Layout-Aware Dreamer for Embodied Referring Expression Grounding

Mingxiao Li*[1], Zehao Wang*[2], Tinne Tuytelaars[2], Marie-Francine Moens[1]

[1] Computer Science Department of KU Leuven
[2] Electrical Engineering Department (ESAT-PSI) of KU Leuven
mingxiao.li@cs.kuleuven.be
zehao.wang@esat.kuleuven.be
tinne.tuytelaars@esat.kuleuven.be
sien.moens@cs.kuleuven.be

Abstract

In this work, we study the problem of Embodied Referring Expression Grounding, where an agent needs to navigate in a previously unseen environment and to localize a remote object described by a concise high-level natural language instruction. When facing such a situation, a human tends to imagine what the destination may look like and to explore the environment based on prior knowledge of the environmental layout, such as the fact that a bathroom is more likely to be found near a bedroom than a kitchen. We have designed an autonomous agent called Layout-aware Dreamer (LAD), including two novel modules, that is, the *Layout Learner* and the *Goal Dreamer* to mimic this cognitive decision process. The *Layout Learner* learns to infer the room category distribution of neighboring unexplored areas along the path for coarse layout estimation, which effectively introduces layout common sense of room-to-room transitions to our agent. To learn an effective exploration of the environment, the *Goal Dreamer* imagines the destination beforehand. Our agent achieves new state-of-the-art performance on the public leaderboard of the REVERIE dataset in challenging unseen test environments with improvement in navigation success (SR) by 4.02% and remote grounding success (RGS) by 3.43% compared to the previous state-of-the-art. The code is released at https://github.com/zehao-wang/LAD

Introduction

In recent years, embodied AI has matured. In particular, a lot of works (Chen et al. 2022; Hao et al. 2020; Majumdar et al. 2020; Wang et al. 2019; Song et al. 2022; Chen et al. 2021a; Georgakis et al. 2022) have shown promising results in Vision-and-Language Navigation (VLN) (Anderson et al. 2018c; Krantz et al. 2020). In VLN, an agent is required to reach a destination following a fine-grained natural language instruction that provides detailed step-by-step information along the path, for example "Walk forward and take a right turn. Enter the bedroom and stop at the bedside table". However, in real-world applications and human-machine interactions, it is tedious for people to give such detailed step-by-step instructions. Instead, a high-level instruction only describing the destination, such as "Go to the bedroom and clean the picture on the wall.", is more usual.

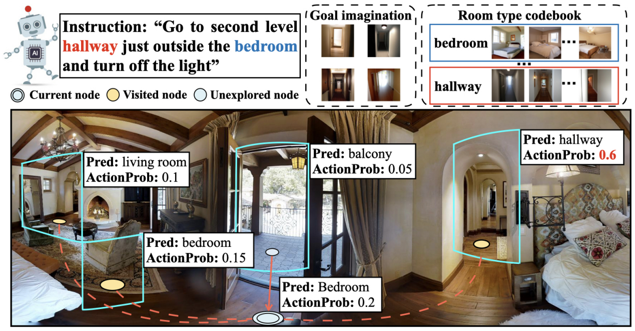

Figure 1: The agent is required to navigate and find the mentioned object in the environment. Based on acquired commonsense knowledge, the agent correctly identifies the current and surrounding room types. Based on the imagination of the destination, it correctly chooses the unexplored right yellow dot as the next step to go.

In this paper, we target such high-level instruction-guided tasks. Specifically, we focus on the Embodied Referring Expression Grounding task (Qi et al. 2020; Zhu et al. 2021). In this task, an agent receives a high-level instruction referring to a remote object, and it needs to explore the environment and localize the target object. When given a high-level instruction, we humans tend to imagine what the scene of the destination looks like. Moreover, we can efficiently navigate to the target room, even in previously unseen environments, by exploiting commonsense knowledge about the layout of the environment. However, for an autonomous agent, generalization to unseen environments still remains challenging.

Inspired by how humans make decisions when receiving high-level instructions in unseen environments, as shown in Fig. 1, we design an agent that can identify the room type of current and neighboring navigable areas based on a room type codebook and previous states. On top of that, it learns to combine this information with goal imagination to jointly infer the most probable moving direction. Thus, we propose two modules named **Layout Learner** and **Goal Dreamer** to achieve this goal. In our model, the agent stores trajectory information by building a real-time topological map, where nodes represent either visited or unexplored but partially visible areas. The constructed topological map can be seen as

[*] These authors contributed equally.

a long-term scene memory. At each time step, the node representation is updated by moving the agent to the current node and receiving new observations. The **Layout Learner** module learns to infer the layout distribution of the environment with a room-type codebook constructed using a large-scale pre-trained text-to-image model, GLIDE (Nichol et al. 2022) in our case. The codebook design helps the model to leverage high-level visual commonsense knowledge of room types and boosts performance in layout learning. This prediction is updated at each time step allowing the agent to correct its prediction when more areas are explored. In the **Goal Dreamer** module, we encourage the agent to imagine the destination beforehand by generating a set of images with the text-to-image model GLIDE. The use of this imagination prior helps accurate action prediction. The cross-attention between topological node representations and imagination features is conducted, and its output is used to help the agent make a decision at each time step. In summary, the contributions of our paper are threefold:

- We propose a **Layout Learner** which leverages the visual commonsense knowledge from a room-type codebook generated using the GLIDE model. It not only helps the agent to implicitly learn the environment layout distribution but also to better generalize to unseen environments.
- The novel **Goal Dreamer** module equips the agent with the ability to make action decisions based on the imagination of the unseen destination. This module further boosts the action prediction accuracy.
- Analyzing different codebook room types shows that visual descriptors of the room concept better generalize than textual descriptions and classification heads. This indicates that, at least in this embodied AI task, visual features are more informative.

Related Work

Embodied Referring Expression Grounding. In the Embodied Referring Expression Grounding task (Qi et al. 2020; Zhu et al. 2021), many prior works focus on adapting multimodal pre-trained networks to the reinforcement learning pipeline of navigation (Qi et al. 2020; Lin, Li, and Yu 2021) or introducing pretraining strategies for good generalization (Qiao et al. 2022; Hong et al. 2021). Some recent breakthroughs come from including on-the-fly construction of a topological map and a trajectory memory as done in VLN-DUET (Chen et al. 2022). Previous models only consider the observed history when predicting the next step. Different from them, we design a novel model to imagine future destinations while constructing the topological map.

