Supplementary Material for "Meta-Explore: Exploratory Hierarchical Vision-and-Language Navigation Using Scene Object Spectrum Grounding"

Minyoung Hwang[1], Jaeyeon Jeong[1], Minsoo Kim[3], Yoonseon Oh[2], Songhwai Oh[1]
[1]Electrical and Computer Engineering and ASRI, Seoul National University
[2]Department of Electronic Engineering, Hanyang University
[3]Interdisciplinary Major in Artificial Intelligence, Seoul National University
{minyoung.hwang, jaeyeon.jeong}@rllab.snu.ac.kr, goldbird5@snu.ac.kr, yoh21@hanyang.ac.kr, songhwai@snu.ac.kr

We provide additional details and analyses of the proposed method in this supplementary material. Section A provides model details. Section B provides detailed settings and data preprocessing for experiments. Section C provides evaluation results with detailed analyses. Section D provides implementation details and detailed results for the ablation study.

A. Model Details

A.1. Algorithm Details

Algorithm 1 summarizes the overall hierarchical exploration process. The mode selector supervises the process and chooses whether the agent should explore or exploit at each time step.

A.2. Exploitation Module

A.2.1 Reference SOS Features.

In the proposed method, we approximate the reference SOS feature of an object token by using prior information about objects in the training data. For instance, for the 'chair' object, we collected the widths and heights of the detected bounding boxes as shown in Figure 1. Figure 2 shows two representative values: median and mean for each distribution. We choose the median values, which minimizes the L1 error, to represent the reference bounding box of each object. To generate rotation-invariant SOS features, we convert the four vertices of the bounding box detected from the front view image of size 640×480 to the vertices of a bounding box detected from the panoramic view image of size 2048×512 using coordinate transformations. To simplify the implementation, we assume that the converted bounding box has a rectangle shape with the vertices transformed into coordinates in a panoramic view. The reference SOS feature is calculated as the logarithmic magnitude of the Fourier transform of the panoramic mask with mean pooling on the vertical spectral axis. Considering that the shift in the spatial-domain only affects the phase of the Fourier transform, the location of a reference bounding box does not matter.

Algorithm 1: Meta-Explore

```
$P_{explore} \leftarrow 1$
$Success \leftarrow False$
Initialize $G_t$ and node features
while $t < T$ do
    Update $G_t$
    Update node features
    $H_t \leftarrow$ cross-modal embedding at time $t$
    if $P_{explore} \geq 0.5$ then
        $a_t \leftarrow \arg\max_{V_i}(F_{explore}([H_t]_i))$
        $\hat{p}_t \leftarrow F_{progress}(H_t)$
        $t \leftarrow t + 1$
    else
        $V_{local} \leftarrow$ unvisited but observed nodes in $G_t$
        $v_{local} \leftarrow \arg\max_{v' \in V_{local}}(S_{nav}(\tau'(v_0, v')))$
        $\tau \leftarrow PathPlanning(v_t, v_{local})$
        while not arrived at $v_{local}$ do
            $a_t \leftarrow pop(\tau)$
            $t \leftarrow t + 1$
        end
    end
    $P_{explore} \leftarrow 1 - S_{mode}(H_t)$
    if $a_t$ is $stop$ and $d(v_t, v_{goal}) < d_{success}$ then
      $Success \leftarrow True$
end
```

A.2.2 Navigation Score

To compare local goal candidates, we design a navigation score of a corrected trajectory τ' as equation 1. This metric can also be interpreted as a weighted correlation coefficient among SOS features and object tokens weighted by the similarities between them.

Dataset	Instruction	Object Tokens	Target Object
R2R	“Walk through the kitchen. Go past the sink and stove stand in front of the dining table on the bench side.”	[“kitchen”, “sink”, “stove”, “stand”, “dish”, “table”, “chair”]	“chair”
SOON	“This is a brand new white, rectangular wooden table, which is above a few chairs, under a pot of flowers. It is in a very neat study with many books.”	[“book”, “chair”, “pitcher”, “flower”, “table”, “table”]	“table”
REVERIE	“Go to the bedroom with the fireplace and bring me the lowest hanging small picture on the right wall across from the bedside table with the lamp on it”	[“bedroom”, “fireplace”, “bed”, “table”, “stand”, “art”]	“art”

Table 1. **Object Parsing Examples.** For each dataset, object tokens are extracted from the instructions. Target objects are inferred from the instructions using VQA. Words that have similar meanings are unified into a single object word for categorization.

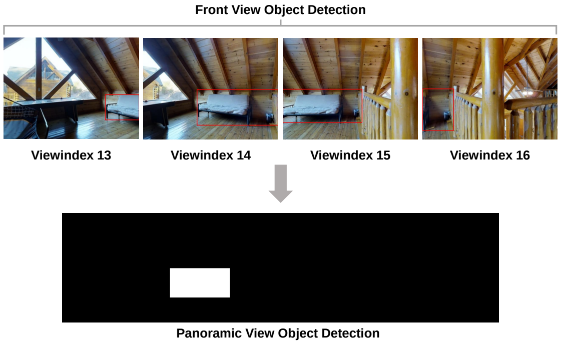

Figure 1. **Bounding box coordinate transformation.** Front-view visual observations from different angles at the same location. Each bounding box shows the ‘chair’ detection. We use coordinate transformation to convert coordinates into panoramic view.

