Bridging the Gap Between Learning in Discrete and Continuous Environments for Vision-and-Language Navigation

Yicong Hong[1]* Zun Wang[1]* Qi Wu[2] Stephen Gould[1]
[1]The Australian National University, [2]University of Adelaide
{yicong.hong, zun.wang, stephen.gould}@anu.edu.au
qi.wu01@adelaide.edu.au

Project URL: https://github.com/YicongHong/Discrete-Continuous-VLN

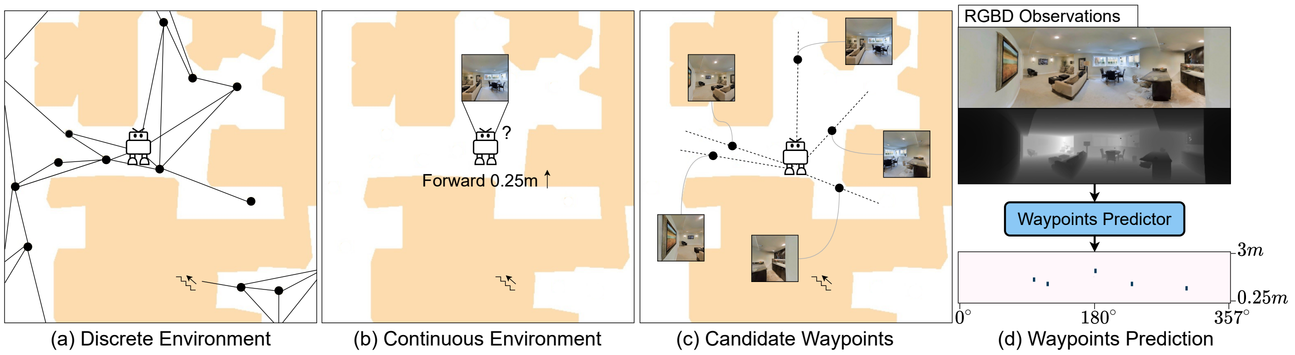

Figure 1. Vision-and-language navigation in discrete versus in continuous environments. (a) Agents in discrete environments rely on the connectivity graph to navigate with panoramic high-level actions, (b) but they need to perform low-level controls to move in continuous spaces. (c,d) We propose a candidate waypoints predictor to predict accessible positions in continuous environments, to bridge the discrete-to-continuous gap.

Abstract

Most existing works in vision-and-language navigation (VLN) focus on either discrete or continuous environments, training agents that cannot generalize across the two. Although learning to navigate in continuous spaces is closer to the real-world, training such an agent is significantly more difficult than training an agent in discrete spaces. However, recent advances in discrete VLN are challenging to translate to continuous VLN due to the domain gap. The fundamental difference between the two setups is that discrete navigation assumes prior knowledge of the connectivity graph of the environment, so that the agent can effectively transfer the problem of navigation with low-level controls to jumping from node to node with high-level actions by grounding to an image of a navigable direction.

To bridge the discrete-to-continuous gap, we propose a predictor to generate a set of candidate waypoints during navigation, so that agents designed with high-level actions can be transferred to and trained in continuous environments. We refine the connectivity graph of Matterport3D to fit the continuous Habitat-Matterport3D, and train the waypoints predictor with the refined graphs to produce accessible waypoints at each time step. Moreover, we demonstrate that the predicted waypoints can be augmented during training to diversify the views and paths, and therefore enhance agent's generalization ability.

Through extensive experiments we show that agents navigating in continuous environments with predicted waypoints perform significantly better than agents using low-level actions, which reduces the absolute discrete-to-continuous gap by 11.76% Success Weighted by Path Length (SPL) for the Cross-Modal Matching Agent and 18.24% SPL for the VLN↻BERT. Our agents, trained with a simple imitation learning objective, outperform previous methods by a large margin, achieving new state-of-the-art results on the testing environments of the R2R-CE and the RxR-CE datasets.

* Authors contributed equally

1. Introduction

Vision-and-language navigation (VLN) [3] is a challenging cross-domain research problem which requires an agent to interpret human instructions and navigate in previously unseen environments by executing a sequence of actions. Two distinct scenarios have been proposed for VLN research, being navigation in discrete environments (R2R, RxR [3, 33]) and in continuous environments (R2R-CE, RxR-CE [32]). Due to the large domain gap, navigation in the two scenarios have been studied independently in previous works, and a large number of prominent advances achieved by agents in discrete spaces, such as by leveraging pre-trained visiolinguistic transformers [20, 24] or by using scene memory [16, 53] cannot be directly applied to agents traversing continuous spaces.

The fundamental difference between navigation in discrete and in continuous environments is the reliance on the connectivity graph which contains numbers of sparse nodes (waypoints) distributed across accessible spaces of the environment. With the prior knowledge of the connectivity graph, the agent can move with a panoramic high-level action space [17], *i.e.*, teleport to an adjacent waypoint on the graph by selecting a single direction from the discrete set of navigable directions. Compared to navigation in continuous environments, which usually relies on a limited field of view to infer low-level controls (*e.g.*, turn left 15 degrees or move forward 0.25 meters) [32], navigation with panoramic actions and the connectivity graph simplifies the complicated decision making problem by formulating it as an explicit text-to-image grounding task. First, the agents are not required to infer the important concepts of accessibility (openspace vs. obstacles) from sensory inputs. Second, distinct visual representations can be defined for each navigable direction, so the agents only need to match contextual clues from instruction to visual options to move, which greatly reduces the agent's state space and facilitates the learning. As a result, many previous works in VLN with high-level actions address the navigation problem mostly from the visual-textual matching perspective. A large number of innovations such as back-translation [51], back-tracking [29, 37], scene memory [16, 53] and transformer-based pre-training [20, 24, 38] bring remarkable improvement but they cannot be directly transferred to agents in continuous environments. There remains about a 20% gap in success rate for agents with the same architecture navigating in discrete and in continuous spaces [32].