Map-based Navigation. In general language-guided navigation tasks, online map construction gains increasing attention (e.g., (Chaplot et al. 2020a,b; Irshad et al. 2022)). A metric map contains full semantic details in the observed space and precise information about navigable areas. Recent works focus on improving subgoal identification (Min et al. 2021; Blukis et al. 2022; Song et al. 2022) and path-language alignment (Wang et al. 2022). However, online metric map construction is inefficient during large-scale training, and its quality suffers from sensor noise in real-world applications. Other studies focus on topological maps (Hahn et al. 2021; Chen et al. 2021a, 2022), which provide a sparser map representation and good backtracking properties. We use topological maps as the agent's memory. Our agent learns layout-aware topological node embeddings that are driven by the prediction of room type as the auxiliary task, pushing it to include commonsense knowledge of typical layouts in the representation.

Visual Common Sense Knowledge. Generally speaking, visual common sense refers to knowledge that frequently appears in a day-to-day visual world. It can take the form of a hand-crafted knowledge graph such as Wikidata (Vrandečić and Krötzsch 2014) and ConceptNet (Liu and Singh 2004), or it can be extracted from a language model (Cirik, Morency, and Berg-Kirkpatrick 2022). However, the knowledge captured in these resources is usually abstract and hard to align with objects mentioned in free language. Moreover, if, for instance, you would like to know what a living room looks like, then several images of different living rooms will form a more vivid description than its word definition. Existing work (Gao et al. 2021a) tries to boost the agent's performance using an external knowledge graph in the Grounding Remote Referring Expressions task. Inspired by the recent use of prompts (Petroni et al. 2019; Brown et al. 2020) to extract knowledge from large-scale pre-trained language models (PLM) (Devlin et al. 2019; Brown et al. 2020; Kojima et al. 2022), we consider pre-trained text-to-image models (Nichol et al. 2022; Ramesh et al. 2022) as our visual commonsense resources. Fine-tuning a pre-trained vision-language model has been used in multimodal tasks (Lu et al. 2019; Su et al. 2019). However, considering the explicit usage of prompted images as visual common sense for downstream tasks is novel. Pathdreamer (Koh et al. 2021) proposes a model that predicts future observations given a sequence of previous panoramas along the path. It is applied in a VLN setting requiring detailed instructions for path sequence scoring. Our work studies the role of general visual commonsense knowledge and focuses on room-level imagination and destination imagination when dealing with high-level instructions. The experiments show that, on the one hand, including visual commonsense knowledge essentially improves task performance. On the other hand, visual common sense performs better than text labels on both environmental layout prediction and destination estimation.

Methodology

Overview

Task Setup. In Embodied Referring Expression Grounding task (Qi et al. 2020; Zhu et al. 2021), an agent is spawned at an initial position in an indoor environment. The environment is represented as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ stands for navigable nodes and $\mathcal{E}$ denotes connectivity edges. The agent is required to correctly localize a remote target object described by a high-level language instruction. Specifically, at the start position of each episode,

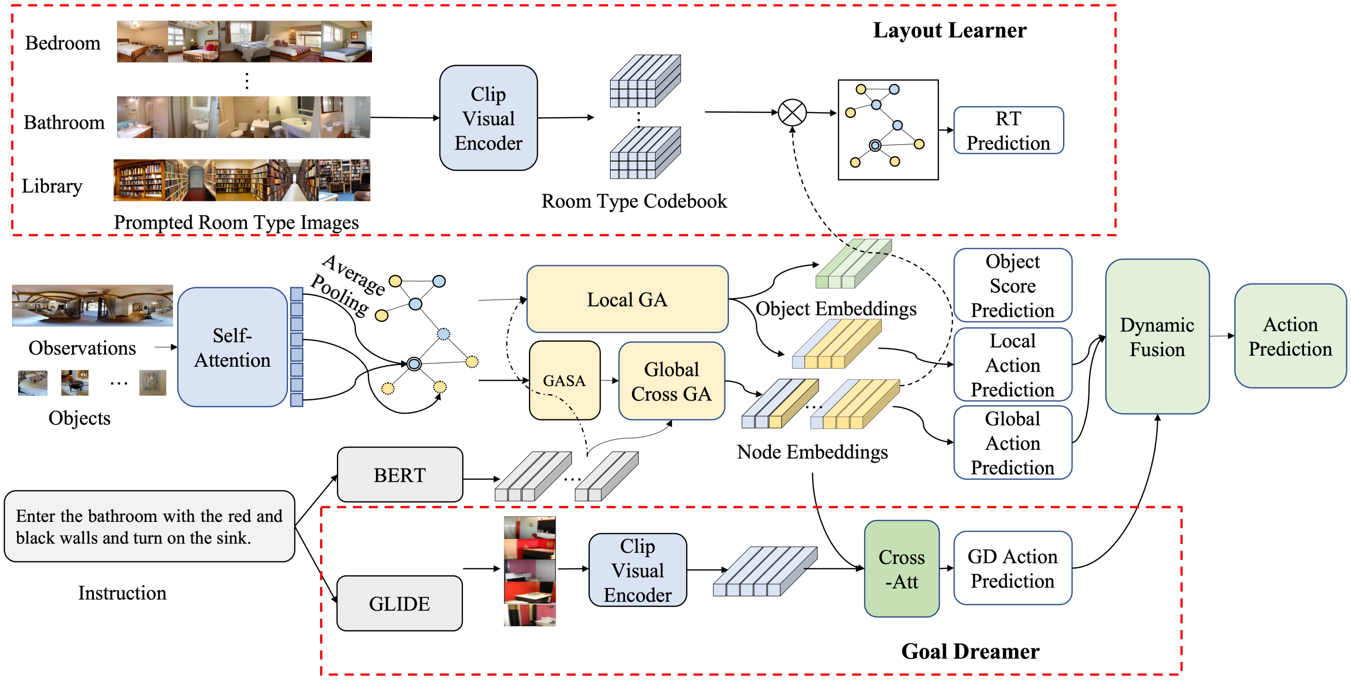

Figure 2: The model architecture of our Layout-aware Dreamer (LAD). Our model predicts the room type of all nodes of the topological graph; for simplicity, we only show the predictions of several nodes here. The center part is the baseline model, which takes the topological graph and instruction as inputs and dynamically fuses the local and global branch action decisions to predict the next action. The dashed boxes show our proposed Layout Learner and Goal Dreamer.

the agent receives a concise high-level natural language instruction $\mathcal{X} =< w_1, w_2, ..., w_L >$, where L is the length of the instruction, and w_i represents the ith word token. The panoramic view $\mathcal{V}_t = \{v_{t,i}\}_{i=1}^{36}$ of the agent location at time step t is represented by 36 images which are the observations of the agent with 12 heading angles and 3 elevations. At each time step t, the agent has access to the state information S_t of its current location consisting of panoramic view $\mathcal{V}_t$, and M neighboring navigable nodes $A_t = [a_{t,1}, ..., a_{t,M}]$ with only a single view for each of these, namely the view observable from the current node. These single views of neighboring nodes form $\mathcal{N}_t = [v_{t,1}, ..., v_{t,M}]$. Then the agent is required to take a sequence of actions $< a_0, ...a_t, ... >$ to reach the goal location and ground the object specified by the instruction in the observations. The possible actions a_t at each time step t are either selecting one of the navigable nodes A_t or stopping at the current location denoted by $a_{t,0}$.

