Volumetric Environment Representation for Vision-Language Navigation

Rui Liu Wenguan Wang Yi Yang*
ReLER, CCAI, Zhejiang University
https://github.com/DefaultRui/VLN-VER

Abstract

Vision-language navigation (VLN) requires an agent to navigate through an 3D environment based on visual observations and natural language instructions. It is clear that the pivotal factor for successful navigation lies in the comprehensive scene understanding. Previous VLN agents employ monocular frameworks to extract 2D features of perspective views directly. Though straightforward, they struggle for capturing 3D geometry and semantics, leading to a partial and incomplete environment representation. To achieve a comprehensive 3D representation with fine-grained details, we introduce a Volumetric Environment Representation (VER), which voxelizes the physical world into structured 3D cells. For each cell, VER aggregates multi-view 2D features into such a unified 3D space via 2D-3D sampling. Through coarse-to-fine feature extraction and multi-task learning for VER, our agent predicts 3D occupancy, 3D room layout, and 3D bounding boxes jointly. Based on online collected VERs, our agent performs volume state estimation and builds episodic memory for predicting the next step. Experimental results show our environment representations from multi-task learning lead to evident performance gains on VLN. Our model achieves state-of-the-art performance across VLN benchmarks (R2R, REVERIE, and R4R).

1. Introduction

Vision-language navigation (VLN) requires an agent to navigate in a 3D environment following natural language instructions [3, 64]. As a holistic understanding of the environment plays a pivotal role in decision-making within VLN, environment representation learning serves as a foundation for formulating accurate navigation policies.

Early VLN approaches [3, 23] typically learn the navigation policy through the sequence-to-sequence (Seq2Seq) framework [72], which directly maps instructions and multi-view perspective observations to actions. They simply embed their immediate observation of local environment into the hidden states of recurrent units. As a result, they lack of explicit environment representations and struggle to access their past states during long-time exploration [61, 82]. To address this issue, later VLN agents are equipped with an external memory module [61], which stores the environment representations distinctly from navigation states. In this way, they can explicitly model and maintain the environment layouts and contents in a form of topological graph [2, 11, 16, 19, 32, 62] or semantic map [1, 4, 10, 12, 26, 34, 38, 55, 75, 94] (Fig. 1). Despite their promising performance with advanced frameworks (*e.g.*, graph neural network [44] and Transformer [79]), their environment representations are still built upon 2D perspective features extracted by monocular frameworks. While straightforward, they compress depth information onto the perspective plane, sacrificing the integral scene structure in the 3D space. Thus, they encounter challenges in capturing 3D geometry and semantics in complex scenes. Such an incomplete environment representation easily leads to sub-optimal navigation decisions.

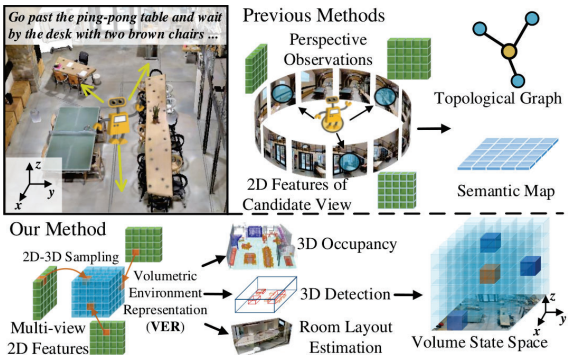

Figure 1. The agent observes its surroundings with corresponding perspective features of different candidate views (●). Previous methods construct the topological graph or semantic map based on these 2D features. Our VER aggregates the multi-view features into structured 3D cells via 2D-3D sampling. VER is a powerful representation for both 3D perception tasks and VLN, providing a volume state space for decision-making.

*Corresponding author: Yi Yang.