Despite the great efficiency of learning in a discrete environment, navigation in continuous spaces is much closer to the real-world. In this paper, we address the problem of bridging the learning between the two domains, aiming to effectively adapt agents designed for discrete VLN to continuous environments. First of all, we identify and quantitatively evaluate the value of high-level controls in VLN, showing the importance of knowing accessible waypoints. Second, inspired by Sim2Real-VLN [2], we introduce a powerful candidate waypoints predictor to estimate navigable locations in continuous spaces, the module constructs a local navigability graph centered at the agent at each time step using the visual observations. In Sim2Real-VLN [2], the sub-goal module is trained on the pre-defined connectivity graphs of the Matterport3D environment (MP3D) [6], where there exist edges going through obstacles and nodes in inaccessible spaces. In contrast, we transfer the discrete MP3D graph onto the continuous Habitat-MP3D [48] spaces, and represent the transferred waypoints as targets for learning a mixture of Gaussian probability map, resulting in a robust waypoints predictor in unvisited environments that supports navigation. Furthermore, we propose a simple augmentation method to move the position of waypoints during training of the agent, so that the agent can learn to reach the same target using diverse observations and step lengths, thereby improving the generalization ability.

With the proposed candidate waypoints predictor, we evaluate the performance of agents designed for discrete VLN in continuous environments. The selected agents, including the cross-modal matching agent (CMA) [54] and the VLN↻BERT [24] are widely applied methods that are distinct in network architecture. Our experiments show that agents in continuous environments trained with the predicted waypoints significantly improves over navigation without using waypoints, reducing the discrete-continuous gap, and achieving new state-of-the-art performance of 39% and 19.61% SPL on the benchmarking R2R-CE and RxR-CE test sets [3,32,33], respectively. This results suggest that our proposed candidate waypoints predictor can enable an effective discrete-to-continuous transfer as well as showing a huge potential of benefiting other navigation problems.

2. Related Work

Vision-and-Language Navigation Visiolinguistic cross-modal grounding is a crucial skill for addressing vision-and-language navigation problems. Tasks for indoor navigation with low-level instructions such as R2R [3] and RxR [33], outdoor navigation such as Touchdown [11], navigation with dialog such as CVDN [52] and HANNA [40], and navigation for remote object grounding such as REVERIE [43] and SOON [59] all require the agent's ability to associate time-dependent visual observations to instructions for decision making. To facilitate the learning, Fried *et al.* [17] exploit the connectivity graphs defined for Matterport3D environments [6] and propose to navigate with panoramic actions, which allows teleportation of agent among adjacent nodes (waypoints) on the graph by choosing an image pointing towards the node. Following this idea, later works that are focusing on cross-modality learning [23, 36, 54], data augmentation [18,41,51], waypoint-tracking [16,29,37,53], and pre-training for Transformer-based models [20, 24, 34, 38] are implemented based on the connectivity graph and the high-level action space. Despite the great advances in learning the correspondence among vision, language and action, these agents are not applicable to the more practical scenario – navigation in continuous environments, which requires the agent's ability of inferring spatial accessibility.

Continuous VLN Continuous environments such as AI2-THOR [30], House3D [56], CHALET [58], Gibson [57], Habitat [48] and iGibson [49] have been setup for embodied AI research in synthetic and photo-realistic scenes. To study vision-and-language navigation in continuous environments (VLN-CE), Krantz *et al.* [32] transfer the discrete paths in R2R [3] (and RxR [33]) dataset to continuous trajectories based on the Habitat simulator [48], and Irshad *et al.* propose a hierarchical model for inferring agent's linear and angular velocities in the Robo-VLN environment [27]. In addition, methods such as applying semantic map representations [28] and using language-aligned waypoints supervision [44] are explored in VLN-CE. Experiments demonstrate a huge performance gap between agents that navigate in discrete and continuous environments.

Hierarchical Visual Navigation Hierarchical visual navigation, involving the problem of mapping, planning and control, have been extensively studied in previous literatures [2, 7, 8, 10, 12–14, 19, 31, 39, 47]. Active Neural SLAM [7] plans towards a long-term goal with a map and agent pose estimated from visual and sensory inputs. Chen *et al.* [12] construct topological maps for planning by sparsifying continuous paths collected in the environments, and Chen *et al.* [10] predicts a single audio-conditioned subgoal at each step while our model estimates key positions around the agent. Recent work that is the most relevant to ours are the Waypoint Models [31] and the Sim2Real-VLN [2]. Waypoint Models predicts a coarse view (direction) and a sub-goal in that view to explore, where all predictions are conditioned on vision, language and agent's state. Whereas our method decouples the waypoints prediction and agent's decision making process, and leverages the learned navigability to provide accessible directions for the agent to act, without extra efforts in modifying network architectures or training methods for adapting the new waypoint predictor. Sim2Real-VLN also predicts adjacent subgoals but the model is trained on the connectivity graphs defined in MP3D [6], where there exist a large number of invalid edges going across obstacles. In contrast, we construct graph nodes over the Habitat-MP3D spaces, and we further study the idea of training agents directly in the continuous environments with high-level actions.

3. Background

In this section, we will first introduce the background of VLN in both discrete [3] and continuous environments [32]. We apply the reinforced cross-modal matching agent (CMA) proposed by Wang *et al.* [54] to explain the general formulation of a VLN network. Then, we will evaluate the value of high-level actions by some contrastive experiments as a proof of concept for this paper.