Base Architecture

Our architecture is built on the basis of the VLN-DUET (Chen et al. 2022) model, which is the previous state-of-the-art model on the REVERIE dataset. In the following paragraphs, we briefly describe several important components of this base architecture, including topological graph construction, the global and local cross-attention modules with minimal changes. For more details, we refer the reader to Chen et al. (2022).

Topological Graph. The base model gradually builds a topological graph $\mathcal{G}_t = \{\boldsymbol{v}, \boldsymbol{e} \mid \boldsymbol{v} \subseteq \mathcal{V}, \boldsymbol{e} \subseteq \mathcal{E}\}$ to represent the environment at time step t. The graph contains three types of nodes: (1) visited nodes; (2) navigable nodes; and (3) the current node. In Fig. 2, they are denoted by a blue circle, yellow circle, and double blue circle, respectively. Both visited nodes and the current node have been explored and the agent has access to their panoramic views. The navigable nodes are unexplored and can be partially observed from other visited nodes. When navigating to a node $a_{t,0}$ at time step t, the agent extracts panoramic image features $\mathcal{R}_t$ from its panoramic view $\mathcal{V}_t$ and object features $\mathcal{O}_t$ from provided object bounding box. The model then uses a multi-layer transformer with self-attention to model the relation between the image features $\mathcal{R}_t$ and the object features $\mathcal{O}_t$. The fused features $\mathcal{R}_t$ and $\mathcal{O}_t$ are treated as local visual features of node $a_{t,0}$. During exploring the environment, the agent updates the node visual representations as follows: (1) For the current node, the node representation is updated by concatenating and average pooling the local features $\hat{\mathcal{R}}_t$ and $\hat{\mathcal{O}}_t$. (2) As unvisited nodes could be partially observed from different directions of multiple visited nodes, the average of all image features of the partial views are taken as its representation. (3) The features of visited nodes remain unchanged. The final representation of nodes is the sum of location embedding, step embedding and visual embedding. The location embedding of a node is formed by the concatenation of Euclidean distance, heading and elevation angles relative to the current node. The step embedding embeds the last visited time step of each visited node, and time zero is set for unvisited nodes.

Language Encoder. We use a multi-layer transformer encoder (Vaswani et al. 2017) to encode the natural language instruction $\mathcal{X}$. Following the convention, we feed the sum of the token embedding, position embedding and token type embedding into the transformer encoder, and the output denoted as $\mathcal{T}$ is taken as language features.

Global Node Self-Attention. Different from the VLN-DUET model, to enable each node to perceive global environment information without influenced by the language information, we conduct one more graph aware self-attention (GASA) (Chen et al. 2022) over node embeddings $\mathcal{H}_t$ of graph $\mathcal{G}_t$ before interacting with word embeddings. For simplicity, we use the same symbol $\mathcal{H}_t$ to denote the encoded graph node embeddings.

Cross Graph Encoder. Following the work of Devlin et al. (2019), we use a multimodal transformer (Lu et al. 2019) to model both the global and local graph-language relation. We name the global and local graph-language cross-attention models (Global Cross GA and Local GA) as global branch and local branch, respectively. For the global branch, we perform cross-attention of node embeddings $\mathcal{H}_t$ over language features $\mathcal{T}$, while only the current node and its neighboring navigable nodes are used to compute the cross-attention in the local branch. We feed the outputs of the global branch to the Layout Learner for layout prediction. In addition, both the global and local branch outputs are further used to make the navigation decision, as shown in Fig. 2.

$$\tilde{\mathcal{H}}_t^{(glo)} = \text{Cross-Attn}(\mathcal{H}_t, \mathcal{T}, \mathcal{T}) \tag{1}$$

$$\tilde{\mathcal{H}}_t^{(loc)} = \text{Cross-Attn}(\{\mathcal{H}_t(\mathcal{A}_t), \mathcal{H}_t(a_{t,0})\}, \mathcal{T}, \mathcal{T}) \tag{2}$$

where Cross-Attn$(query, key, value)$ is a multi-layer transformer decoder, and $\mathcal{A}_t$ stands for neighbouring navigable nodes of the current node $a_{t,0}$. $\mathcal{H}_t(\cdot)$ represents extracting corresponding rows from $\mathcal{H}_t$ by node indices.

Layout Learner

This module aims to learn both the implicit environment layout distribution and visual commonsense knowledge of the room type, which is achieved through an auxiliary layout prediction task with our room type codebook. This auxiliary task is not used directly at inference time. The main purpose of having it during training is learning representations to capture this information, which in turn improves global action prediction.

Building Room Type Codebook. We fetch room type labels from the MatterPort point-wise semantic annotations which contain 30 distinct room types. We then select the large-scale pre-trained text-to-image generation model GLIDE (Nichol et al. 2022) as a visual commonsense resource. To better fit the embodied grounding task, we create prompt P_{room} to prompt visual commonsense knowledge not only based on the room type label but including high-frequency objects of referring expressions in the training set. Specifically, when building the room type codebook, we create prompts by filling in the following template.

A [room type] with [obj 1], ... and [obj n].

where [room type] is a room label annotated in Matterport3D dataset (Chang et al. 2017) with manual spelling completion, such as map "l" to "living room" and "u" to "utility room". [obj 1] .. [obj n] are high-frequency object words that co-occur with specific room labels in the training instructions of the REVERIE dataset. A frequency above 0.8 is considered a high frequency. This threshold ensures diversity and limits the number of candidates. For instance, we create the prompt "A dining room with table and chairs" for room type "dining room", "a bathroom with towel and mirror" for room type "bathroom". For each room type, we generate a hundred images and select S representative ones in the pre-trained CLIP feature space (i.e., the image closest to each clustering center after applying a K-Means cluster algorithm). An example is shown in Fig. 3. Our selection strategy guarantees the diversity of the generated images, i.e., rooms from various perspectives and lighting conditions. The visual features of the selected images for different room types form the room type codebook $E_{room} \in \mathbb{R}^{K \times S \times 765}$, where K is the total number of room types and S represents the number of images for each room type. This codebook represents a commonsense knowledge base with visual descriptions of what a room should look like.