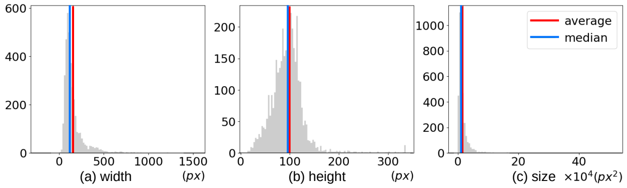

Figure 2. **Bounding box statistics.** We collect width, height, and size of detected bounding boxes. The histograms show statistics for ‘chair’ objects. Yellow line and red line show the median and average values of each distribution, respectively.

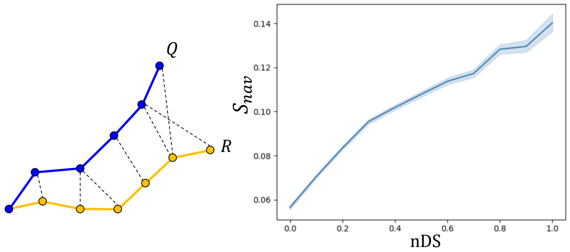

Figure 3. **Relationship between navigation score (S_{nav}) and normalized distance sum (nDS) in R2R.** We measure navigation scores for augmented trajectories which include both successful and failed trajectories. R and Q illustrate an example case of a ground truth trajectory and a query trajectory. Maximum hop of a query trajectory is 15. Trajectories with high nDS scores also have high navigation scores.

$$S_{nav}(\mathcal{T}') = \frac{\sum_{i=1}^{B}\sum_{j=1}^{t'}(\frac{\hat{\delta}(w_i^o)}{|\hat{\delta}(w_i^o)|}\cdot\frac{\vec{S}'_j}{|\vec{S}'_j|})((\hat{\delta}(w_i^o)-\hat{\delta}(\overline{w}^o))\cdot(\vec{S}'_j-\overline{\vec{S}}'))}{\sqrt{\frac{t'}{B}\cdot\sum_{i=1}^{B}(\hat{\delta}(w_i^o)-\hat{\delta}(\overline{w}^o))^2\sum_{j=1}^{t'}(\vec{S}'_j-\overline{\vec{S}}')^2}} \tag{1}$$

Figure 3 shows the relationship between the navigation score and an evaluation metric in the R2R navigation task. Both metrics measure how similar the current trajectory is to the ground truth trajectory. We generate 49,986 augmented trajectories with an average length of $8.23m$ based on 5596 ground truth trajectories. To generate various samples, we separate each augmented trajectory $(v_1, v_2, ..., v_t)$ into t augmented trajectories $(v_1), (v_1, v_2), ...,$ and $(v_1, v_2, ..., v_t)$. The final 421,383 augmented trajectories include trajectories with 1 to 15 nodes and include both successful and unsuccessful trajectories. We classify the trajectories with the normalized distance sum (nDS) between ground truth trajectory R and a query trajectory Q as follows:

$$\text{nDS}(R,Q) = \exp\left(-\frac{\sum_{v_i\in R}\min_{u_j\in Q} d(v_i,u_j) + \sum_{u_j\in Q}\min_{v_i\in R} d(u_j,v_i)}{\frac{|R|+|Q|}{2}d_{success}}\right), \tag{2}$$

which requires the ground truth information of R. $d(u,v)$ denotes the geodesic distance between two nodes, u and v, and $d_{success}$ denotes the success distance. The plot in Figure 3 shows a linear relationship between the nDS and the navigation score. The results imply that the proposed navigation score effectively scores the augmented trajectories even though it only relies on the given target instruction and observation from the augmented paths, without any location information about the nodes on the ground truth trajectory.

A.3. Implementation Details

We use ViT-B/16 [1] pretrained on ImageNet to extract features from the viewpoint panoramic images. We use pretrained LXMERT [2] for the language encoder and cross-modal transformer. We implement the mode selector as a two-layer feed-forward network.

B. Experiment Setup

B.1. Dataset Statistics

R2R. The average length of instructions is 32 words. The average path length of the ground truth trajectory of each instruction is six steps. The number of train, val seen, val unseen, and test episodes are $14{,}025$, 1020, 2349, and 4173.
SOON. The average path length of the ground truth trajectory of each instruction is four to seven steps. The number of train, validation seen instruction, validation seen house, validation unseen house episodes are 3085, 245, 195, and 205.
REVERIE. The average path length of the ground truth trajectory of each instruction is 9.5 steps. The number of train, val seen, val unseen, and test episodes are $10{,}466$, 1423, 3521, and 6292.

B.2. Data Preprocessing

To calculate reference SOS features, we preprocess object tokens from language instructions. Using a pretrained visual question answering (VQA) model [3] with the question *"What is the target object? Answer in one word."*, we extract target objects from the instructions in R2R and REVERIE datasets. For SOON dataset, the target object names are already given. After extracting target objects, we perform object parsing for the instructions as shown in Table 1. The final object tokens are sorted by order of appearance in the instructions for R2R and REVERIE. For SOON, considering that the full instruction is divided into 5 parts: object name, object attribute, object relationship, target area, and neighbor areas, we sort the object tokens by reversed order of sentences.