3.1. Navigation Setups

The task of vision-and-language navigation is formulated as follows: Given a natural language instruction $\boldsymbol{U}$ as a sequence of l words $\langle w_1, w_2, \ldots, w_l \rangle$, an agent initialized at a certain position $\boldsymbol{p}_1$ is asked to travel to a target position $\boldsymbol{p}_T$ in an environment $\mathcal{E}$ by executing a sequence of actions $\boldsymbol{a}$. The overall navigation can be considered as a Partially Observable Markov Decision Process (PMODP) $\langle \boldsymbol{p}_1, \boldsymbol{a}_1, \boldsymbol{p}_2, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{t-1}, \boldsymbol{p}_t \rangle$, where each action $\boldsymbol{a}_t$ takes the agent to a new position $\boldsymbol{p}_{t+1}$ and the agent receives a new visual observation $\boldsymbol{V}_t$.[2]

Navigate with High-Level Actions Navigation with high-level actions [3, 17] is based on the connectivity graph $\mathcal{G}$ specified for each $\mathcal{E}$. Each connectivity graph contains a set of j nodes (waypoints) distributed across the entire environment $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_j\} \triangleq \mathcal{G}$. Essentially, $\mathcal{G}$ discretizes $\mathcal{E}$, constraining the agent's position $\boldsymbol{p}_t \in \mathcal{G}$ at all time. The connectivity graph also provides the navigable directions, which greatly facilitates the learning. Following the panoramic action space proposed by Fried *et al.* [17], agent at each $\boldsymbol{p}_t$ receives a panorama which is composed of n single-view inputs (RGB images) each pointing towards an adjacent waypoint, respectively. The set of visual features corresponding to those directions are represented as $\{\boldsymbol{v}^p_1, \boldsymbol{v}^p_2, \ldots, \boldsymbol{v}^p_k\}$. Each feature $\boldsymbol{v}^p_i$ is enhanced with a directional encoding $\boldsymbol{d}^p_i$ to indicate the relative orientation of the waypoint with respect to the agent's heading, denoted as $\boldsymbol{f}^p_i = [\boldsymbol{v}^p_i; \boldsymbol{d}^p_i]$, which has been shown influential in later works [23, 25]. At each navigation step, the agent's state $\boldsymbol{h}_t$ which keep tracks of all the past observations and decisions, will be updated by a recurrent network (LSTM [22]) using the attended instructions $\boldsymbol{c}^{lang}_t$ and past decision $\boldsymbol{a}_{t-1}$ as

$$\boldsymbol{h}_t = \text{LSTM}\left(\left[\boldsymbol{c}^{lang}_t; \boldsymbol{a}_{t-1}\right], h_{t-1}\right) \tag{1}$$

Based on the formulation above, decision making in discrete environments is implemented as a feature matching problem. In a compact form, the action probability for selecting an adjacent waypoint can be expressed as

$$p_i = \text{Softmax}\left(\left[\boldsymbol{h}_t; \boldsymbol{c}^{lang}_t; \boldsymbol{c}^{rgb}_t\right]^\top \boldsymbol{W}_p \boldsymbol{f}^p_i\right) \tag{2}$$

[2]POMDPs also provide the agent with an intermediate reward at each time step, which we consider to be zero in this work.

where $\boldsymbol{c}_t^{rgb}$ represents the attended images and $\boldsymbol{W}_p$ indicates learnable projections. As a result, the agent navigates with two effective mechanisms: (1) *view selection*, the agent chooses a direction with a visual observation that has the highest correspondence to the agent's state. (2) *waypoint teleportation*, the agent teleports directly to a neighbor waypoint that is positioned in the selected view.

Navigate with Low-Level Actions VLN in continuous environments (VLN-CE) [32] is established over the Habitat simulator [48], where the agent's position $\boldsymbol{p}_t$ can be any point in the open space. Due to the absence of the connectivity graph, agents need to learn to identify accessible positions in space and avoid obstacles. As proposed in VLN-CE, the agent needs to infer low-level actions (turn left 15 degrees, turn right 15 degrees, move forward 0.25 meters or stop) from the egocentric observation. To encode a single front-view image with directional clues, spatial embeddings have been concatenated to each $j = 1, ..., 16$ patch of the ResNet [21] output features before attentive pooling [32], *i.e.* $\boldsymbol{f}_j^{front} = [\boldsymbol{v}_j^{front}; \boldsymbol{d}_j]$ and $\boldsymbol{c}_t^{img} = \text{AttnPool}_{j=1,...,16}(\boldsymbol{f}_j^{front})$. The agent's state is updated also using Eq.1 but the action probability is computed as

$$\boldsymbol{p} = \text{Softmax}\left(\left[\boldsymbol{h}_t; \boldsymbol{c}_t^{lang}; \boldsymbol{c}_t^{rgb}; \boldsymbol{c}_t^{d}\right] \boldsymbol{W}_c\right) \tag{3}$$

where $\boldsymbol{c}_t^d$ represents the attended depth features and $\boldsymbol{W}_c$ projects the concatenated encodings to $\mathbb{R}^4$, corresponding to the four pre-defined actions.

With the above formulation, the learning of language-visual and language-directional correspondences becomes much more implicit. The navigation episodes with low-level actions is about 10 times longer than high-level actions, which leads to a very expensive training process.

Evaluation Metrics There are five standard metrics in VLN for evaluating the agent's performance, including Trajectory Length (TL), Navigation Error (NE), Success Rate (SR), normalized inverse of the Path Length (SPL) [1], normalized Dynamic Time Warping (nDTW) and Success weighted by normalized Dynamic Time Warping (SDTW) [26]. See Appendix for more details.

3.2. What is the Value of High-Level Actions?

We argue that *view selection* and *waypoint teleportation* are the two critical advantages resulting from high-level actions on graphs; As shown in Figure 2, view selection transfers the problem of inferring low-level controls to selecting a navigable direction, which greatly expands the agent's decision space at each time step. Moreover, by teleporting to a distant waypoint, the agent is able to learn and navigate very efficiently. The connectivity graphs provide not only information about the spatial navigability, but also an exact distance to travel which transfers the agent to a location that is suitable for future decision making (see Figure 3). Fried *et al.* [17] demonstrate a huge improvement by leveraging high-level actions in MP3D, in this section, we validate the value of these two mechanisms with the RCM agent [54] in Habitat-MP3D as a proof of concept.