Environment Layout Prediction. Layout information is critical for indoor navigation, especially when the agent only receives high-level instructions describing the goal positions, such as "Go to the kitchen and pick up the mug beside the sink". This module equips the agent with both the capability of learning room-to-room correlations in the environment layout and a generalized room type representation. With the help of the visual commonsense knowledge of the rooms in the room type codebook, we perform layout prediction. We compute the similarity score between node representations $\tilde{\mathcal{H}}_t^{(glo)}$ and image features E_{room} in the room type codebook and further use this score to predict the room type of each node in the graph $\mathcal{G}_t$. The predicted room type logits are supervised with ground truth labels $\mathcal{C}_t$.

$$\hat{\mathcal{C}_t^i} = \sum_{j=1}^{S} \tilde{\mathcal{H}}_t^{(glo)} E_{room(i,j)} \tag{3}$$

$$\mathcal{L}_t^{(\text{LP})} = \text{CrossEntropy}(\hat{\mathcal{C}}_t, \mathcal{C}_t) \tag{4}$$

where S is the number of images in the room type codebook for each room type, and $\hat{\mathcal{C}_t^i}$ represents the predicted score of ith room type. We use $\hat{\mathcal{C}}_t$ to denote the predicted score distribution of a node, thus $\hat{\mathcal{C}}_t = [\hat{\mathcal{C}_t^0}, \cdots, \hat{\mathcal{C}_t^K}]$.

Goal Dreamer

A navigation agent without a global map can be short-sighted. We design a long-horizon value function to guide the agent toward the imagined destination. For each instruction, we prompt five images from GLIDE (Nichol et al.

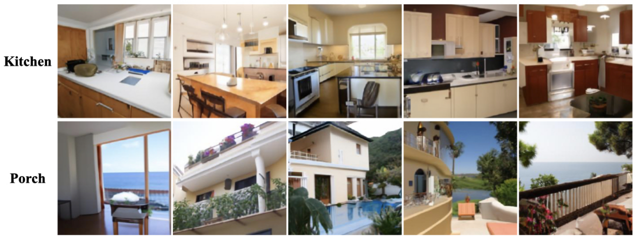

Figure 3: Prompted examples of the room codebook.

2022) as the imagination of the destination. Three examples are shown in Fig 4. Imagination features $E^{(im)}$ are extracted from the pre-trained CLIP vision encoder (Radford et al. 2021). Then at each time step t, we attend the topological global node embeddings $\tilde{\mathcal{H}}_t^{(glo)}$ to $E^{(im)}$ through a cross-attention layer (Vaswani et al. 2017).

$$\hat{\mathcal{H}}_t^{(glo)} = \text{Cross-Attn}(\tilde{\mathcal{H}}_t^{(glo)}, E^{(im)}, E^{(im)}) \tag{5}$$

The hidden state $\hat{\mathcal{H}}_t^{(glo)}$ learned by the Goal Dreamer is projected by a linear feed-forward network (FFN)[1] to predict the probability distribution of the next action step over all navigable but not visited nodes.

$$Pr_t^{(im)} = \text{Softmax}(\text{FFN}(\hat{\mathcal{H}}_t^{(glo)})) \tag{6}$$

We supervise this distribution $Pr_t^{(im)}$ in the warmup stage of the training (see next Section) with the ground truth next action $\mathcal{A}_{gt}$.

$$\mathcal{L}_t^{(D)} = \text{CrossEntropy}(Pr_t^{(im)}, \mathcal{A}_{gt}) \tag{7}$$

Optimizing $Pr_t^{(im)}$ guides the learning of latent features $\hat{\mathcal{H}}_t^{(glo)}$. $\hat{\mathcal{H}}_t^{(glo)}$ will be fused with global logits in the final decision process as described in the following section.

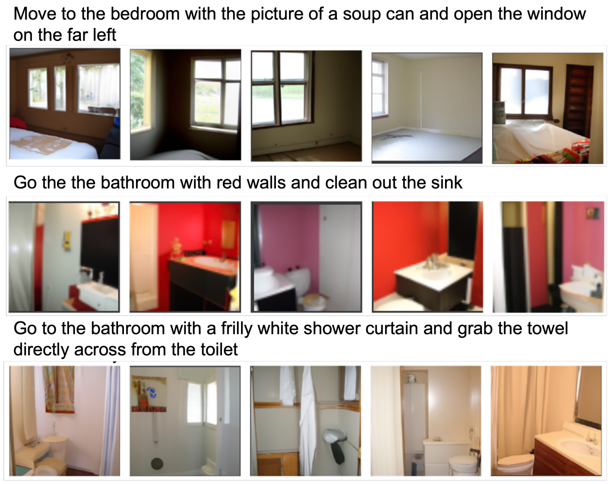
Move to the bedroom with the picture of a soup can and open the window on the far left
Go the the bathroom with red walls and clean out the sink
Go to the bathroom with a frilly white shower curtain and grab the towel directly across from the toilet

Figure 4: Images of the destination generated by GLIDE based on the given instruction.

Decision Maker

Action Prediction. We follow the work of Chen et al. (2022) to predict the next action to be performed in both the global and local branches and dynamically fuse their results to enable the agent to backtrack to previous unvisited nodes.

$$Pr_t^{(\text{floc})} = \text{DynamicFuse}(\tilde{\mathcal{H}}_t^{(loc)}, \tilde{\mathcal{H}}_t^{(glo)}) \tag{8}$$

The proposed goal dreamer module equips the agent with the capability of learning guidance towards the target goal, hence we further fuse the goal dreamer's latent features $\hat{\mathcal{H}}_t^{(glo)}$ with global results weighted by a learnable λ_t. The λ_t is node-specific; thus we apply a feed-forward network (FFN) to predict these weights conditioned on node representations.

$$\lambda_t = \text{FFN}([\tilde{\mathcal{H}}_t^{(glo)}; \hat{\mathcal{H}}_t^{(glo)}]) \tag{9}$$

The fused action distribution is formulated as:

$$\begin{aligned} Pr_t^{(\text{fgd})} = (1 - \lambda_t) * \text{FFN}(\tilde{\mathcal{H}}_t^{(glo)}) \\ + \lambda_t * \text{FFN}(\hat{\mathcal{H}}_t^{(glo)}) \end{aligned} \tag{10}$$