B.3. Baselines

Seq2Seq [4] uses sequence-to-sequence action prediciton to generate actions from the agent trajectory. Speaker-Follower [5] uses the speaker model to augment natural language instructions and evaluate the candidate action sequence. FAST [6] uses both local and global signals to look forward the unobserved environment during exploration and backtrack to the originally visited nodes when needed. SMNA [7] uses visual-textual co-grounding module that encodes the past instructions and the instructions and actions to be done. SMNA also uses a progress monitor to estimate the current progress of the agent relative to the total instructions. Regretful-Agent [8] improves SMNA via two modules. The regret module decides whether to continue to explore or rollback to previous state by a learned policy, and the progress marker decides the direction the agent should head to by selecting visited nodes with progress estimates. RCM [9] applies reinforcement learning to enforce the global matching between the agent trajectory and the given natural language instruction. Via cycle-reconstruction reward, RCM allows the agent to comprehend the natural language instruction and penalize paths that do not match with the given instructions. FAST-MATTN [10] introduces a Navigator-Pointer model to both navigate to the target point and to localize the object from the navigation point according to the language guidance. AuxRN [11] introduces four auxiliary tasks that help learning the navigation policy: a trajectory retelling task, a progress estimation task, an angle prediction task, and cross-modal matching task, and improves navigation success by aligning representations in these unseen domains with seen domain. HAMT [12] uses transformer instead of a recurrent unit to predict actions from a long-range trajectory of observations and actions. Airbert [13] uses ViLBert [14] to measure the correlation between the language instructions and the viewpoint trajectories. VLN↻BERT [15] adds a recurrent unit in the transformer to predict the action from the trajectory. SIA [16] first pretrains the agent to learn the cross-modality between object grounding task and scene grounding task, and then generates real action sequences with memory-based attention. SSM [17] integrates information during exploration and constructs a scene memory and chooses the most probable node among visited nodes during backtracking. GBE [18] models the navigation state as a graph and explores the environment based on the navigation graph. DUET [19] uses two models, a local encoder and a global map planner, to fuse the local observations and coarse scale encoding for planning actions.

C. Navigation Experiments

In this section, we analyze the evaluation results of navigation experiments with different evaluation metrics. The results are provided in the paper.

C.1. Detailed Analyses in R2R

Navigation Error (NE). Navigation error (NE) is measured as the average distance between the final location of the agent and the target location of episode in meters. Because each episode is recorded as success if NE is less than $3m$, NE is strongly related with the success rate. Meta-Explore shows the lowest NE in the val seen and test unseen splits of the R2R navigation task. The results imply that hierarchical exploration with local goal search helps the agent arrive to the target location closer than other baselines.
Trajectory Length (TL). Among all the R2R navigation baselines, Seq2Seq shows the lowest TL. However, Seq2Seq shows low success rate and low SPL in all data splits. Compared to navigation baselines with SPL higher than 50% in the test split, VLN↻BERT, SMNA, and HAMT-e2e show lower TL than Meta-Explore. However, all three of these methods show a lower success rate, SPL, and NE than Meta-Explore. According to R2R [4], train episodes show a wide range of average trajectory length

Methods	Memory	Exploit	Val Seen				Val Unseen						Test Unseen			
			SR↑	SPL↑	OSR↑	TL↓	SR↑	SPL↑	FSR↑	FSPL↑	OSR↑	TL↓	SR↑	SPL↑	OSR↑	TL↓
Human	-	-	-	-	-	-	-	-	-	-	-	-	81.51	53.66	86.83	21.18
Seq2Seq [4]	Rec	✗	29.59	24.01	35.70	12.88	4.20	2.84	2.16	1.63	8.07	11.07	6.88	3.99	10.89	**3.09**
VLN↻BERT [15]	Rec	✗	51.79	47.96	53.90	13.44	30.67	24.90	18.77	15.27	35.02	16.78	29.61	23.99	32.91	15.86
RCM [9]	Rec	✗	23.33	21.82	29.44	10.70	9.29	6.97	4.89	3.89	14.23	11.98	7.84	6.67	11.68	10.60
SMNA [7]	Rec	**homing**	41.25	39.61	43.29	**7.54**	8.15	6.44	4.54	3.61	11.28	**9.07**	5.80	4.53	8.39	9.23
FAST-MATTN [10]	Rec.	✗	50.53	45.50	55.17	16.35	14.40	7.19	7.84	4.67	28.20	45.28	19.88	11.61	30.63	39.05
HAMT [12]	Seq	✗	43.29	40.19	47.65	12.79	32.95	30.20	18.92	17.28	36.84	14.08	30.40	26.67	33.41	13.62
SIA [16]	Seq.	✗	61.91	57.08	65.85	13.61	31.53	16.28	22.41	11.56	44.67	41.53	30.80	14.85	44.56	48.61
Airbert [13]	Seq.	✗	47.01	42.34	48.98	15.16	27.89	21.88	18.23	14.18	34.51	18.71	30.28	23.61	34.20	17.91
DUET [19]	Top. Map	✗	71.75	63.94	**73.86**	13.86	46.98	33.73	32.15	23.03	51.07	22.11	**52.51**	36.06	**56.91**	21.30
Meta-Explore (Ours)	Top. Map	**local goal**	71.68	63.90	73.79	13.84	47.49	34.03	**32.32**	23.30	**51.21**	22.12	-	-	-	-
Meta-Explore* (Ours)	Top. Map	**local goal**	**71.89**	**65.71**	73.44	13.03	**47.66**	**40.27**	32.15	**27.21**	50.55	18.48	51.18	**44.04**	53.8	10.23

Table 2. **Comparison and evaluation results of the baselines and our model in REVERIE Navigation Task.**
Gray shaded rows describe hierarchical navigation baselines. Three memory types: Rec (recurrent), Seq (sequential), and Top. Map (topological map)

from $5m$ to $25m$, while the test episodes have an average trajectory length of $9.93m$. This implies that the agent is trained with longer trajectories than the test split trajectories, thereby the navigation policy might have learned to navigate longer paths better than shorter paths.