We first consider four different scenarios as shown in Table 1: (1) The agent is trained and evaluated on the ground-truth connectivity graph with both view selection and waypoint teleportation. (2) Experiment that only allows view selection; At each step, the agent's local connectivity is computed on the fly using the ground-truth graph, it can choose a navigable direction but it can only move forward 0.25 meters. (3) As in the original VLN setup, navigable directions are not explicitly provided to the agent, but the agent is allowed to teleport forward if its heading is aligning with an edge on the connectivity graph [3]. We consider this setup as an experiment that allows waypoint teleportation but not view selection. (4) The original VLN-CE setup; The agent only perform low-level actions during navigation. For fairness, we apply panoramic visual input (12 cameras at 30 degrees separation) and panoramic attention to all experi-

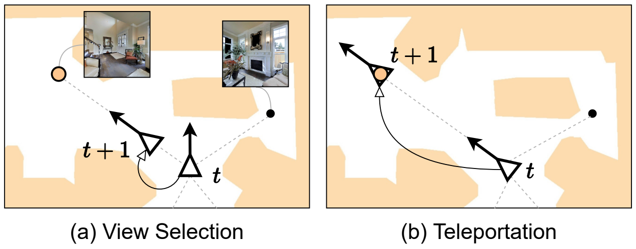

Figure 2. Navigation by (a) view selection and (b) waypoint teleportation in continuous environments.

#	Action		R2R-CE Val-Seen				R2R-CE Val-Unseen				
	Select	Teleport	NE↓	nDTW↑	SR↑	SPL↑	NE↓	nDTW↑	SR↑	SPL↑	Time
1	✓	✓	6.28	58.37	41.48	36.58	6.51	55.41	39.32	33.89	0.55
2	✓		5.70	61.06	46.31	43.20	6.38	56.27	37.72	35.02	1.79
3		✓	8.10	43.19	23.36	19.85	7.62	47.69	28.32	24.64	0.78
4			7.20	50.67	29.53	27.29	7.54	49.19	27.29	24.97	2.61

Table 1. Comparison of navigation in continuous environments by view selection (Select) and waypoint teleportation (Teleport) with the support of connectivity graphs.

#	Select	Dist	R2R-CE Val-Seen				R2R-CE Val-Unseen				
			NE↓	nDTW↑	SR↑	SPL↑	NE↓	nDTW↑	SR↑	SPL↑	Time
1		0.25	7.20	50.67	29.53	27.29	7.54	49.19	27.29	24.97	2.61
2		1.00	7.21	52.52	29.66	27.19	7.51	50.14	25.47	23.51	1.28
3		2.00	7.60	47.90	24.16	22.25	8.06	45.74	22.96	20.59	1.18
4		3.00	7.66	48.28	23.62	21.73	7.87	46.29	21.37	19.39	0.90
5	✓	0.25	6.10	58.24	39.33	38.26	6.52	55.26	32.02	31.15	1.58
6	✓	1.00	6.88	53.09	36.11	33.91	6.85	52.44	34.81	32.51	0.46
7	✓	2.00	6.56	52.83	38.52	35.62	6.96	50.25	33.05	30.35	0.27
8	✓	3.00	6.75	51.85	32.21	29.28	7.00	49.35	31.00	28.12	0.20

Table 2. Navigation results using different forward distances without the connectivity graph. *Dist* means the fixed forwarding distance (meters) in selected direction. *Time* is the averaged inference time (seconds) per episode.

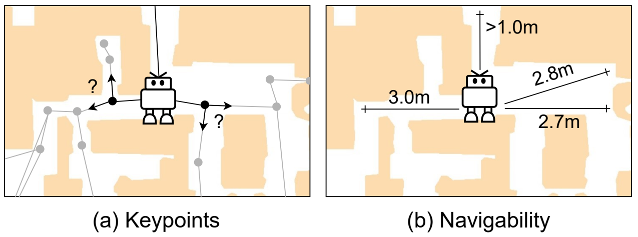

Figure 3. Keypoints versus navigability. Keypoints are closely related to the structure of the environment, which requires an agent to make a critical decision for navigation, while navigability only reflects explorable directions (and distances).

ments (See Appendix for more details). Comparing M#1[3] to M#3 and M#2 to M#4, we can see that view selection greatly improves the agent's performance. By knowing the navigable directions (M#2), agent can achieve SR close to navigation on graphs (M#1). Previous work suggests the difficulty in learning repetitive low-level actions that only return small change in view [15], our results further show that forwarding by teleportation greatly reduces execution time and slightly increases SR (M#1, M#3), while navigability is the bottleneck of learning to navigate.

Then, we investigate the influence of view selection and step size on agent's performance without considering the connectivity graph (see Table 2). In this case, an agent navigates by choosing a view and forwarding a fixed distance. Results show that *view selection* significantly boosts the agent's performance at all choices of step length (M#5-M#8). By adopting a larger step length, the agent also navigates much more efficiently due to less decisions to make. Note that, although an appropriate forward distance (*e.g.* M#6: 1 meter) leads to high performance, such value is unknown beforehand and it is unlikely to be suitable for all spatial structures. Based on the experiments, we expect to allow VLN agents in continuous spaces to take advantages of the two mechanisms, where the key is to provide the agent with candidate waypoints at any position in space.

4. Candidate Waypoints Predictor

Following the above observations, in order to bridge the discrete-to-continuous gap, we propose a candidate waypoints predictor that generates virtual waypoints for agents in continuous environments. At each navigational step, the waypoints predictor infers a local sub-graph which consists of a set of edges pointing from the agent towards accessible positions in space. As a result, VLN in continuous spaces can be performed effectively using high-level actions.

4.1. Network Architecture and Processing

As shown in Figure 4(a), the waypoints predictor has three key modules; visual encoders, a multi-layer Transformer, and a non-linear classifier. We apply two ResNet-50 [21], one pre-trained on ImageNet [46] and another one pre-trained for point-goal navigation [55], for encoding RGB and depth images, respectively. The Transformer network, consisting of two layers each with 12 self-attention heads, is applied for modeling the spatial relationships between views. The classifier is a multi-layer perceptron that projects the Transformer outputs to probabilities of adjacent waypoints in space. The parameters of the visual encoders are fixed after initialization, whereas the Transformer and the classifier will be updated during training.