The objective for supervising the whole decision procedure by ground truth node $\mathcal{A}_{gt}$ in the next time step is:

$$\begin{aligned} Pr_t^{(\text{DSAP})} &= Pr_t^{(\text{floc})} + Pr_t^{(\text{fgd})} \\ \mathcal{L}_t^{(\text{DSAP})} &= \text{CrossEntropy}(Pr_t^{(\text{DSAP})}, \mathcal{A}_{gt}) \end{aligned} \tag{11}$$

where $Pr_t^{(\text{DSAP})}$ is the estimated single-step prediction distribution (DSAP) over all nodes. Not only the global-local fusion already proposed in previous work but also our goal-dreaming branch is now included to predict the next action.
Object Grounding. We simply consider object grounding as a classification task and use a FFN to generate a score for each object in $\mathcal{O}_t$ of the current node. We then supervise this score with the annotated ground truth object $\mathcal{O}_{gt}$.

$$\begin{aligned} \hat{\mathcal{O}}_t &= \text{FFN}(\mathcal{O}_t) \\ \mathcal{L}_t^{(\text{OG})} &= \text{CrossEntropy}(\hat{\mathcal{O}}_t, \mathcal{O}_{gt}) \end{aligned} \tag{12}$$

Training and Inference

Warmup stage. Previous researches (Chen et al. 2021b, 2022; Pashevich, Schmid, and Sun 2021; Hao et al. 2020) have shown that warming up the model with auxiliary supervised or self-supervised learning tasks can significantly boost the performance of a transformer-based VLN agent. We warm up our model with five auxiliary tasks, including three common tasks in vision-and-language navigation: masked language modeling (MLM) (Devlin et al. 2019), masked region classification (MRC) (Lu et al. 2019), object grounding (OG) (Lin, Li, and Yu 2021) if object annotations exist; and two new tasks, that is, layout prediction (LP) and single action prediction with the dreamer (DSAP) explained in sections Layout Learner and Goal Dreamer, respectively. In the LP, our agent predicts the room type of each node in the topological graph at each time step, aiming to model the room-to-room transition of the environment, the objective $\mathcal{L}_t^{(\text{LP})}$ of which is shown in Eq. 4. To encourage the agent

[1] FFNs in this paper are independent without parameter sharing.

to conduct goal-oriented exploration, in the DSAP, as illustrated in $\mathcal{L}_t^{(\mathrm{D})}$ (Eq. 7) and $\mathcal{L}_t^{(\mathrm{DSAP})}$ (Eq. 11) we use the output of the goal dreamer to predict the next action. The training objective of the warmup stage is as follows:

$$\begin{aligned}\mathcal{L}_t^{\mathrm{WP}} = &\mathcal{L}_t^{(\mathrm{MLM})} + \mathcal{L}_t^{(\mathrm{MRC})} + \mathcal{L}_t^{(\mathrm{OG})} + \\ &\mathcal{L}_t^{(\mathrm{LP})} + \mathcal{L}_t^{(\mathrm{D})} + \mathcal{L}_t^{(\mathrm{DSAP})}\end{aligned} \tag{13}$$

Imitation Learning and Inference. We use the imitation learning method DAgger (Ross, Gordon, and Bagnell 2011) to further train the agent. During training, we use the connectivity graph $\mathcal{G}$ of the environment to select the navigable node with the shortest distance from the current node to the destination as the next target node. We then use this target node to supervise the trajectory sampled using the current policy at each iteration. The training objective here is:

$$\mathcal{L}_t^{\mathrm{IL}} = \mathcal{L}_t^{(\mathrm{OG})} + \mathcal{L}_t^{(\mathrm{LP})} + \mathcal{L}_t^{(\mathrm{DSAP})} \tag{14}$$

During inference, our agent builds the topological map on-the-fly and selects the action with the largest probability. If the agent decides to backtrack to the previous unexplored nodes, the classical Dijkstra algorithm (Dijkstra 1959) is used to plan the shortest path from the current node to the target node. The agent stops either when a stop action is predicted at the current location or when it exceeds the maximum action steps. When the agent stops, it selects the object with the maximum object prediction score.

Experiments

Datasets

Because the navigation task is characterized by realistic high-level instructions, we conduct experiments and evaluate our agent on the embodied goal-oriented benchmark REVERIE (Qi et al. 2020) and the SOON (Song et al. 2022) datasets.

REVERIE dataset: The dataset is split into four sets, including the training set, validation seen set, validation unseen set, and test set. The environments in both validation unseen and test sets do not appear in the training set, while all environments in validation seen have been explored or partially observed during training. The average length of instructions is 18 words. The dataset also provides object bounding boxes for each panorama, and the length of ground truth paths ranges from 4 to 7 steps.

SOON dataset: This dataset has a similar data split as REVERIE. The only difference is that it proposes a new validation on the seen instruction split which contains the same instructions in the same house but with different starting positions. Instructions in SOON contain 47 words on average, and the ground truth paths range from 2 to 21 steps with 9.5 steps on average. The SOON dataset does not provide bounding boxes for object grounding, thus we use here an existing object detector (Anderson et al. 2018b) to generate candidate bounding boxes.

Evaluation Metrics

Navigation Metrics. Following previous work (Chen et al. 2022; Anderson et al. 2018a), we evaluate the navigation performance of our agent using standard metrics, including Trajectory Length (TL) which is the average path length in meters; Success Rate (SR) defined as the ratio of paths where the agent's location is less than 3 meters away from the target location; and SR weighted by inverse Path Length (SPL).

Object Grounding Metrics. We follow the work of Qi et al. (2020) using Remote Grounding Success (RGS), which is the ratio of successfully executed instructions, and RGS weighted by inverse Path Length (RGSPL).

Implementation Details

The model is trained for 100k iterations with a batch size of 32 for single action prediction and 50k iterations with a batch size of 8 for imitation learning with DAgger (Ross, Gordon, and Bagnell 2011). We optimize both phases by the AdamW (Loshchilov and Hutter 2018) optimizer with a learning rate of 5e-5 and 1e-5, respectively. We include two fixed models for preprocessing data, i.e., GLIDE (Nichol et al. 2022) for generating the room codebook and the imagined destination, and CLIP (Radford et al. 2021) for image feature extraction. The whole training procedure takes two days with a single NVIDIA-P100 GPU.

Results

Comparisons to the State of the Art

Results on REVERIE. In Table 1, we compare our model with prior works in four categories: (1) Imitation Learning + Reinforcement learning models: RCM (Wang et al. 2019), SIA (Lin, Li, and Yu 2021); (2) Supervision model: Self-Monitor (Ma et al. 2019), REVERIE (Qi et al. 2020); (3) Imitation Learning with external knowledge graph: CKR (Gao et al. 2021b); and (4) Imitation Learning with topological memory: VLN-DUET (Chen et al. 2022). Our model outperforms the above models with a large margin in challenging unseen environments. Significantly, our model surpasses the previous state of the art VLN-DUET by approximately 10% (SR) and 5% (RGS) in the val-unseen split. On the test split, our model beats VLN-DUET with improvements of SR by 4.02% and RGS by 3.43%. The results demonstrate that the proposed LAD better generalizes to unseen environments, which is critical for real applications.

