example for each specific task in REVERIE as the prompt to generate the environment-aware instruction, contrary to using a single fixed demonstration for all tasks.

Specifically, we choose the training set of the Fine-Grained R2R dataset (FGR2R) [15] as the demonstration set  $\mathcal{D}$ , of which each sample will be used as a demonstration example  $D_{step}$ . As shown in Fig. 4, FGR2R decomposes each low-level instruction  $I_{low}$  of R2R dataset [4] into step-by-step instructions  $I_{step}$ . Then, given a high-level instruction  $I_{high}$  of REVERIE, a proper  $I_{low}$  will be selected as the demonstration example  $D_{step}$  by a matching algorithm. In particular, we use  $I_{high}$  as query  $Q$  and each low-level instruction  $I_{low}$  as the key  $K_i$ , both of which are embedded by the Sentence-BERT [32]. The semantic distance score between the two embeddings is calculated by the cosine similarity:

$$s(Q, K_i) = \frac{e(Q) \cdot e(K_i)}{\|e(Q)\| \|e(K_i)\|}, \quad (1)$$

where  $e(\cdot)$  is the embedding function. If  $K_i$  has the highest similarity score to the given query  $Q$ , its corresponding step-by-step instruction  $I_{step}$  will be selected as the demonstration example  $D_{step}$  for the given high-level instruction  $I_{high}$ .

## 4.2. Room-and-Object Aware Scene Perceiver

Though the world knowledge acquired from the static LLMs planner could benefit the embodied task promisingly, the static LLMs planner may generate wrong or irrelevant plans, which misleads the agent. To address this issue, the LLM planner should be aware of and interact with the

dynamic observations. In [35], the names of objects obtained from the ground truth or pre-trained detectors have been added to in-context prompts. However, the agents of these works act in a very limited space with simple scenes and monotonous objects. By contrast, REVERIE involves large-area exploration between different floors and rooms, where the scenes are more complex with more diverse objects. Considering these factors, we propose a room-and-object aware scene perceiver (ROASP) for the LLM planner, which predicts not only the room type but also the visible objects of the current location. Rather than using separate classifiers and detectors to individually predict each position’s room types and visible object categories, we use CLIP [30] as the proposed room-and-object aware scene perceiver. Thanks to CLIP’s strong ability of zero-shot image classification in the open world, the ROASP can well handle these two tasks.

Specifically, we first fetch the room type labels from the MatterPort3D [6] semantic annotations and the object type labels are extracted from the REVERIE training dataset. They are used to build the codebook for the room categories  $\mathbf{C}_{room}$  and the object categories  $\mathbf{C}_{obj}$ , respectively. Then, at each timestep  $t$ , the agent perceives the environment and obtains the panoramic visual observation  $\mathcal{R}_t = \{r_i\}_{i=1}^n$ . For each single-view observation  $r_i$  in the panorama, the image feature  $f_r$  is extracted by the CLIP Image Encoder

$$f_r = E_{CLIP}^v(r_i), \quad (2)$$

where  $E_{CLIP}^v(\cdot)$  represents the CLIP Image Encoder. For each room category  $c_{room}$  and each object category  $c_{obj}$ , we respectively construct a text phrase of room  $T_{room}$  as “a photo of a  $\{c_{room}\}$ ” and a text phrase of object  $T_{obj}$  as “a photo of a  $\{c_{obj}\}$ ”. Then the text feature is derived through the pretrained CLIP Text Encoder as:

$$f_{room} = E_{CLIP}^t(T_{room}), \quad (3)$$

$$f_{obj} = E_{CLIP}^t(T_{obj}), \quad (4)$$

where  $E_{CLIP}^t(\cdot)$  represents the CLIP Text Encoder. At last, the similarity score  $S_{room}$  between the image feature  $f_r$  and the text feature  $f_{room}$  as well as the similarity score  $S_{obj}$  between the image feature  $f_r$  and the text feature  $f_{obj}$  are respectively computed as:

$$S_{room} = \text{Softmax}(f_{room} \cdot f_r^T), \quad (5)$$

$$S_{obj} = \text{Softmax}(f_{obj} \cdot f_r^T). \quad (6)$$

Considering that the current environment normally belongs to only one type of room, though the panoramic images have multiple views, the room that the agent is currently centered in should have the largest influence on each view. Thus, we average the predicted room type scores  $S_{room}$  from multiple views and choose the room type with

|      |  |
|------|--|
| HLI  | Empty the washing machine on level one.                                    |
| GOSP | Goal: The target object is washing machine. It is usually in laundry room. |
| SODP | Step 1: exit the bedroom<br>Step 2: go down the stairs<br>...              |