Given an arbitrary point in the openspace of an environment, we first gather its RGB and depth panoramas, each panorama is represented by 12 single-view images at 30 degrees separation. These images will be encoded by the visual encoders to become a sequence of RGB and depth features, denoted as $\langle \boldsymbol{v}_1^{rgb}, \boldsymbol{v}_2^{rgb}, \dots, \boldsymbol{v}_{12}^{rgb} \mid \boldsymbol{v}_i^{rgb} \in \mathcal{V}^{rgb} \rangle$ and $\langle \boldsymbol{v}_1^{d}, \boldsymbol{v}_2^{d}, \dots, \boldsymbol{v}_{12}^{d} \mid \boldsymbol{v}_i^{d} \in \mathcal{V}^{d} \rangle$, respectively. Each pair of $\boldsymbol{v}_i^{rgb}$ and $\boldsymbol{v}_i^{d}$ feature is merged by a non-linear layer $\boldsymbol{W}^m$, yielding $\boldsymbol{v}_i^{rgbd}$. All 12 visual representations $\boldsymbol{v}_i^{rgbd}$ are sent to the Transformer in parallel for modeling relationship and inferring adjacent waypoints. Since each single-view image is square and has 90 degrees field of view that covers three 30 degrees sectors, we constrain the self-attention of each $\boldsymbol{v}_i^{rgbd}$ to be performed with a single neighbor at each side, *i.e.* with $\boldsymbol{v}_{i-1}^{rgbd}$ and $\boldsymbol{v}_{i+1}^{rgbd}$. Feature tokens outputs by the transformer $\tilde{\boldsymbol{v}}_i^{rgbd}$, where each contains information over a sector centered at the center of image i, will be fed to a classifier to predict a heatmap of 120 angles-by-12 distances. Each angle is of 3 degrees, and the distances range from 0.25 meters to 3.00 meters with 0.25 meters separation. By performing non-maximum-suppression (NMS) over the resulting heatmap, we obtain K neighboring waypoints.

4.2. Connectivity Graphs in Habitat-MP3D

To obtain the ground-truth connectivity graph $\mathcal{G}^*$ for training the waypoint predictor, we adapt the connectivity graph pre-defined for MP3D ($\mathcal{G}^{MP3D}$) to fit the continuous environments in Habitat-MP3D. Notice that the two graphs only reflect partial accessibility in space, nodes defined on the graphs are more inclined to sparse keypoints in the environment which requires a navigation agent to make a crucial decision, and edges indicate directions worth exploring rather than simply navigable (Figure 3). For example, it is important for an agent to predict a waypoint in front of a doorway so that the agent can decide whether to enter the room once it reaches the waypoint. In $\mathcal{G}^{MP3D}$, there exists edges that go across obstacles, which are inappropriate for predicting accessible waypoint. In contrast, all nodes and edges on $\mathcal{G}^*$ are defined in openspace. Additional nodes are added to ensure that the graph is connected. Overall, the entire set of connectivity graphs contains 13,358 (10,559) nodes spread across 90 Habitat-MP3D (MP3D)

[3]Model #1 in the table.

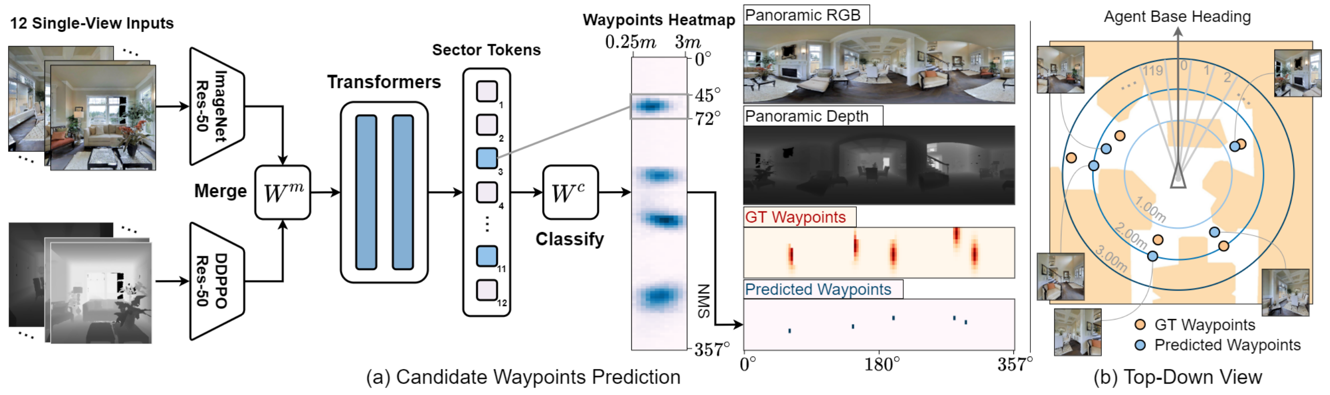

Figure 4. The candidate waypoints predictor. The module takes RGBD panoramic inputs, uses a multi-layer Transformer to model spatial relationships, and predicts the positions of waypoints in agent's neighborhood.

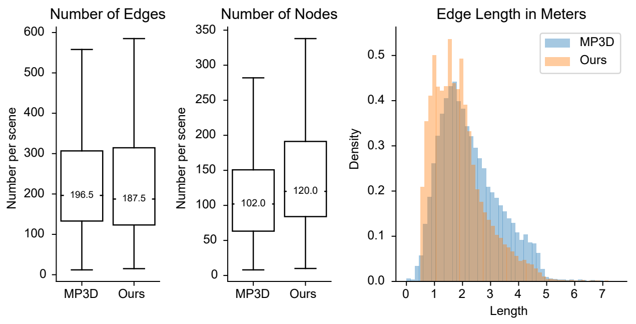

Figure 5. Statistics of connectivity graphs in MP3D and Habitat-MP3D (Ours) environments.

#	Model	MP3D Train				MP3D Val-Unseen			
		$\|\Delta\|$	%Open ↑	d_C ↓	d_H ↓	$\|\Delta\|$	%Open ↑	d_C ↓	d_H ↓
h	Baseline	1.29	81.73	1.13	2.17	1.37	80.18	1.08	2.06
i	U-Net	1.15	63.60	1.05	2.10	1.21	52.54	1.01	2.00
j	Ours	1.30	82.56	1.12	2.13	1.40	79.86	1.07	2.00

Table 3. Comparison of the waypoint predictors. M#h indicates the first row (Model #h) in the table.

environments, where each node is connected by in averaged 3.31 (4.07) edges with length in average 1.87 (2.26) meters. More statistics are shown in Figure 5.