Results on SOON. Table 2 presents the comparison of our proposed LAD with other models including the state-of-the-art VLN-DUET model. The LAD model significantly outperforms VLN-DUET across all evaluation metrics in the challenging test unseen split. Especially, the model improves the performance on SR and SPL by 6.15% and 6.4%, respectively. This result clearly shows the effectiveness of the proposed Layout Learner and Goal Dreamer modules.

Ablation Studies

We verify the effectiveness of our key contributions via an ablation study on the REVERIE dataset.

Methods	Val-seen					Val-unseen					Test-unseen				
	Navigation			Grounding		Navigation			Grounding		Navigation			Grounding	
	TL ↓	SR↑	SPL↑	RGS↑	RGSPL↑	TL ↓	SR↑	SPL↑	RGS↑	RGSPL↑	TL ↓	SR↑	SPL↑	RGS↑	RGSPL↑
RCM	10.70	23.33	21.82	16.23	15.36	11.98	9.29	6.97	4.89	3.89	10.60	7.84	6.67	3.67	3.14
SelfMonitor	7.54	41.25	39.61	30.07	28.98	9.07	8.15	6.44	4.54	3.61	9.23	5.80	4.53	3.10	2.39
REVERIE	16.35	50.53	45.50	31.97	29.66	45.28	14.40	7.19	7.84	4.67	39.05	19.88	11.61	11.28	6.08
CKR	12.16	57.27	53.57	39.07	-	26.26	19.14	11.84	11.45	-	22.46	22.00	14.25	11.60	-
SIA	13.61	61.91	57.08	45.96	42.65	41.53	31.53	16.28	22.41	11.56	48.61	30.8	14.85	19.02	9.20
VLN-DUET	13.86	**71.75**	**63.94**	**57.41**	**51.14**	22.11	46.98	33.73	32.15	22.60	21.30	52.51	36.06	31.88	22.06
LAD (Ours)	16.74	69.22	57.44	52.92	43.46	26.39	**57.00**	**37.92**	**37.80**	**24.59**	25.87	**56.53**	**37.8**	**35.31**	**23.38**

Table 1: Results obtained on the REVERIE dataset as compared to other existing models including the current state-of-the-art model VLN-DUET.

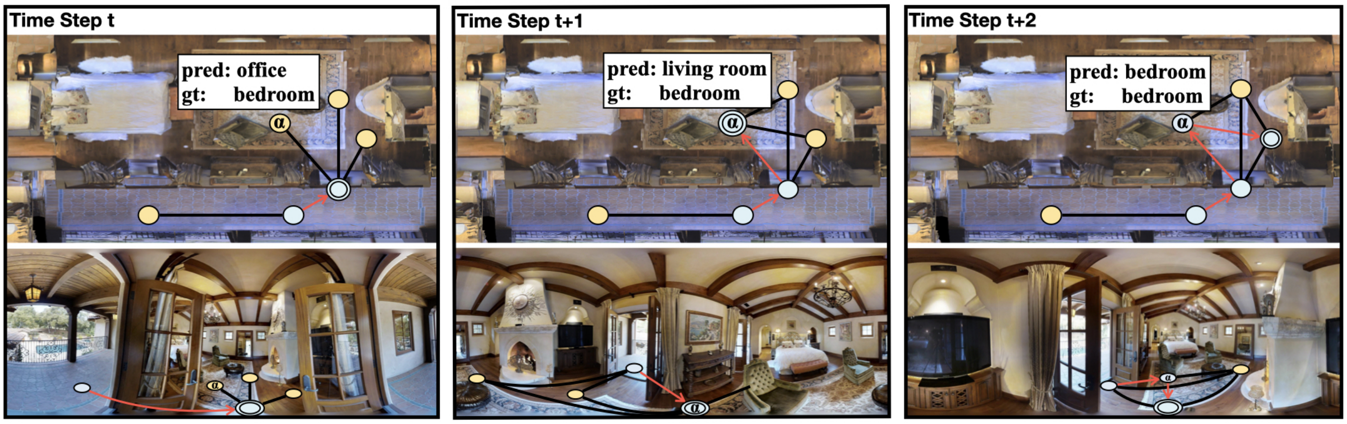

Figure 5: Room type belief will be updated while exploring. Blue double circle denotes the current location, a yellow circle refers to an unexplored but visible node, a blue circle represents a visited node, the red line is the agent's trajectory. Each column contains bird-view and egocentric views of current agent states.

Split	Methods	TL ↓	SR ↑	SPL ↑	RGSPL ↑
Val Unseen	GBE	28.96	19.52	13.34	1.16
	VLN-DUET	36.20	36.28	22.58	3.75
	LAD	32.32	**40.24**	**29.44**	**4.20**
Test Unseen	GBE	27.88	12.90	9.23	0.45
	VLN-DUET	41.83	33.44	21.42	4.17
	LAD	35.71	**39.59**	**27.82**	**7.08**

Table 2: Results of our LAD model obtained on the SOON dataset compared with the results of other state-of-the-art models.

Base	Layout Learner	Goal Dreamer	SR↑	SPL↑	RGS↑	RGSPL↑
✓			52.34	34.45	35.02	22.87
✓	✓		56.04	37.66	37.06	24.58
✓		✓	53.45	37.41	34.34	24.03
✓	✓	✓	57.00	37.92	37.80	24.59

Table 3: Comparisons of baseline model and baseline with our proposed modules.

Are the Layout Learner and Goal Dreamer helpful? We first verify the contribution of the Layout Learner and Goal Dreamer. For a fair comparison, we re-implement the result of VLN-DUET (Chen et al. 2022) but replace the visual representations to CLIP features and report the results in Row 1 of Table 3. The performance boost in this re-implementation compared to VLN-DUET results in Table 1 indicates that CLIP features are more suitable for visual-language tasks than original ImageNet ViT features. Comparing the results in row 2 with row 1, it is clear that integrating the Layout Learner to the baseline model improves its performance across all evaluation metrics, which verifies our assumption that the layout information is vital in high-level instruction following tasks. One might notice that in row 3 the Goal Dreamer module can boost the performance in SR, SPL, and RGSPL, but it slightly harms the performance in RGS. A lower RGS but higher RGSPL shows that the model with Goal Dreamer takes fewer steps to reach the goal, meaning that it conducts more effective goal-oriented exploration, which supports our assumption.