C.2. Detailed Analyses in SOON

Oracle Success Rate (OSR). In the SOON navigation task, Meta-Explore achieves the highest OSR in the test split while it does not improve the OSR in the val seen instruction and val seen house splits. The proposed method shows a significant generalization result compared to the baselines. AuxRN shows the highest OSR in both the val seen instruction split and the val seen house split as 78.5% and 97.8%, respectively, but shows the OSR in the test unseen split as 11.0%. On the other hand, Meta-Explore shows OSR as 96.0%, 52.7%, and 48.7% in the val seen instruction, val seen house, and test unseen splits, respectively. Meta-Explore outperforms AuxRN on OSR by 442.7% in the test split.

Object Grounding Performance (FSPL). Following [18], we measure the object grounding performance with the target finding success weighted by path length (FSPL). Although Meta-Explore show the highest success rate and SPL in the val seen instruction and test splits, it does not improve FSPL over baseline methods. We expect to achieve better performance on FSPL if the agent uses the SOS features as deterministic clues to find the target object at the end of each episode.

C.3. Evaluation Results in REVERIE benchmark

Table 2 compares Meta-Explore with the baselines in the REVERIE navigation task. While the proposed method does not improve performance in the val seen split, Meta-Explore outperforms other baselines in the val unseen on success rate, SPL, FSR, FSPL, and OSR. However, the improvement of performance is lower than the improvements shown in R2R and SOON benchmarks. We found 252 meaningless object categories (e.g., verbs, adjectives, and prepositions) and 418 replaceable object categories (e.g., typographical errors and synonyms) in the REVERIE dataset. 10.7% and 41.2% of a total of 46,476 words in the bounding box dataset correspond to meaningless and replaceable object categories, respectively. Because our exploitation method utilizes object-based parsing of the given instruction to match with the detected object categories, the effectiveness of the proposed method is lessened due to inaccuracies and inconsistencies in the dataset. We expect to have higher performance if the mistakes in the dataset are fully fixed. To provide evidence for this hypothesis, we evaluate Meta-Explore with a modified dataset, which is partially fixed. Typographical errors are fixed and words that have similar meanings are unified into a single object category. For instance, 'blackboard', 'whiteboard', and 'bulletin' are all unified into 'board'. The results are shown as the performance of **Meta-Explore*** in Table 2. The results imply that the proposed method can effectively enhance the SPL by classifying the detected objects correctly, using the modified dataset.

Comparison between exploitation policies in the REVERIE navigation task is shown in Table 3. Among the four exploitation methods: random, spatial, spectral local goal search and homing, spectral-domain local goal search shows the highest performance. The results in Table 3 are consistent with the results in R2R and SOON.

Local Goal	Val Seen					Val Unseen				
	SR↑	SPL↑	FSR↑	OSR↑	TL↓	SR↑	SPL↑	FSR↑	OSR↑	TL↓
Oracle	79.20	64.17	62.83	84.05	18.53	59.07	38.23	40.36	66.86	26.71
Random	0.21	0.04	0.00	20.31	46.02	1.11	0.18	0.34	26.70	0.05
Homing	68.45	50.54	55.24	73.23	17.95	43.60	28.25	29.59	49.28	25.64
Spatial	67.53	40.21	54.25	70.91	26.92	40.90	23.25	27.61	45.84	26.92
Spectral	**71.68**	**63.90**	**57.34**	**73.79**	**13.84**	**47.49**	**34.03**	**32.32**	**51.21**	**22.12**

Table 3. Comparison of Exploitation Policies. (REVERIE)