4.3. Confirmation of Predictor Performance

Dataset For each node on $\mathcal{G}*$, we construct its local subgraph by positioning adjacent waypoints over a discretized space. As shown in Figure 4(b), we first partition the 3 meters radius range around each node into 120 3-degree sectors and 12 0.25-meter rings. Then, we assign waypoints into their corresponding partitions, and convert the discretized circle into a 120-by-12 heatmap which contains the positions of the waypoints. The heatmap is applied as prediction ground-truth, and each waypoint on the heatmap is represented as a Gaussian distribution with variance of 1.75m and 15° to allow some tolerance for the prediction. We follow the data splits in R2R to divide the connectivity graphs, using only 61 graphs (9,556 nodes) to train the predictor.

Training and Results We train the candidate waypoints predictor by minimizing the mean squared error between the predicted heatmaps and the ground-truth heatmaps. As shown in Table 3, we compare the results with (M#h) a baseline model which does not employ a Transformer so that each single-view image is projected to a sector of waypoint scores independently, and (M#i) a convolutional U-Net [45] which is similar to the sub-goal module in Sim2Real [2]. We set the maximum number of prediction from each heatmap to be 5. All models are trained on a NVIDIA 3090 GPU with a learning rate of 10^{-6} and batch size 64 using the AdamW optimizer [35].

We evaluate the performance of the waypoint predictors using four metrics: $|\Delta|$ measures the difference in number of target waypoints and predicted waypoints. %Open measures the ratio of predicted waypoints that is in open space (not hindered by any obstacle). d_C and d_H are the Chamfer distance and the Hausdorff distance, respectively, which are commonly used metrics for measuring the distance between point clouds. As shown in Table 3, U-Net results in the waypoints closest to the ground-truths, but a large portion of the predictions are block by obstacles, which is inappropriate for navigation. Comparing to the baseline, our model with the Transformer achieves similar %Open but lower distances, which will be applied in the following sections.

5. Bridging the Discrete to Continuous Gap

Thanks to the waypoints predictor, agents designed for high-level actions and pre-trained on graphs [24] can be transferred to continuous environments without any predefined graph. In this section, we demonstrate that the performance gap between the two navigation setups is significantly reduced. Furthermore, we show that using augmented waypoints for training leads to more robust agents, and new state-of-the-art results can be achieved on the benchmarking R2R-CE [3, 32] and RxR-CE [33] datasets.

Methods	#	Train Connectivity		Val Connectivity		R2R-CE Val-Seen					R2R-CE Val-Unseen				
		Graph	Predictor	Graph	Predictor	TL	NE↓	nDTW↑	SR↑	SPL↑	TL	NE↓	nDTW↑	SR↑	SPL↑
CMA	1	✓		✓		10.21	6.28	58.37	41.48	36.58	10.47	6.51	55.41	39.32	33.89
	2	✓			Freeze	8.71	6.83	53.16	33.83	30.23	8.28	6.81	52.86	33.50	29.92
	3		Freeze		Freeze	9.39	5.81	59.06	43.89	40.20	8.77	6.50	54.34	37.49	33.90
	4		Augmented		Freeze	9.57	5.36	61.46	48.05	43.89	9.11	6.19	55.56	40.80	36.73
	5	Low-Level Actions				8.87	7.20	50.67	29.53	27.29	8.22	7.54	49.19	27.29	24.97
VLN↻BERT	6	✓		✓		14.21	4.63	61.46	52.21	42.53	14.34	5.22	57.71	48.89	40.36
	7	✓			Freeze	11.42	5.62	56.19	43.22	37.10	11.22	5.91	53.35	39.94	34.42
	8		Freeze		Freeze	11.24	5.06	59.46	49.40	43.43	12.12	5.68	53.50	42.96	38.53
	9		Augmented		Freeze	11.83	5.07	59.18	52.35	46.08	11.85	5.52	54.20	45.19	39.91
	10	Low-Level Actions				8.47	7.40	48.07	26.85	25.19	7.42	7.66	47.71	23.19	21.67

Table 4. Agent performance in R2R-CE. 4% of samples which does not have a valid discrete path are deleted (details in Appendix), and all ground-truth paths are set to the shortest paths on graph for a fair comparison between navigation with and without the connectivity graph. *Graph* and *Predictor* means applying ground-truth graph and using either *Freezed* or *Augmented* predicted waypoints, respectively.

5.1. Setups

Datasets We evaluate the performance of agents on two datasets, R2R-CE [3, 32] and RxR-CE [33], using the Habitat simulator [48]. Both datasets are collected based on the discrete MP3D environments, including 61 environments for training, 11 for unseen validation and 18 reserved environments for testing. SPL [1] and nDTW [26] are the main evaluation metrics for R2R-CE and RxR-CE, respectively.

Experiments Our main experiment consists of two parts; First, we compare the agent's performance with and without using the connectivity graphs $\mathcal{G}^*$ (Table 4). For fairness, we filter out 4% of samples from the original R2R-CE which has a continuous ground-truth path that cannot be converted to a valid discrete path on $\mathcal{G}^*$ that honors the navigation instruction, and the start and end points of some paths have been slightly adjusted so that all point are on $\mathcal{G}^*$. Second, we compare our method to previous approaches (Table 5 and Table 6). In this case, we evaluate our agents on the untouched validation and test data of R2R-CE and RxR-CE.

Navigators We experiment with two navigators that have been widely applied in previous works while distinct in network architecture or navigation method. (1) CMA [54] is a simple sequence-to-sequence network with visual and language attentions as explained in Eq.1 and Eq.2. (2) The Recurrent VLN-BERT (VLN↻BERT) [24] is a Transformer-based network pre-trained for discrete VLN on R2R, it leverages the pre-defined `[CLS]` token in BERT as the agent's state and applies it for multi-modal attentions. For CMA, the entire network, including the language embeddings, is trained from scratch. Whereas the VLN↻BERT is initialized from the pre-trained Transformers [20]. We refer the readers to Appendix for details about the network architectures and the different configurations of agents in R2R-CE and RxR-CE.