Visual or textual common sense? In this work, we consider several images to describe a commonsense concept. In this experiment, we study whether visual descriptors of room types lead to a better generalization than directly using the classification label or a textual description while learning an agent. In the first line of Table 4, we show the results of directly replacing the visual codebook module with a room label classification head. It shows a 3% drop in both navigation and grounding success rates. This indicates that a single room type classification head is insufficient for learning good latent features of room concepts. We further compare the results of using a visual codebook with using

FFN	Text	Visual	SR↑	SPL↑	RGS↑	RGSPL↑
✓			53.73	35.22	34.68	23.58
	✓		51.38	35.57	32.38	22.05
		✓	57.00	37.92	37.80	24.59

Table 4: Codebook type comparison: visual room codebook versus textual room codebook and direct classification head.

a textual codebook. Since we use text to prompt multiple room images as our visual room codebook encoded with the CLIP (Radford et al. 2021) visual encoder, for a fair comparison, we encode the text prompts as a textual codebook using the CLIP text encoder. Then we replace the visual codebook in our model with the textual one and re-train the whole model. As shown in Table 4, the textual codebook has a 5.62% drop in navigation success rate (SR) and a 5.58% drop in remote grounding success rate (RGS). This indicates that visual descriptors of commonsense room concepts are informative and easier to follow for an autonomous agent.

Could the room type prediction be corrected while exploring more? In this section, we study the predicted trajectory. As shown in Fig. 5, the incorrect room type prediction of node α is corrected after exploration of the room. At time step t, the observation only contains chairs, the prediction of room type of node α is office. When entering the room at time step $t+1$, the table and television indicate this room is more likely to be a living room. While grabbing another view from a different viewpoint, the room type of node α is correctly recognized as a bedroom. Since the instruction states to find the pillow inside the bedroom, the agent could correctly track its progress with the help of room type recognition and successfully execute the instruction. This indicates that the ability to correct former beliefs benefits the layout understanding of the environment and further has a positive influence on the action decision process. We further discuss the room type correction ability quantitatively. The following Fig. 6 shows the room type recognition accuracy w.r.t. time step t in the validation unseen set of the REVERIE dataset. It shows that room type recognition accuracy increases with increased exploration of the environment. We also observe that the overall accuracy of the room type recognition is still not satisfactory. We assume the following main reasons: first, room types defined in MatterPort3D have ambiguity, such as family room and living room do not have a well-defined difference; second, many rooms do not have a clear boundary in the visual input (no doors), so it is hard to distinguish connected rooms from the observations. These ambiguities require softer labels while learning, which is also a reason why using images as commonsense resource performs better than using textual descriptors and linear classification heads as is seen in Table 4.

Limitations and Future Work

In this paper, we describe our findings while including room type prediction and destination imagination in the Embodied Referring Expression Grounding task, but several limitations still require further study.

Imagination is not dynamic and it is only conditioned on the given instruction. Including observations and dynamically modifying the imagination with a trainable generation module could be helpful for fully using the knowledge gained during exploration. This knowledge could guide the imagination model to generate destination images of a style similar to the environment. It is also possible to follow the idea of PathDreamer (Koh et al. 2021) and Dreamer (Hafner et al. 2019, 2020), which generate a sequence of hidden future states based on the history to enhance reinforcement learning models.

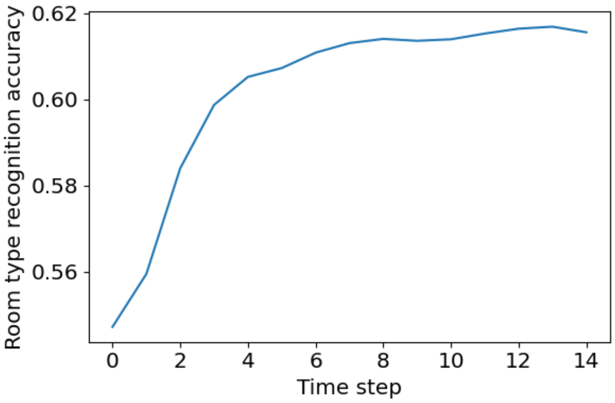

Figure 6: Room recognition accuracy of the validation unseen set of the REVERIE dataset.

Constant number of generated visual features. Due to the long generation time and storage consumption, we only generate five images as the goal imaginations. It is possible to increase diversity by generating more images. Then, a better sampling strategy for the visual room codebook construction and destination imagination could be designed, such as randomly picking a set of images from the generated pool. Since we have observed overfitting in the later stage of the training, it is possible to further improve the generalization of the model by including randomness in this way.

Conclusion

In this work, to enhance the environmental understanding of an autonomous agent in the Embodied Referring Expression Grounding task, we have proposed a Layout Learner and Goal Dreamer. These two modules effectively introduce visual common sense, in our case via an image generation model, into the decision process. Extensive experiments and case studies show the effectiveness of our designed modules. We hope our work inspires further studies that include visual commonsense resources in autonomous agent design.

Acknowledgements

This research received funding from the Flanders AI Impuls Programme - FLAIR and from the European Research Council Advanced Grant 788506.

References

Anderson, P.; Chang, A.; Chaplot, D. S.; Dosovitskiy, A.; Gupta, S.; Koltun, V.; Kosecka, J.; Malik, J.; Mottaghi, R.; Savva, M.; et al. 2018a. On Evaluation of Embodied Navigation Agents. *arXiv preprint arXiv:1807.06757*.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018b. Bottom-up and Top-down Attention for Image Captioning and Visual Question Answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6077–6086.

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; and Van Den Hengel, A. 2018c. Vision-and-language Navigation: Interpreting Visually-grounded Navigation Instructions in Real Environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3674–3683.

Blukis, V.; Paxton, C.; Fox, D.; Garg, A.; and Artzi, Y. 2022. A Persistent Spatial Semantic Representation for High-level Natural Language Instruction Execution. In *Conference on Robot Learning*, 706–717. PMLR.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language Models are Few-shot Learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niessner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. *International Conference on 3D Vision (3DV)*.

Chaplot, D. S.; Gandhi, D.; Gupta, A.; and Salakhutdinov, R. 2020a. Object Goal Navigation using Goal-Oriented Semantic Exploration. In *In Neural Information Processing Systems*.

Chaplot, D. S.; Gandhi, D.; Gupta, S.; Gupta, A.; and Salakhutdinov, R. 2020b. Learning To Explore Using Active Neural SLAM. In *International Conference on Learning Representations (ICLR)*.

Chen, K.; Chen, J. K.; Chuang, J.; Vázquez, M.; and Savarese, S. 2021a. Topological Planning with Transformers for Vision-and-language Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11276–11286.