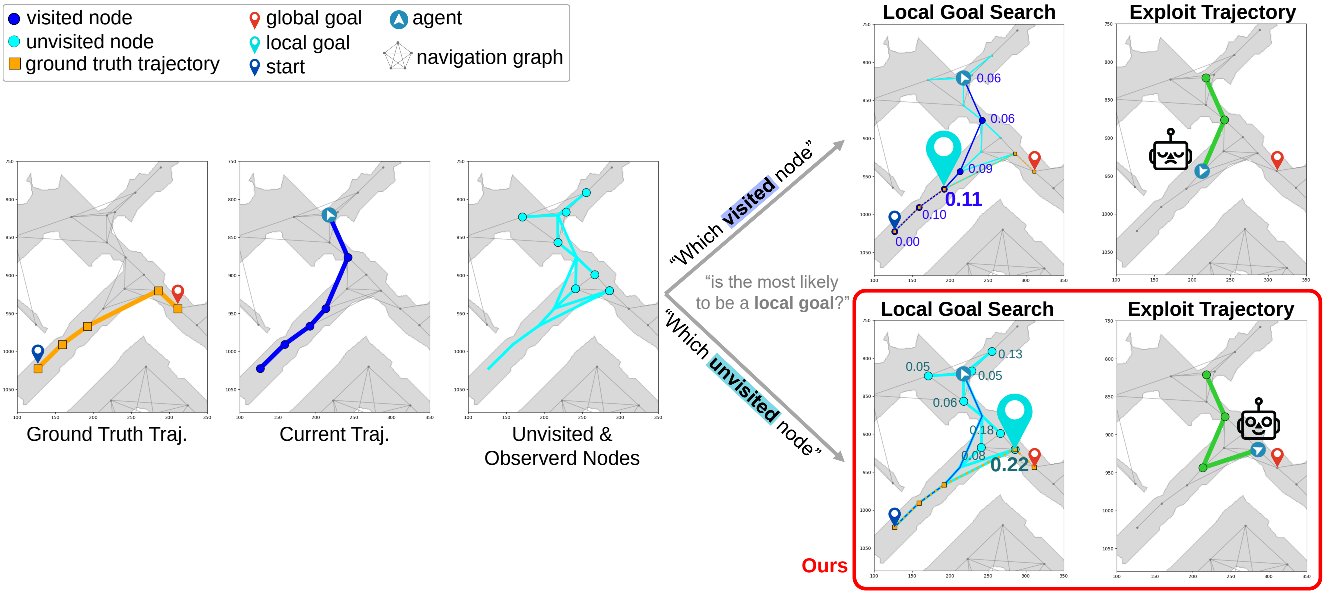

Figure 4. **Local goal search scenarios in R2R.** Ground truth trajectory (orange) and current trajectory at time $t = 6$ (blue) are shown in the left. Traj. denotes trajectory. The number next to each node denotes the navigation score S_{nav} of the shortest path trajectory from the start node to the corresponding node. If the local goal is chosen from the previously visited nodes, the local goal becomes the node with $S_{nav} = 0.11$. If the local goal is chosen from the unvisited but observed nodes, the local goal becomes the node with $S_{nav} = 0.22$.

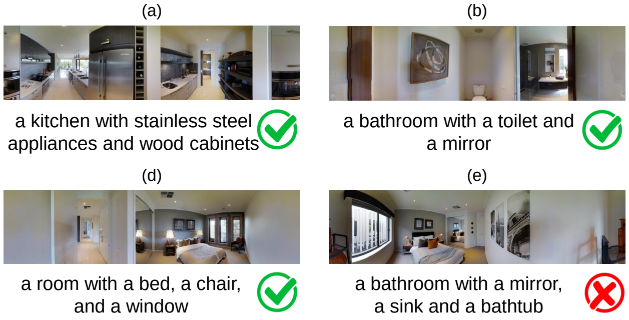
(a)
a kitchen with stainless steel appliances and wood cabinets
(b)
a bathroom with a toilet and a mirror
(d)
a room with a bed, a chair, and a window
(e)
a bathroom with a mirror, a sink and a bathtub

Figure 5. **Sample image captions.** (a), (b), (c), and (d) show captions that successfully describe the scenes. (e) and (f) shows failure cases of caption generation. For successful language-triggered hierchical exploration, image captions should correctly describe the scenes. However, current image captioning methods often generates misdescribed captions, thereby leading to a low navigation performance.

C.4. Local Goal Search

In this section, we provide sample local goal search scenarios. Figure 4 shows two scenarios of choosing the local goal when the agent moves to a wrong direction. The agent is given an instruction *"Turn right and turn right again after the desk on the right. Wait next to the cabinets and microwave."*. In both scenarios, we assume that the agent chooses the local goal as the node with the highest navigation score among the possible candidates. If the local goal is chosen from the previously visited nodes, the agent has to move back toward the explored regions. In contrast, if the local goal is chosen from unvisited but observed nodes, the agent can choose a local goal which is close to the global goal. The two scenarios imply that the local goal search in Meta-Explore is more effective than exploitation methods that return the agent to a previously visited node.

D. Ablation Study

D.1. Language-triggered Hierarchical Exploration

In the proposed method, the target instruction and local goal candidates are compared in spectral-domain using SOS features. Since semantic information can also be expressed in language-domain, we further experiment with the local goal search method using synthesized language captions from visual observations in the R2R navigation task. We compare three types of representation domains: spatial, spectral, and language, which are implemented as panoramic RGB image embeddings, SOS features, and sentence embeddings, respectively. To compare features in different domains, we transfer the source domain to another using augmentation or cross-domain similarity.

D.1.1 Implementation Details

We address that the agent can use image captioning to extract contextual information from visual observations such as room type, color, and object placements. To compare local goal candidates and target instruction in language do-

Domains		Val Seen		Val Unseen	
Nav. Target	Local Goal	SR	SPL	SR	SPL
Lang.	✗	79.92	72.79	70.63	59.81
Lang.	Spatial	78.84	71.96	71.05	58.86
Lang.	Lang. Aug.	77.96	70.77	69.52	57.26
Spectral Aug.	Spectral	**80.61**	**75.15**	**71.78**	**61.68**

Table 4. Comparison and evaluation results of the local goal search methods using different target and candidate domains. (R2R)

main, we use pretrained ViT [1] and GPT-2 [20] to generate the caption for each viewpoint as Figure 5. The Figure shows four successful cases and two failure cases of image captions. To find a local goal using the generated captions, we calculated the similarities between the captions corresponding to local goal candidates and the target instruction using a fine-tuned sentence transformer 'all-MiniLM-L6-v2' [21]. The local goal is chosen as the candidate with the highest similarity. Additionally, we use pretrained CLIP [22] to evaluate local goal search based on cross-modal similarities between the visual observations of local goal candidates and the target instruction.