Figure 5: Text inputs contains three parts: High-level Instruction in REVERIE (HLI), Goal-Oriented Static Planning (GOSP) and Scene-Oriented Dynamic Planning (SODP) returned instructions.

the greatest score as the room type prediction  $\hat{c}_{\text{room}}$ . For object predictions, if the object occupies more proportion in a view, the matching score  $S_{\text{obj}}$  should be higher. Thus, we select  $k$  prominent objects with the top- $k$  matching scores as the auxiliary environment feedback in addition to the predicted room.

### 4.3. March with Interactive Prompting

When the generation of the goal-oriented planning and the scene-oriented planning with perceptions from the environment is finished, the agent can march towards the target object at each timestep  $t$  under the guidance of the interactive prompting. In this section, we will give a detailed description of how the interactive prompting works during the process of navigation, which mainly consists of two parts, *i.e.* the assembled instruction and the instruction update.

**Assembled Instruction** At each timestep  $t$ , the agent observes the environment and receives the assembled instructions obtained from the above-mentioned modules, and choose an action to perform. Specifically, as shown in Fig. 5, the assembled instructions  $W$  of the interactive prompting mainly consist of three parts: the high-level instruction (HLI)  $W_I$  in REVERIE, the GOSP instruction  $W_G$  and the SODP instruction  $W_S$ . We concatenate these three parts of instructions as the assembled instruction  $W = [W_I, W_G, W_S]$  and use WordPieces [20] to tokenize all the words into a sequence of tokens as the textual input for the agent. Then, the agent will act under the guidance of such assembled instruction. Note that the use of the original high-level instruction  $W_I$  can improve the model’s tolerance on the noise of intermediate planning instructions.

**Instruction Update** The GOSP is only conducted once at the beginning of the task. While the SODP is conducted depending on the feedback of environments. Specifically, at each timestep  $t$ , if the ROASP finds the room has changed where the predicted room  $\hat{c}_{\text{room}}^t$  does not equal to  $\hat{c}_{\text{room}}^{t-1}$ , the SODP will be triggered again. Then, a new step-by-step instruction such as “Step 2: go down the stairs” for the next few steps will be generated by the LLM and added to the previous assembled instruction  $W$  after the last step-by-step

instruction of “Step 1: exit the bedroom”. Then, the agent will act under the guidance of the updated instructions  $W'$ .

## 5. Experiment

### 5.1. Evaluation Setup

**Dataset** REVERIE [27] contains 10,567 panoramic images within 90 buildings (4,140 target objects divided into 489 categories) and 21,702 instructions with 18 words on average. Each target viewpoint has 7 distinct panoramic objects with 50 bounding boxes on average. It consists of four splits: train, validation seen, validation unseen and test unseen.

**Evaluation Metrics** The performance of agents is evaluated in two ways: navigation and object grounding. For the navigation sub-task, the metrics are **Success Rate (SR)**, **Oracle Success Rate (OSR)**, and **Success weighted by Path Length [3] (SPL)**, where SPL is the main metric. For the grounding sub-task, the metrics are **Remote Grounding Success rate (RGS)** and **RGS weighted by Path Length (RGSPL)**, where RGSPL is the main metric for this sub-task. For all these metrics, higher is better.

**TL Trajectory Length** measures the average length of all the predicted navigation trajectories in meters.

**SR Success Rate** measures the ratio of successful tasks, of which the agent’s stop location is less than 3 meters away from the target location.

**OSR Oracle Success Rate** measures the ratio of tasks of which one of its trajectory viewpoints can observe the target object within 3 meters.

**SPL Success weighted by Path Length** trades-off SR (Success Rate) against TL (Trajectory Length). It measures both the accuracy and efficiency of navigation.