Training All experiments are conducted on a single NVIDIA RTX 3090 GPU based on the PyTorch framework [42] and the Habitat simulator [48]. We apply a simple cross-entropy loss between the ground-truth actions and agent's predictions as the learning objective, which is minimized with an AdamW optimizer [35] during training. All the agents are trained using schedule sampling [5] with a decay frequency (moving from teacher forcing to student forcing) of per 5 epochs. As in previous work [32], for the CMA agents, the decay ratio is set to 0.75 and the learning rate is 10^{-4}. For the VLN↻BERT, the decay ratio is set to 0.50 (0.75 in RxR-CE due to the longer paths) and the learning rate is 10^{-5}. For agents which navigate on graphs, we consider the candidate node which has the shortest Dijkstra path to the target as ground-truth. For agents which navigate with predicted waypoints, we use a trained candidate waypoints predictor (from §4) to generate accessible sub-goals to allow the agents perform *view selection*. The ground-truth waypoint is the one which has the shortest geodesic distance to the target and to the sub-goal in R2R-CE and RxR-CE, respectively (see Appendix). Once a waypoint is chosen, instead of teleporting directly to the waypoint, we decompose such high-level goal to low-level controls, then, ask the agent to execute those controls progressively.

5.2. Main Results

Bridge the Gap As shown in Table 4, agents navigate on the ground-truth graphs (M#1, M#6) perform significantly better than agents navigate with low-level actions (M#5, M#10), scoring 8.92% and 18.69% higher SPL scores for the CMA and VLN↻BERT, respectively. M#2 and M#7 show that when the connectivity graph is absent, using a waypoints predictor can bring up the agents' performance by a large margin. Moreover, if the agents are trained on the predicted waypoints (M#3, M#8), so that the navigators can learn to adapt the patterns of generated waypoints, then agents can achieve similar performance as navigating on the ground-truth graphs – For CMA, the agent scores the

Methods	R2R-CE Val-Seen						R2R-CE Val-Unseen						R2R-CE Test-Unseen				
	TL	NE↓	nDTW↑	OSR↑	SR↑	SPL↑	TL	NE↓	nDTW↑	OSR↑	SR↑	SPL↑	TL	NE↓	OSR↑	SR↑	SPL↑
VLN-CE [32]	9.26	7.12	54	46	37	35	8.64	7.37	51	40	32	30	8.85	7.91	36	28	25
LAW [44]	–	–	58	–	40	37	–	–	54	–	35	31	–	–	–	–	–
SASRA [28]	8.89	7.17	53	–	36	34	7.89	8.32	47	–	24	22	–	–	–	–	–
Waypoint Models [31]	8.54	5.48	–	53	46	43	7.62	6.31	–	40	36	34	8.02	6.65	37	32	30
Ours (CMA)	11.47	5.20	61	61	51	45	10.90	6.20	55	52	41	36	11.85	**6.30**	**49**	**38**	**33**
Ours (VLN↻BERT)	12.50	5.02	58	59	50	44	12.23	5.74	54	53	44	39	13.31	**5.89**	**51**	**42**	**36**

Table 5. Comparison on agent performance in R2R-CE. Samples in all data splits are identical as in VLN-CE [32].

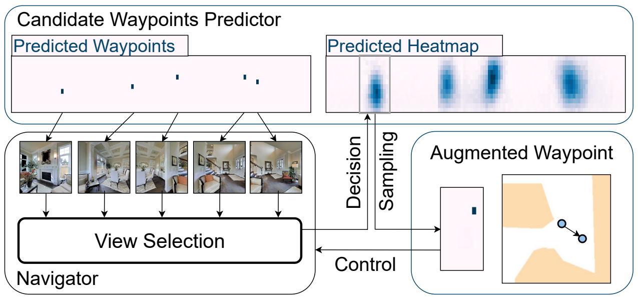

Figure 6. Waypoint Augmentation.

same SPL as M#1. For VLN↻BERT, the gaps of SR and SPL are reduced by 77% and 90%, respectively. Trace back to early experiments with fixed step length, there are about 15% of steps executed by M#6 in Table 2 collides with obstacles, while this number is only 7% once the waypoints predictor is introduced, which again reflects the importance of the predictor. However, in some rare cases our waypoints predictor may miss a direction connecting the agent to the destination (such as traversing stairs) resulting in a navigation failure. Sim2Real-VLN [2] suggests that the quality of the predicted waypoint candidates is a critical bottleneck of the agent's performance, whereas our results indicate that overall our waypoints predictor is generalizable to new environments, which successfully allows the agents to learn and act with high-level actions. We refer to the Appendix for visualization of the predicted waypoints and navigation trajectories, as well as further discussion on limitations.

Waypoint Augmentation We propose to augment the waypoints produced by the candidate waypoints predictor while training the agent to enhance the agent's generalization ability. As shown in Figure 6, the method samples a new waypoint from a patch of heatmap which corresponds to the agent's selected view. The sampled waypoints will take the agent to new positions in space, which requires the agent to observe diverse views, travel with different step lengths and interacts with different obstacles to complete the same navigation task. Results in Table 4 show that the agents' performances are further improved by training with augmented waypoints (M#4, M#9); the SPL of CMA in unseen environments even exceeds CMA navigating on graphs (M#1). These results indicate the effectiveness of augmenting waypoints when applying high-level actions.

Methods	RxR-CE Test-Unseen					
	TL	NE↓	SR↑	SPL↑	nDTW↑	SDTW↑
VLN-CE [32]	7.33	12.1	13.93	11.96	30.86	11.01
Ours (CMA)	20.04	**10.4**	**24.08**	**19.07**	**37.39**	**18.65**
Ours (VLN↻BERT)	20.09	**10.4**	**24.85**	**19.61**	**37.30**	**19.05**

Table 6. Comparison on agent performance in RxR-CE.