D.1.2 Experiment Results

Table 4 shows the evaluation results of the local goal search methods using different target and candidate domains in R2R navigation task. *Nav. Target* denotes the target of VLN, initially given as language. *Lang. Aug.* denotes language captions generated from images. *Spectral Aug.* denotes reference SOS features generated from language instructions. Among the three representation domains, the spectral-domain features enhance navigation performance the most. This implies that hierarchical exploration is most effective when used with spectral visual features. Table 1 and Table 2 in the paper also show the improvement of navigation performance by using both hierarchical exploration and spectral visual features over DUET [23], which uses the same ViT-B/16 to extract spatial visual features, resulting in 17.1% increase in SR and 20.6% increase in SPL in the SOON test unseen split.

D.2. Image-Goal Navigation in Continuous Domain

To implicate further applications of Meta-Explore in a continuous domain, we evaluate our method on the photo-realistic Habitat [24] simulator with continuous action space with realistic noises to solve an image-goal navigation task. The objective is to arrive at the target location of the given goal image in an unseen environment. We mainly focus on the effectiveness of hierarchical exploration using local goal search in this experiment. The results are shown in Table 5.

D.2.1 Exploration-Exploitation Selection

We extend Meta-Explore to continuous environments to address the impact of hierarchical exploration in realistic environments. The mode selector decides when to explore and exploit. In the exploration mode, the agent explores around a local area until the meta-controller decides to stop the exploration. The exploration module consists of graph construction module and navigation module. We use recurrent action policy that takes the current and target image features and outputs low-level actions for exploration. We illustrate that the explore-exploit switching decision occurs in stuck scenarios, such as entering a small place or getting stranded in a corner. Figure 6 shows the overview of exploitation in image-goal navigation by searching a local goal. When the control mode is changed to exploitation mode, the agent returns to the closest previously visited node. Then, the agent finds a local goal among the nodes in the constructed topological map and moves toward the local goal using dijkstra's algorithm [25]. The local goal is chosen as the node which has the most similar SOS feature with the SOS feature of the target image based on cosine similarity. The agent repeats this explore-exploit behavior until it finds the goal. This explore-exploit switching decision increases the navigation success rate.

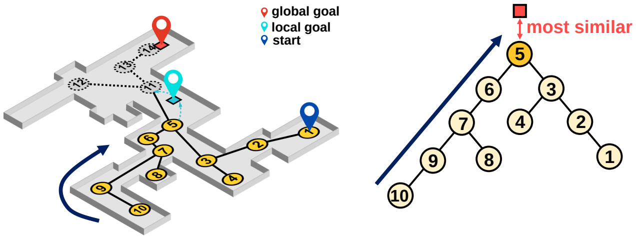

Figure 6. **Exploitation by searching a local goal.** In the exploit mode, the agent aims to escape from the stranded local area. It first searches for the most similar node to the goal. Then, it finds an optimal local goal which is unexplored and also similar to the goal image. We use SOS features to compare with the target image.

D.2.2 Experiment Details

We evaluate Meta-Explore in the Gibson dataset [30] with Habitat [24] simulator to solve an image-goal navigation task. Habitat simulator allows the agent to navigate in photo-realistic indoor environments. The exploration policy of the agent is trained using 72 scenes. We evaluate Meta-Explore using 14 unseen scenes. We use panoramic RGBD observations and construct image-based graph memory. To construct a context frequency vector, we detect objects via Mask2Former [31] pretrained in ADE-20K dataset [32], to effectively detect the objects that are generally located in indoor scenes. We use a discrete action space, {`stop`, `move forward`, `turn left`, `turn right`} for navigation. With `move forward` action, an agent moves forward by 0.25 m, while `turn left` and `turn right` denotes a 10° rotation, counter-clockwise and clockwise,

Methods	Exploit	Need	Domain		Easy		Medium		Hard		Overall	
		Pose Info.	spatial	frequency	SR	SPL	SR	SPL	SR	SPL	SR	SPL
VGM [26]	✗	**no**	RGBD	✗	0.86	0.80	0.81	**0.68**	0.61	0.46	0.76	**0.64**
Neural Planner [27]	✔	global	RGBD	✗	0.72	0.41	0.65	0.39	0.42	0.27	0.60	0.36
NTS [28]	✔	global	RGBD	✗	0.87	0.65	0.58	0.38	0.43	0.26	0.63	0.43
ANS [29]	✔	global	RGBD	✗	0.74	0.21	0.68	0.23	0.30	0.11	0.58	0.18
Meta-Explore (homing)	✔	**local**	RGBD	SOS	0.82	0.61	0.83	0.61	0.70	**0.48**	0.78	0.57
Meta-Explore (localgoal)	✔	**no**	RGBD	SOS	**0.94**	**0.84**	**0.88**	0.63	**0.71**	0.18	**0.84**	0.55

Table 5. **Evaluation results for Image-goal Navigation Task.**
(SR: success rate, SPL: success weighted by path length)

respectively. The difficulty of each episode is determined by the geodesic distance between the initial and the goal location; *easy*: 1.5 m∼3 m, *medium*: 3 m∼5 m, and *hard*: 5 m∼10 m. The actuation noise model [29] is also applied to the agent in order to evaluate in realistic situations. We also demonstrated navigation experiments in the real world using a Jackal robot. The episodes are sampled from simulation point goal episodes with all difficulties; *easy*, *medium* and *hard*. We demonstrate both straight and curved trajectories to evaluate that our model is not task-specific. We used the model only trained in Habitat simulator with Gibson dataset. To collect panoramic RGBD observations, we use one panoramic RGB camera and four front-view RGBD cameras. In order to implement collision avoidance similar to the construction of navigable mesh in Habitat simulator, we implemented a collision avoidance module by clipping the action value based on the depth image observation.