**RGS Remote Grounding Success rate** measures the ratio of tasks that successfully locate the target object.

**RGSPL RGS weighted by Path Length** is RGS.

**Implementation Details** Our model is trained on a single 3090 GPU for 30,000 iterations. We set the batch size to 4 and the learning rate to  $1 \times 10^{-5}$ . The best model is selected according to performance on the validation unseen split. We use the same pretrained model and augmented data as [10] for a fair comparison. For the LLMs, we use the public GPT-2 [31] model for in-context learning. For the scene preceptor, we keep the top 3 object predictions for each position.

### 5.2. Comparison with State-of-The-Art Methods

As shown in Table 1, we compare MiC with the state-of-the-art methods on the REVERIE benchmark. Our method

| Methods         | Val Unseen |              |              |              |              |              | Test Unseen  |              |              |              |              |              |
|-----------------|------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                 | Navigation |              |              | Grounding    |              |              | Navigation   |              |              | Grounding    |              |              |
|                 | TL         | OSR↑         | SR↑          | SPL↑         | RGS↑         | RG SPL↑      | TL           | OSR↑         | SR↑          | SPL↑         | RGS↑         | RG SPL↑      |
| Human           | -          | -            | -            | -            | -            | -            | 21.18        | 86.83        | 81.51        | 53.66        | 77.84        | 51.44        |
| Seq2Seq         | 11.07      | 8.07         | 4.20         | 2.84         | 2.16         | 1.63         | 10.89        | 6.88         | 3.99         | 3.09         | 2.00         | 1.58         |
| RCM [37]        | 11.98      | 14.23        | 9.29         | 6.97         | 4.89         | 10.60        | 7.84         | 3.89         | 11.68        | 6.67         | 3.67         | 3.14         |
| SMNA [23]       | 9.07       | 11.28        | 8.15         | 6.44         | 4.54         | 3.61         | 9.23         | 8.39         | 5.80         | 4.53         | 3.10         | 2.39         |
| FAST-MATTN [27] | 45.28      | 28.20        | 14.40        | 7.19         | 7.84         | 4.67         | 39.05        | 30.63        | 19.88        | 11.61        | 11.28        | 6.08         |
| ORIST [26]      | 10.90      | 25.02        | 16.84        | 15.14        | 8.52         | 7.58         | 11.38        | 29.20        | 22.19        | 18.97        | 10.68        | 9.28         |
| CKR [12]        | 26.26      | 31.44        | 19.14        | 11.84        | 11.45        | -            | 22.46        | 30.40        | 22.00        | 14.25        | 11.60        | -            |
| RecBERT [16]    | 16.78      | 35.02        | 30.67        | 24.90        | 18.77        | 15.27        | 15.86        | 32.91        | 29.61        | 23.99        | 16.50        | 13.51        |
| Airbert [13]    | 18.71      | 34.51        | 27.89        | 21.88        | 18.23        | 14.18        | 17.91        | 34.20        | 30.28        | 23.61        | 16.83        | 13.28        |
| HAMT [9]        | 14.08      | 36.84        | 32.95        | 30.20        | 18.92        | 17.28        | 13.62        | 33.41        | 30.40        | 26.67        | 14.88        | 12.08        |
| HOP [28]        | 16.46      | 36.24        | 31.78        | 26.11        | 18.85        | 15.73        | 16.38        | 33.06        | 30.17        | 24.34        | 17.69        | 14.34        |
| TD-STP [38]     | -          | 39.48        | 34.88        | 27.32        | 21.16        | 16.56        | -            | 40.26        | 35.89        | 27.51        | 19.88        | 15.40        |
| DUET [11]       | 22.11      | 51.07        | 46.98        | 33.73        | 32.15        | 23.03        | 21.30        | 56.91        | 52.51        | 36.06        | 31.88        | 22.06        |
| HM3D-DUET [10]  | -          | 62.14        | 55.89        | 40.85        | 36.58        | 26.76        | -            | 62.30        | 55.17        | 38.88        | 32.23        | 22.68        |
| MiC             | 20.64      | <b>62.37</b> | <b>56.97</b> | <b>43.60</b> | <b>37.52</b> | <b>28.72</b> | <b>18.11</b> | <b>62.40</b> | <b>55.74</b> | <b>41.97</b> | <b>35.25</b> | <b>26.17</b> |