Comparison to SoTA We train the agents with predicted waypoints on the original R2R-CE and RxR-CE datasets, respectively, and compare the results with the previous state-of-the-arts[4] (Table 5, Table 6). Our method significantly outperforms previous methods across all dataset splits. On the test server of R2R-CE, our CMA agent improves over the CMA in Waypoint Models [31] by 6% SR and 3% SPL. It is also worth mentioning that comparing to the Waypoint Models which applies DDPPO [55] and 64 GPUs (5 days) for training the agent, our approach enables an efficient imitation learning, which drastically reduces the training cost to a single GPU (3.5 days) while achieving better results. On the multilingual RxR-CE dataset, both CMA and VLN↻BERT achieve more than 10% higher SR and 6% higher nDTW than the previous best model.

6. Conclusion

In this paper, we introduce a candidate waypoints predictor to produce accessible waypoints in space, which enables agents designed for discrete environments to learn and navigate in continuous environments with high-level actions. Experiments show that our method is generalizable for different agents and to unseen environments, it greatly bridges the discrete-to-continuous gap and achieves new state-of-the-art performances on the R2R-CE and the RxR-CE datasets. We believe this work is an important step towards taking VLN research in MP3D to the more realistic scenarios, including simulation in continuous environments and even robots in the real-world. For future work, weakly supervised and language-conditioned waypoint prediction could be explored. Moreover, the idea of predicting adjacent waypoints and performing *navigable-view selection* have the great potential to be applied in a wide range of research such as PointNav [48], ObjectNav [4], Audio-Visual Nav [9] and embodied task-completion problems [50].

[4]R2R-CE Leaderboard: https://eval.ai/web/challenges/challenge-page/719/leaderboard/1966, RxR-CE Leaderboard: https://ai.google.com/research/rxr/habitat

References

[1] 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. 4, 7

[2] Peter Anderson, Ayush Shrivastava, Joanne Truong, Arjun Majumdar, Devi Parikh, Dhruv Batra, and Stefan Lee. Sim-to-real transfer for vision-and-language navigation. In *Conference on Robot Learning*, pages 671–681. PMLR, 2021. 2, 3, 6, 8

[3] 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. 2, 3, 4, 6, 7

[4] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 8

[5] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1171–1179, 2015. 7

[6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE, 2017. 2, 3

[7] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations*, 2019. 3

[8] 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. 3

[9] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 17–36. Springer, 2020. 8

[10] Changan Chen, Sagnik Majumder, Ziad Al-Halah, Ruohan Gao, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Learning to set waypoints for audio-visual navigation. In *International Conference on Learning Representations*, 2020. 3

[11] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2019. 2

[12] Kevin Chen, Junshen K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11276–11286, 2021. 3

[13] Kevin Chen, Juan Pablo de Vicente, Gabriel Sepulveda, Fei Xia, Alvaro Soto, Marynel Vázquez, and Silvio Savarese. A behavioral approach to visual navigation with graph localization networks. *arXiv preprint arXiv:1903.00445*, 2019. 3

[14] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *International Conference on Learning Representations*, 2018. 3

[15] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2018. 5

[16] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. *arXiv preprint arXiv:2007.05655*, 2020. 2

[17] 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 *Advances in Neural Information Processing Systems*, pages 3314–3325, 2018. 2, 3, 4

[18] Tsu-Jui Fu, Xin Eric Wang, Matthew F Peterson, Scott T Grafton, Miguel P Eckstein, and William Yang Wang. Counterfactual vision-and-language navigation via adversarial path sampler. In *European Conference on Computer Vision*, pages 71–86. Springer, 2020. 2

[19] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2017. 3

[20] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. 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*, pages 13137–13146, 2020. 2, 3, 7

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5

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

[23] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3

[24] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language

bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1643–1653, June 2021. 2, 3, 6, 7

[25] Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. Are you looking? grounding to multiple modalities in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6551–6557, 2019. 3

[26] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019. 4, 7

[27] Muhammad Zubair Irshad, Chih-Yao Ma, and Zsolt Kira. Hierarchical cross-modal agent for robotics vision-and-language navigation. *arXiv preprint arXiv:2104.10674*, 2021. 3

[28] Muhammad Zubair Irshad, Niluthpol Chowdhury Mithun, Zachary Seymour, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Sasra: Semantically-aware spatio-temporal reasoning agent for vision-and-language navigation in continuous environments. *arXiv preprint arXiv:2108.11945*, 2021. 3, 8

[29] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6741–6749, 2019. 2

[30] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 3

[31] Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15162–15171, 2021. 3, 8

[32] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, 2020. 2, 3, 4, 6, 7, 8

[33] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412, 2020. 2, 3, 6, 7

[34] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah A Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. 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)*, pages 1494–1499, 2019. 3

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6, 7

[36] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2

[37] 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 Conference on Computer Vision and Pattern Recognition*, pages 6732–6740, 2019. 2

[38] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. *In Proceedings of the European Conference on Computer Vision*, 2020. 2, 3

[39] Xiangyun Meng, Nathan Ratliff, Yu Xiang, and Dieter Fox. Scaling local control to large-scale topological navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 672–678. IEEE, 2020. 3

[40] Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. 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)*, pages 684–695, 2019. 2

[41] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Qinfeng Shi, and Anton van den Hengel. Counterfactual vision-and-language navigation: Unravelling the unseen. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 7

[43] 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. 2

[44] Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel Chang. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4018–4028, 2021. 3, 8

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 6

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5

[47] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations*, 2018. 3

[48] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 2, 3, 4, 7, 8

[49] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Shyamal Buch, Claudia D'Arpino, Sanjana Srivastava, Lyne P Tchapmi, et al. igibson, a simulation environment for interactive tasks in large realisticscenes. *arXiv preprint arXiv:2012.02924*, 2020. 3

[50] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020. 8

[51] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of NAACL-HLT*, pages 2610–2621, 2019. 2

[52] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406, 2020. 2

[53] 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, 2021. 2

[54] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019. 2, 3, 4, 7

[55] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations*, 2019. 5, 8

[56] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, 2018. 3

[57] 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 Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018. 3

[58] Claudia Yan, Dipendra Misra, Andrew Bennnett, Aaron Walsman, Yonatan Bisk, and Yoav Artzi. Chalet: Cornell house agent learning environment. *arXiv preprint arXiv:1801.07357*, 2018. 3

[59] 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. 2