In this article, we propose a Volumetric Environment Representation (VER) that quantizes the physical world into structured 3D cells (Fig. 1). These cells, arranged within a predefined volumetric space, maintain both height and depth dimensions. Each cell corresponds to local context of the 3D space. VER aggregates multi-perspective 2D features within these cells through an *environment encoder* (§3.1). Compared to previous partial representations derived from hidden states and external memory, our VER captures the full geometry and semantics of the physical world. These 3D cells stores the properties of the corresponding space in the scene by predicting 3D occupancy [70, 77], room layout [106], and 3D object boxes [51]. However, directly reconstructing the high-quality VER from 2D perspective views is challenging to capture the fine-grained details. As a response, we propose a *coarse-to-fine* VER extraction architecture, which uses learnable up-sampling operations to construct the representations progressively. It is supervised by multi-resolution semantic labels at different scales, utilizing the coarse-to-fine representations as hierarchical inputs. The annotations of the 3D tasks are collected for multi-task learning (§3.4).

At each navigation step, our agent initially encodes the multi-view observations into VER (§3.2). With VER, instructions can be more effectively grounded in the 3D context. This is achieved by establishing cross-modal correlations between linguistic words and 3D cells of VER. Based on the correlations, a *volume state estimation* module is proposed to calculate transition probabilities over the surrounding cells. With the help of this module, our agent performs comprehensive decision-making in volumetric space, and then maps the volume state into local action space. In addition, an *episodic memory* module is established to online collect the information of observed viewpoints and build a topological graph providing global action space (§3.3). The node embeddings in the graph are from neighbor pillar representations in VER corresponding to the respective viewpoints. To balance the long-range action reasoning and language grounding, our agent combines both the local action probabilities derived from the volume state and the global action probabilities obtained from the episodic memory.

Our agent is evaluated on three VLN benchmarks, *i.e.*, R2R [3], REVERIE [64], and R4R [39] (§4.1). It yields solid performance gains (about **3**% SR and **4**% SPL on R2R *test*, **4**% SR and **4**% SPL on REVERIE *val unseen*). The ablation study confirms the efficacy of core model designs (§4.2). Additional results show our model achieves promising performance in 3D occupancy prediction, 3D detection, and room layout estimation (§4.3).

2. Related Work

Vision-Language Navigation (VLN). Early VLN agents [3, 23] are built upon Seq2Seq [66, 72] framework to reserve the observation history in hidden state. Thus they struggle to capture long-range context as the path length increases. Later efforts are devoted to multimodal representation learning, navigation strategy learning, and data generation [27]. As a primary step, *multimodal representation learning* helps agents understand the environments and establish relations between the instructions and visual observations. Inspired by the success of vision-language pretraining [58, 65, 73], recent approaches [14, 33] use transformer-based architectures [42, 79] for joint visual and textual representations. Some attempts further exploit the visual information by modeling semantic relation [37] and spatial information [11, 19, 32, 83, 86, 100]. For *navigation strategy learning*, many VLN models [24, 88] use imitation and reinforcement learning-based training strategies. Previous solutions [45, 85] introduce world models [29] to perform mental simulations and make mental planning. Furthermore, the scarcity of human instructions and limited diversity of the scene hinder the agent to learn navigation policy and generalize to unseen environments well [54]. Therefore, several VLN *data generation* strategies have been proposed to create new trajectories from existing datasets [15, 28, 93], generate more instructions [41, 74, 84, 90], or create synthetic environments [48, 49]. In addition, driven by large models [5], existing agents [13, 57, 71, 101] demonstrate promising zero-shot performance.

Despite their outstanding contributions, most of them rely on 2D visual cues in perspective observations. These representations are constrained by occlusion and limited geometric information, especially in complex scenes. In this paper, we propose VER, a unified environment representation learned by 3D perception tasks. During navigation, our agent performs volume state estimation on VER, facilitating comprehensive decision-making within the 3D space.