Chen, S.; Guhur, P.-L.; Schmid, C.; and Laptev, I. 2021b. History Aware Multimodal Transformer for Vision-and-language Navigation. *Advances in Neural Information Processing Systems*, 34: 5834–5847.

Chen, S.; Guhur, P.-L.; Tapaswi, M.; Schmid, C.; and Laptev, I. 2022. Think Global, Act Local: Dual-scale Graph Transformer for Vision-and-Language Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16537–16547.

Cirik, V.; Morency, L.-P.; and Berg-Kirkpatrick, T. 2022. HOLM: Hallucinating Objects with Language Models for Referring Expression Recognition in Partially-Observed Scenes. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5440–5453.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*.

Dijkstra, E. W. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische mathematik*, 1(1): 269–271.

Gao, C.; Chen, J.; Liu, S.; Wang, L.; Zhang, Q.; and Wu, Q. 2021a. Room-and-Object Aware Knowledge Reasoning for Remote Embodied Referring Expression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3064–3073.

Gao, C.; Chen, J.; Liu, S.; Wang, L.; Zhang, Q.; and Wu, Q. 2021b. Room-and-object Aware Knowledge Reasoning for Remote Embodied Referring Expression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3064–3073.

Georgakis, G.; Schmeckpeper, K.; Wanchoo, K.; Dan, S.; Miltsakaki, E.; Roth, D.; and Daniilidis, K. 2022. Cross-modal Map Learning for Vision and Language Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15460–15470.

Hafner, D.; Lillicrap, T.; Ba, J.; and Norouzi, M. 2019. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations*.

Hafner, D.; Lillicrap, T. P.; Norouzi, M.; and Ba, J. 2020. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations*.

Hahn, M.; Chaplot, D. S.; Tulsiani, S.; Mukadam, M.; Rehg, J. M.; and Gupta, A. 2021. No rl, No Simulation: Learning to Navigate without Navigating. *Advances in Neural Information Processing Systems*, 34: 26661–26673.

Hao, W.; Li, C.; Li, X.; Carin, L.; and Gao, J. 2020. Towards Learning a Generic Agent for Vision-and-language Navigation via Pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13137–13146.

Hong, Y.; Wu, Q.; Qi, Y.; Rodriguez-Opazo, C.; and Gould, S. 2021. A Recurrent Vision-and-Language BERT for Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1643–1653.

Irshad, M. Z.; Mithun, N. C.; Seymour, Z.; Chiu, H.-P.; Samarasekera, S.; and Kumar, R. 2022. SASRA: Semantically-aware Spatio-temporal Reasoning Agent for Vision-and-Language Navigation in Continuous Environments.

Koh, J. Y.; Lee, H.; Yang, Y.; Baldridge, J.; and Anderson, P. 2021. Pathdreamer: A World Model for Indoor Navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large Language Models are Zero-Shot Reasoners. *arXiv preprint arXiv:2205.11916*.

Krantz, J.; Wijmans, E.; Majumdar, A.; Batra, D.; and Lee, S. 2020. Beyond the Nav-graph: Vision-and-language Navigation in Continuous Environments. In *European Conference on Computer Vision*, 104–120. Springer.

Lin, X.; Li, G.; and Yu, Y. 2021. Scene-intuitive Agent for Remote Embodied Visual Grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7036–7045.

Liu, H.; and Singh, P. 2004. ConceptNet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4): 211–226.

Loshchilov, I.; and Hutter, F. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Vilbert: Pretraining Task-agnostic Visiolinguistic Representations for Vision-and-language Tasks. *Advances in neural information processing systems*, 32.

Ma, C.-Y.; Lu, J.; Wu, Z.; AlRegib, G.; Kira, Z.; Socher, R.; and Xiong, C. 2019. Self-monitoring Navigation Agent via Auxiliary Progress Estimation. *arXiv preprint arXiv:1901.03035*.

Majumdar, A.; Shrivastava, A.; Lee, S.; Anderson, P.; Parikh, D.; and Batra, D. 2020. Improving Vision-and-language Navigation with Image-text Pairs from the Web. In *European Conference on Computer Vision*, 259–274. Springer.

Min, S. Y.; Chaplot, D. S.; Ravikumar, P. K.; Bisk, Y.; and Salakhutdinov, R. 2021. FILM: Following Instructions in Language with Modular Methods. In *International Conference on Learning Representations*.

Nichol, A. Q.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; Mcgrew, B.; Sutskever, I.; and Chen, M. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 16784–16804. PMLR.

Pashevich, A.; Schmid, C.; and Sun, C. 2021. Episodic Transformer for Vision-and-language Navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15942–15952.

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; and Miller, A. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2463–2473.

Qi, Y.; Wu, Q.; Anderson, P.; Wang, X.; Wang, W. Y.; Shen, C.; and van den Hengel, A. 2020. REVERIE: Remote Embodied Visual Referring Expression in Real Indoor Environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qiao, Y.; Qi, Y.; Hong, Y.; Yu, Z.; Wang, P.; and Wu, Q. 2022. HOP: History-and-Order Aware Pre-training for Vision-and-Language Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15418–15427.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical Text-conditional Image Generation with Clip Latents. *arXiv preprint arXiv:2204.06125*.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. A Reduction of Imitation Learning and Structured Prediction to No-regret Online Learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 627–635. JMLR Workshop and Conference Proceedings.

Song, C. H.; Kil, J.; Pan, T.-Y.; Sadler, B. M.; Chao, W.-L.; and Su, Y. 2022. One Step at a Time: Long-Horizon Vision-and-Language Navigation with Milestones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15482–15491.

Su, W.; Zhu, X.; Cao, Y.; Li, B.; Lu, L.; Wei, F.; and Dai, J. 2019. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *International Conference on Learning Representations*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: A Free Collaborative Knowledge Base. *Communications of the ACM*, 57(10): 78–85.

Wang, X.; Huang, Q.; Celikyilmaz, A.; Gao, J.; Shen, D.; Wang, Y.-F.; Wang, W. Y.; and Zhang, L. 2019. Reinforced Cross-modal Matching and Self-supervised Imitation Learning for Vision-language Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6629–6638.

Wang, Z.; Li, M.; Wu, M.; Moens, M.-F.; and Tuytelaars, T. 2022. Find a Way Forward: a Language-Guided Semantic Map Navigator. *arXiv preprint arXiv:2203.03183*.

Zhu, F.; Liang, X.; Zhu, Y.; Yu, Q.; Chang, X.; and Liang, X. 2021. Soon: Scenario Oriented Object Navigation with Graph-based Exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12689–12699.