Table 1: Comparison with the state-of-the-art methods on REVERIE.

| Components           | Navigation |       |       | Grounding |         |
|----------------------|------------|-------|-------|-----------|---------|
|                      | OSR↑       | SR↑   | SPL↑  | RGS↑      | RG SPL↑ |
| <b>HLI(Baseline)</b> | 58.02      | 52.71 | 40.49 | 34.93     | 26.82   |
| <b>HLI+GOSP</b>      | 59.92      | 55.28 | 42.46 | 37.13     | 28.24   |
| <b>HLI+SODP</b>      | 60.72      | 56.26 | 42.94 | 36.80     | 27.81   |
| <b>HLI+GOSP+SODP</b> | 62.37      | 56.97 | 43.60 | 37.52     | 28.72   |

Table 2: Ablation of different components in MiC.

outperforms previous methods in all metrics on both validation unseen and test unseen splits. Particularly, compared with the SoTA method HM3D-DUET [10], MiC outperforms HM3D-DUET by a large margin of 3.09% in terms of the main navigation metric SPL and 3.49% of the main object grounding metric RGSPL on the Test Unseen split. Note that MiC shares the same pre-trained model with the HM3D-DUET, these promising result demonstrates that our method can effectively improve the navigation and object grounding ability of agents.

### 5.3. Ablation Analysis

**Contribution of different MiC Components** In Table 2, we evaluate the effect of different components in our proposed MiC. HLI denotes only using the original high-level instruction (HLI) provided by REVERIE.

Compared to the baseline HLI, GOSP improves the performance of both navigation (2.57%↑ on SR, 1.97%↑ on SPL) and object grounding (2.20%↑ on RGS, 1.42%↑ on RGSPL) with a non-trivial margin, showing the effectiveness of the proposed goal-oriented static planning. SODP further surpasses GOSP in the navigation metric (0.98%↑ on SR, 0.48%↑ on SPL) while falling a little behind in the grounding metrics (0.33%↑ on RGS, 1.43%↑ on RGSPL). The reason may be that the detailed step-by-step planning occupies a large proportion compared to the target object

| Methods         | Navigation |       |       | Grounding |         |
|-----------------|------------|-------|-------|-----------|---------|
|                 | OSR↑       | SR↑   | SPL↑  | RGS↑      | RG SPL↑ |
| <b>Baseline</b> | 58.02      | 52.71 | 40.49 | 34.93     | 26.82   |
| <b>Static</b>   | 60.24      | 55.35 | 41.74 | 36.30     | 27.03   |
| <b>Dynamic</b>  | 60.72      | 56.26 | 42.94 | 36.80     | 27.81   |

Table 3: Comparison of different plan generation settings.

| Methods                                | Relevancy | Rationality |
|--|-----------|-------------|
| <b>Scene-Oriented Dynamic Planning</b> | 2.06      | 1.93        |
| - w/o Dynamic Demonstration            | 1.41      | 1.23        |
| - w/o ROASP                            | 1.64      | 1.55        |

Table 4: Human study of the prompt setting for Scene-Oriented Dynamic Planning.

in the input texts, which can bring the noise for object grounding while improving navigation performance. When combining all these components, the final performance gets further increased in all metrics, which surpasses the baseline with a large margin (4.26%↑ on SR, 3.11%↑ on SPL, 2.59%↑ on RGS and 1.9%↑ on RGSPL). The promising results here show that these components are complementary to each other.

**The Effect of the ROASP** To evaluate the effectiveness of ROASP used for the scene-oriented dynamic planning, we conduct another ablation study via whether incorporating the feedback from the ROASP module on REVERIE validation unseen set. We report results in three settings: **(I)** Baseline: The input assembled instruction only contains the given high-level instruction in REVERIE. **(II)** Static: The input assembled instruction contains the REVERIE and fine-grained static instructions. The difference between