Environment Representation. Existing VLN models introduce various representations for environment modeling, including topological graphs [2, 11, 16, 19, 32, 82], and semantic spatial representations [1, 4, 12, 26, 34, 38, 55, 75, 94]. With a broader view, diverse representations have been proposed for robotics and autonomous driving [25, 47, 68, 76, 96]. In the early stage, 2D occupancy grid maps [22] model occupied and free space in the surroundings based on Bayesian estimation for robot navigation. Classic SLAM systems [21] construct a map directly by integrating information from various sensors, including LiDAR and cameras. However, the representations in SLAM still rely on primitives such as 3D point clouds and image patches. Some efforts [6, 8, 31] focus on developing learnable semantic map representations. To enhance spatial reasoning, scene graph representations [9, 80] define the topological relations between spatial elements of the environment. In addition, neural scene representations [46, 50] embed image observations into latent codes

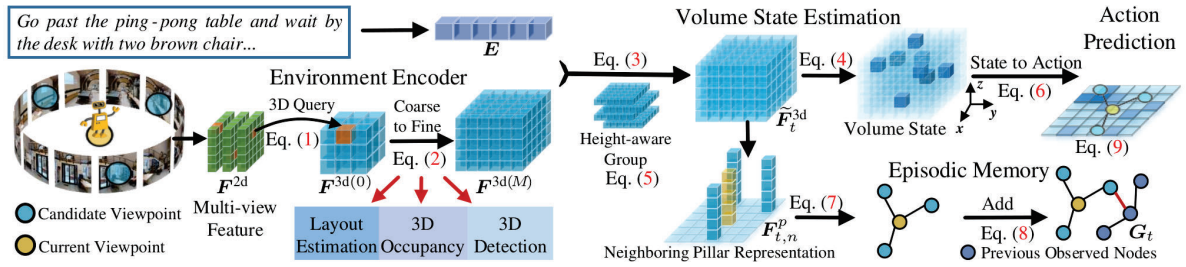

Figure 2. Overview of our model. Given the perspective features of candidate views, a group of 3D queries are used to sample and aggregate them into VER (§3.1). To encode VER, we adopt *coarse-to-fine extraction* and perform multi-task learning on 3D perception. Based on VER, a *volume state estimation* module is proposed to predict state transition (§3.2). The *episodic memory* is used to store past observations using neighboring pillar representations for each viewpoint (§3.3). For decision-making, our agent combines both the local action probabilities from the volume state and the global action probabilities obtained from the episodic memory. See §3 for more details.

for object categories, showcasing scalability to large scenes [40, 59]. Vision-centric BEV perception, which transforms perspective-view inputs to BEV grid representations, has recently received increasing attention [47, 51, 63]. As the BEV representations simplify the vertical geometry, 3D occupancy prediction [36, 78] is further proposed to infer the 3D geometry from perspective images [68, 70, 87, 98].

However, existing VLN agents mainly employ abstract relations or compressed spatial maps, lacking the ability to access complete scene information. Providing more world context can be beneficial for the subsequent decision-making and policy learning. Motivated by this insight, we explore a holistic environment representation VER that voxelizes the world into structured 3D cells. VER captures both semantic information and geometric details of the whole scene. Building upon VER, our agent is able to predict the 3D occupancy, room layout, and 3D boxes accurately.

3. Approach

Problem Formulation. For brevity, we present the technical description in the context of R2R [3]. The navigable area of the environment is organized as an undirected graph, containing a set of nodes (viewpoints) and connectivity edges. In R2R, an embodied agent needs to navigate to a target location in the 3D environment following human instructions with L words (embedded as $\boldsymbol{E}\in\mathbb{R}^{D_w\times L}$, where D_w is the channel dimension). At time step t, the agent looks around and obtains multi-view observations of its surrounding scene from the current location. Each view is represented by a 2D visual feature $\boldsymbol{F}_t^{\text{2d}}\in\mathbb{R}^{D_i\times H\times W}$, where H and W are the spatial shape of image plane, D_i denotes the channel dimension. The local action space $\mathcal{A}_t\in\mathbb{R}^{N_t+1}$ is defined by N_t candidate views, which correspond to neighboring navigable nodes $\{v_{t,n}^*\}_{n=1}^{N_t}$, as well as a [STOP] action. Previous agents predict the action probabilities $\boldsymbol{p}_t^{\text{2d}}\in\mathbb{R}^{N_t+1}$ directly based on $\boldsymbol{F}_t^{\text{2d}}$ of each candidate view. However, these 2D features with limited geometric information are partial representations of the 3D environment, easily leading to suboptimal decision making.