D.2.3 Baselines

We compare our image-goal navigation policy with various baselines. Active Neural SLAM (ANS) constructs a top-down metric map and uses a hierarchical structure consisting of global and local policies. The global policy outputs long-term goals, which are used to generate short-term goals. The local policy uses a geometric path planner to navigate to a short-term goal. NTS [28] constructs a topological graph during exploration and plans subgoals with graph localization and planning, while navigating to the node with local point goal navigation policy. Neural Planner [27] constructs a graph using an estimated connectivity probability calculated from the neural network. VGM [26] uses unsupervised image-based graph memory representation to compare the similarity between goal image and the current observation image. We adapt VGM for graph construction and local navigation policy. PCL [33] encoder with ResNet18 [34] backbone network is used as the visual encoder for VGM [26].

D.2.4 Evaluation Metrics

We evaluate both success rate (SR) and success weighted by inverse path length (SPL) [35]. An episode is recorded as success if the agent takes a `stop` action within 1 m of the target location. SR is denoted as the number of successes divided by the total number of episodes, E. SPL is calculated as $\frac{1}{E}\sum_{i=1}^{E} S_i \frac{l_i}{\max(p_i, l_i)}$. S_i denotes the success as a binary value. p_i and l_i denote the shortest path and actual path length for the i^{th} episode. For each task difficulty, SR and SPL are measured separately.

D.2.5 Experiment Results

Detailed comparisons with the baseline methods are shown in Table 5. The results show that the continuous version Meta-Explore and SOS features help navigation and the exploitation mode provides corrections for misled exploration or undesirable actions. Compared with the exploration policy baseline VGM [26], Meta-Explore shows an enhancement in the overall success rate by 10.5%. The results imply that local goal search helps the agent escape from the current location when the agent recurrently explores a local area but cannot find the target location. Exploitation can reduce unnecessary exploration and help the agent reach the target goal before the maximum time horizon. Among two methods of exploitation, local goal search outperforms homing, presumably because of the noisy actuation model used in the simulator. Due to the noisy actions, the agent can hardly return to a previously visited location by directly reversing the action sequence.

Comparing our method with other graph-based hierarchical navigation methods, Meta-Explore outperforms ANS, Neural Planner, and NTS in the success rate. Our model shows lower performance in SPL for *hard* episodes while the success rate is higher than the baselines. This implies that the exploitation mode of the proposed method allows the agent to explore more uncovered areas. Meanwhile, the proposed method appears to yield a positive impact for *easy* episodes, with the increase on both success rate by 9.3% and SPL by 5.0%. Specifically, our method outperforms ANS in terms of both success rate and SPL across all episodes. When compared to Neural Planner and NTS, our approach shows better performance in both success rate and SPL for *easy* and *medium* episodes, while outperforming Neural Planner and NTS in success rate for *hard* episodes. On the other hand, the proposed method shows

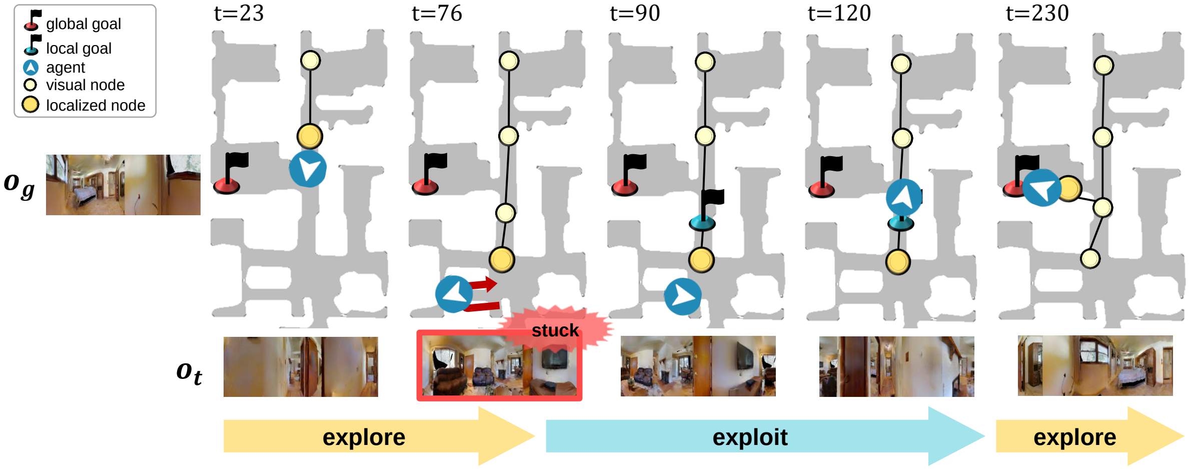

Figure 7. **Experiment visualization for image-goal navigation task in continuous environment.** The mode selector detects stuck event at t = 76 and switches the explore mode to exploit mode. Then, the agent returns toward the local goal, which is chosen as a position nearby one of the nodes in the previously constructed graph.