| <b>REVERIE Instruction:</b> Go to the bathroom with the two towels and empty the lower cabinet.  |   |  |
|--|---|--|
| Scene-Oriented Dynamic Planning  | w/o Dynamic Demonstration   | w/o ROASP  |
| Step 1: exit the bedroom<br>Step 2: and walk into the next bedroom in the hall<br>Step 3: go stand in the bathroom                                       | Step 1: put the towel in the bottom drawer of the toilet  | Step 1: exit the bathroom<br>Step 2: walk into the kitchen<br>Step 3: go stand in the kitchen doorway<br>Step 4: go back to the bedroom and take the towel<br>Step 5: exit the bedroom<br>Step 6: walk into the next bedroom in the hall |
| <b>REVERIE Instruction:</b> Go to the kitchen on level 2 and open the drawer.  |   |  |
| Scene-Oriented Dynamic Planning  | w/o Dynamic Demonstration   | w/o ROASP  |
| Step 1: go upstairs<br>Step 2: go straight pass the couch<br>Step 3: make a slight left towards the refrigerator<br>Step 4: wait near the kitchen drawer | Step 1: open the drawer with the key in it<br>Step 2: remove the lock from the door<br>Step 3: then turn on | Step 1: go upstairs and turn left at the hall<br>Step 2: go back down the stairs<br>Step 3: go up the stairs again<br>Step 4: go back to the kitchen and open the fridge<br>Step 5: take out the fruit and put it in the juicer          |

Figure 6: Examples of generated instructions.

fine-grained static instruction and scene-oriented dynamic instruction is that static fine-grained instruction is generated without ROASP. More specifically, the query prompt for the LLM to generate step-by-step planning is fixed at each timestep, which only consists of the given high-level instruction and the selected demonstration. **(III) Dynamic:** The input assembled instruction contains the high-level instruction in REVERIE and scene-oriented dynamic planning instruction. As shown in Table 3, in the static setting, the performance in all metrics is improved compared to the baseline, indicating the effectiveness of the LLM’s rich world knowledge in fine-grained planning. In the dynamic setting, the performance is further improved with non-trivial margins, showing the effectiveness of ROASP.

**Qualitative Analysis of Prompt Setting** To further evaluate the effect of dynamic demonstration and ROASP in SODP, we perform a human evaluation about the generated plannings (see Table 4) and show the planning results (see Fig. 6). For human evaluation, we randomly selected 100 REVERIE tasks and generate fine-grained step-by-step instructions in setting of SODP, SODP without dynamic demonstration, and SODP without ROASP. We asked 10 volunteers to mark the generated step-by-step instructions in terms of their relevancy and rationality. The relevancy score ranges from 0 (unrelated) to 3 (very related), which takes into account whether the keywords in instructions are related to the REVERIE task. For example, regarding the REVERIE instruction “Go to the kitchen and turn on the microwave”, whether there are keywords in instructions related to the kitchen scene could be rated. Rationality is rated from 0 (bad) to 3 (perfect), considering whether the instruction conforms to the logic of navigation.

The results are presented in Table 4. It shows that our SODP scored 2.06 on Relevancy and 1.93 on Rational-

ity, which could be considered acceptable since the highest score is 3 and it is challenging to generate instructions that are consistent with tasks and actual navigation logic. When removing the dynamic demonstration, the score of generated instruction drops about 31.55% on Relevancy and 36.27% on Rationality, which could also be observed in Fig. 6. Although the instruction generated without dynamic demonstration is related to the task to some extent (*e.g.*, “put the towel in the bottom drawer of the toilet” has the keyword “towel”, the instruction lacks navigation information, such as how to reach the bathroom.) As shown in the bottom example of Fig. 6, instruction without ROASP successfully guided how to go to the destination location kitchen, but it still caused confusion by going upstairs and going downstairs several times, and thus reducing the rationality score, *i.e.* 1.64 on Relevancy and 1.55 on Rationality. More generation results can be found in the supplementary.

## 6. Conclusions

In this work, we propose a novel model, March-in-Chat (MiC), for the REVEIRE task, which only provides concise high-level instructions for the VLN agent. MiC enables the REVERIE agent to talk with an LLM on the fly to generate plans for the next few steps. It consists of three main modules, Goal-Oriented Static Planning (GOSP), Scene-Oriented Dynamic Planning (SODP), and Room-and-Object Aware Scene Perceiver (ROASP) module. We conduct extensive quantitative and qualitative experiments on REVERIE and the promising results show the effectiveness of our method.