lower SPL for *hard* episodes than NTS and Neural Planner. This implies that Meta-Explore tends to explore uncovered areas in both successful and unsuccessful episodes, which could be the result of using the SOS features to understand scenes. Comparing the proposed method using different exploitation methods (homing and local goal search) shows that searching for a local goal leads the agent to better escape from a local area. Figure 7 shows a simple scenario of image-goal navigation using Meta-Explore. The mode selector detects a regretful situation when the agent is recurrently exploring a local area but cannot find the target location. Hierarchical exploration via local goal search helps the agent overcome the situation and move toward the global goal in fixed time.

D.3. VLN in Continuous Domain

Image-goal navigation results in complex settings (continuous environments with noisy actions, max∼300 steps) imply that our model can be transferred to long-horizon VLN with noisy actions. We further extend the proposed method in continuous environments to solve the VLN-CE [36] task. In the VLN-CE [36] task, our agent constructs a topological map by using Conti-CMA [37] as a baseline to find reachable nodes (i.e., waypoints) and reuses the map in the exploitation mode. We compare our continuous version Meta-Explore with various navigation baselines[1]: VLN-CE [36], HCM [38], SASRA [39], and Conti-CMA [37]. We evaluate algorithms using the success rate (SR), success weighted by inverse path length (SPL), oracle success rate (OSR), trajectory length (TL), and navigation error (NE), following the definitions of the evaluation metrics in the paper.

Methods	Memory	Exploit	SR↑	SPL↑	OSR↑	TL↓	NE↓
VLN-CE [36]	Rec	✗	32	30	40	8.64	7.37
HCM† [38]	Rec	✗	-	-	43	15.61	8.93
SASRA [39]	Semantic Map	✗	24	22	-	**7.89**	8.32
Conti-CMA† [37]	Top. Map	✗	41	35	51	10.90	6.20
Meta-Explore (Ours)	Top. Map	local goal	**49**	**38**	**54**	14.88	**4.25**

Table 6. Evaluation results in the VLN-CE val unseen split.

D.3.1 Experiment Results

Results in Table 6 show that our method outperforms other baselines by at least 19.5% in the success rate, 8.6% in SPL, and 5.9% in OSR. We excluded the results of HCM for SR and SPL because HCM measures SR, SPL using oracle stop in the official code, which is not allowed in other baselines. We address that our model can be transferred to long-horizon (max. step 300) VLN with noisy actions in complex settings, as demonstrated by image-goal navigation results in Sec. D.2.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*, 2021. 2, 6

[2] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, pages 5100–5111, 2019. 2

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE*

[1] † indicates reproduced results.

International Conference on Computer Vision, pages 2425–2433, 2015. 3

[4] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018. 3, 4

[5] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2018. 3

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

[7] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *Proceedings of the International Conference on Learning Representations*, 2019. 3, 4

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

[9] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019. 3, 4

[10] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991, 2020. 3, 4

[11] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10012–10022, 2020. 3

[12] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 5834–5847, 2021. 3, 4

[13] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1634–1643, 2021. 3, 4

[14] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2019. 3

[15] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln↻bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1643–1653, 2021. 3, 4

[16] Xiangru Lin, Guanbin Li, and Yizhou Yu. Scene-intuitive agent for remote embodied visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7036–7045, June 2021. 3, 4

[17] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8455–8464, June 2021. 3

[18] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12689–12699, 2021. 3, 4

[19] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16537–16547, 2022. 3, 4

[20] 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. 6

[21] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. 6

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763, 2021. 6

[23] S. Chen et al. 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*, 2022. 6

[24] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam,

Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2021. 6

[25] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959. 6

[26] Obin Kwon, Nuri Kim, Yunho Choi, Hwiyeon Yoo, Jeongho Park, and Songhwai Oh. Visual graph memory with unsupervised representation for visual navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15890–15899, 2021. 7

[27] Edward Beeching, Jilles Dibangoye, Olivier Simonin, and Christian Wolf. Learning to plan with uncertain topological maps. In *Proceedings of the European Conference on Computer Vision*, pages 473–490. Springer, 2020. 7

[28] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12875–12884, 2020. 7

[29] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *Proceedings of the International Conference on Learning Representations*, 2020. 7

[30] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018. 6

[31] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 6

[32] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017. 6

[33] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *Proceedings of the International Conference on Learning Representations*, 2021. 7

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 7

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

[36] Jacob Krantz, Erik Wijmans, Arjun Majundar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision and language navigation in continuous environments. In *Proceedings of the European Conference on Computer Vision*, 2020. 8

[37] Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 8

[38] Muhammad Zubair Irshad, Chih-Yao Ma, and Zsolt Kira. Hierarchical cross-modal agent for robotics vision-and-language navigation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13238–13246. IEEE, 2021. 8

[39] Muhammad Zubair Irshad, Niluthpol Chowdhury Mithun, Zachary Seymour, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Semantically-aware spatio-temporal reasoning agent for vision-and-language navigation in continuous environments. *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2021. 8