## 7. Acknowledgements

Jing Liu is supported by the National Key Research and Development Program of China (No. 2020AAA0106400).



## References

- [1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022. 3
- [2] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Topo-metric map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022. 2
- [3] Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir Roshan Zamir. On evaluation of embodied navigation agents. *CoRR*, abs/1807.06757, 2018. 6
- [4] 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. 1, 2, 5
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, pages 1877–1901, 2020. 3, 4
- [6] 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. 5
- [7] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snaveley, and Yoav Artzi. TOUCHDOWN: natural language navigation and spatial reasoning in visual street environments. In *CVPR*, pages 12538–12547, 2019. 1, 2
- [8] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. 3
- [9] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, pages 5834–5847, 2021. 7
- [10] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Learning from unlabeled 3d environments for vision-and-language navigation. In *ECCV*, 2022. 4, 6, 7
- [11] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, 2022. 2, 7
- [12] Chen Gao, Jinyu Chen, Si Liu, Luting Wang, Qiong Zhang, and Qi Wu. Room-and-object aware knowledge reasoning for remote embodied referring expression. In *CVPR*, pages 3064–3073, 2021. 7
- [13] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *ICCV*, pages 1634–1643, 2021. 7
- [14] Keji He, Yan Huang, Qi Wu, Jianhua Yang, Dong An, Shuanglin Sima, and Liang Wang. Landmark-rxr: Solving vision-and-language navigation with fine-grained alignment supervision. *NeurIPS*, 34:652–663, 2021. 2
- [15] Yicong Hong, Cristian Rodriguez Opazo, Qi Wu, and Stephen Gould. Sub-instruction aware vision-and-language navigation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *EMNLP*, pages 3360–3376, 2020. 5
- [16] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez Opazo, and Stephen Gould. VLN $\odot$ BERT: A recurrent vision-and-language BERT for navigation. In *CVPR*, pages 1643–1653, 2021. 2, 7
- [17] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *ICML*, pages 9118–9147. PMLR, 2022. 2, 3, 5
- [18] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022. 3
- [19] Vihan Jain, Gabriel Magalhães, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. pages 1862–1872, 2019. 2
- [20] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguistics*, 5:339–351, 2017. 6
- [21] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, pages 4392–4412, 2020. 1, 2
- [22] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*, 2021. 5

- [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 *ICLR*, 2019. 7
- [24] Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. pages 684–695. Association for Computational Linguistics, 2019. 2
- [25] Gabriel Poesia, Alex Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. Synchromesh: Reliable code generation from pre-trained language models. In *ICLR*, 2022. 5
- [26] Yuankai Qi, Zizheng Pan, Yicong Hong, Ming-Hsuan Yang, Anton van den Hengel, and Qi Wu. The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation. In *ICCV*, pages 1655–1664, 2021. 2, 7
- [27] 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 *CVPR*, pages 9979–9988, 2020. 1, 2, 6, 7
- [28] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. HOP: history-and-order aware pre-training for vision-and-language navigation. In *CVPR*, pages 15397–15406, 2022. 2, 7
- [29] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. Hop+: History-enhanced and order-aware pre-training for vision-and-language navigation. *IEEE TPAMI*, 2023. 2
- [30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, volume 139, pages 8748–8763, 2021. 5
- [31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3, 6
- [32] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *EMNLP-IJCNLP*, pages 3980–3990, 2019. 5
- [33] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *CVPR*, pages 10737–10746, 2020. 1, 3
- [34] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. ProgPrompt: Generating situated robot task plans using large language models. 2022. 2
- [35] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. *arXiv preprint arXiv:2212.04088*, 2022. 3, 5
- [36] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *CoRL*, pages 394–406, 2019. 1, 2
- [37] 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. 7
- [38] Yusheng Zhao, Jinyu Chen, Chen Gao, Wenguan Wang, Lirong Yang, Haibing Ren, Huaxia Xia, and Si Liu. Target-driven structured transformer planner for vision-language navigation. pages 4194–4203, 2022. 7
- [39] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. SOON: scenario oriented object navigation with graph-based exploration. In *CVPR*, pages 12689–12699, 2021. 1, 2