Overview. To achieve comprehensive scene understanding, we introduce VER, which voxelizes the 3D world into structured 3D cells (Fig. 2). At step t, an *environment encoder* is proposed to sample multi-view features ($\boldsymbol{F}_t^{\text{2d}}$ of each view) into the volumetric space of VER, forming a unified representation $\boldsymbol{F}_t^{\text{3d}}\in\mathbb{R}^{D_e\times X\times Y\times Z}$ (§3.1). X and Y are the shapes of horizontal plane, Z reserves the height information of 3D space, and D_e represents the channel dimension. The volumetric space aligns with gravity in the world coordinate system based on the Manhattan assumption [106]. To encode VER, we devise *coarse-to-fine extraction* with multiple 3D perception tasks supervised by multi-resolution annotations (§3.4). Based on VER, a *volume state estimation* module is proposed to predict state transition probabilities $\boldsymbol{p}_t^{\text{3d}}\in\mathbb{R}^{X\times Y\times Z}$ over surrounding 3D cells (§3.2). With this module, our agent performs comprehensive decision-making in the 3D space and maps $\boldsymbol{p}_t^{\text{3d}}$ to $\boldsymbol{p}_t^{\text{2d}}$. To predict the next step, our agent combines both the local action probabilities derived from the volume state and the global action probabilities obtained from the *episodic memory* (§3.3).

3.1. Environment Encoder

2D-3D Sampling. At step t, the agent observes its surroundings and acquires the multi-view images. We introduce cross-view attention (CVA) to aggregate their features ($\boldsymbol{F}^{\text{2d}}$ for each view) into a unified volumetric representation $\boldsymbol{F}^{\text{3d}}$ with a group of learnable volume queries $\boldsymbol{Q}\in\mathbb{R}^{D_e\times X\times Y\times Z}$ (t is omitted for simplicity). Specifically, for the 3D cell positioned at (x,y,z) within the egocentric world, a single query $\boldsymbol{Q}(x,y,z)\in\mathbb{R}^{D_e}$ is used to sample each image feature $\boldsymbol{F}^{\text{2d}}$ as:

$$\boldsymbol{F}^{\text{3d}}(x,y,z)=\text{CVA}\big(\boldsymbol{Q}(x,y,z),\boldsymbol{F}^{\text{2d}}(h',w')\big), \quad (1)$$

where (h',w') denotes the location of corresponding sampling point on the image plane. Note that we only show

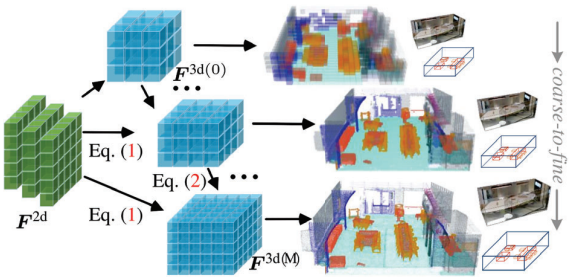

Figure 3. Our *coarse-to-fine* VER representation extraction (§3.1) adopts cascade up-sampling operations with 3D deconvolutions (Eq. 2) and 3D queries (Eq. 1). The training process is supervised at different scales by multi-resolution semantic labels.

the formulation of a single sampling point for conciseness. Since the sampling strategy of vanilla cross-attention is computationally expensive, the deformable attention [51, 104] is introduced and extended in CVA. In this way, $\boldsymbol{Q}(x, y, z)$ selectively attends to a set of key sampling points around a reference instead of the entire $\boldsymbol{F}^{2d}$.

Coarse-to-Fine VER Extraction Architecture. Directly recovering the fine-grained VER from perspective features easily leads to performance and efficiency degradation [17, 95]. The *coarse-to-fine* extraction is proposed to reconstruct VER progressively. Our approach involves cascade up-sampling operations (Fig. 3), dividing this extraction into M levels. At each level, 3D deconvolutions are utilized for lifting spatial resolution, and then CVA is used to query the multi-view 2D features (Eq. 1) for refining the detailed geometry. This enables the direct learning of details and avoids the inaccuracy of interpolation [56, 77, 92] (see Table 7). Between the input coarse feature $\boldsymbol{F}^{3d(0)} \in \mathbb{R}^{D_e \times \frac{X}{2^M} \times \frac{Y}{2^M} \times \frac{Z}{2^M}}$ from Eq. (1) and the target fine feature $\boldsymbol{F}^{3d(M)} \in \mathbb{R}^{D_e \times X \times Y \times Z}$, the intermediate features with varying shapes are calculated as follows:

$$\boldsymbol{F}^{3d(1)} = \uparrow \boldsymbol{F}^{3d(0)}, \cdots, \boldsymbol{F}^{3d(M)} = \uparrow \boldsymbol{F}^{3d(M-1)}, \tag{2}$$

where $\boldsymbol{F}^{3d(1)}, \cdots, \boldsymbol{F}^{3d(M-1)}$ denote the intermediate features from different levels, and '$\uparrow$' denotes the up-sampling.

Multi-task Learning. Our VER offers a unified scene representation for various 3D perception tasks. Existing studies [68, 70] highlight that semantics and geometry are tightly intertwined. We train our environment encoder to extract VER under the supervision of multiple 3D perception tasks (§3.4). This process utilizes $\{\boldsymbol{F}^{3d(0)}, \cdots, \boldsymbol{F}^{3d(M)}\}$ as inputs and is supervised at different scales (Fig. 3). For 3D occupancy prediction, the decoder is implemented as MLPs with the focal loss [52]. For 3D layout estimation, we adopt a query-based head to yield the manhattan room layouts. A combination of the L1 loss and the IoU loss [67] is used as the training objective. For 3D detection, we employ a detection head to predict the 3D boxes [51]. The bipartite matching and the bounding box loss [51, 104] are employed for detection. A weight vector $[2.0, 0.25, 0.25]$ is used to balance the optimization of the three tasks, respectively. During navigation, the agent traverses between different viewpoints and encodes VERs through the frozen environment encoder.

3.2. Volume State Estimation

VLN task is typically viewed as a state estimation and transition problem [76]. With our VER, the agent state is represented as "volume state". As such, the state transition within the locally observed 3D environment, computed by Eq. (3)&(4), is referred as "volume state estimation". Different from previous plane-level state models [1, 4], VER introduces additional height dimension for 3D state estimation. This enables a more accurate action prediction.

Volume State. At the beginning of a navigation episode, the agent is located at a start viewpoint (x_0, y_0, z_0). Based on its perception range, a volume state space $\mathcal{X} \in \mathbb{R}^{X \times Y \times Z}$ is defined corresponding to the 3D physical world with an initial state $s_0{=}(x_0, y_0, z_0)$. At step t, the next intermediate state $s_{t+1}{=}(x_{t+1}, y_{t+1}, z_{t+1})$ is determined by the instruction embeddings $\boldsymbol{E}$ and VER $\boldsymbol{F}_t^{3d}$ for reaching the goal state s_T ($0{<}t{<}T$). As the entire environment is partially observable, the current state transition ($s_t \rightarrow s_{t+1}$ in $\mathcal{X}$) is regarded as a local consideration for action prediction.

State Estimation. A *volume state estimation* module is devised to predict the probability distribution $\boldsymbol{p}_t^{3d} \in \mathbb{R}^{X \times Y \times Z}$ of the intermediate state s_{t+1} conditioned on $\boldsymbol{E}$ and $\boldsymbol{F}_t^{3d}$. The environment representation is first reshaped as $\boldsymbol{F}_t^{3d'} \in \mathbb{R}^{D_e \times XYZ}$, and then adopt multi-layer transformers (MLT) to model the relations between $\boldsymbol{E}$ and $\boldsymbol{F}_t^{3d'}$ as follows:

$$\widetilde{\boldsymbol{F}}_t^{3d} = \text{MLT}\big([\boldsymbol{E}; \boldsymbol{F}_t^{3d'}]\big) \in \mathbb{R}^{D_e \times XYZ}, \tag{3}$$

where $\widetilde{\boldsymbol{F}}_t^{3d}$ is the updated representations, and [;] denotes the concatenation operation. MLT consists of stacked self-attention blocks. Then we use MLPs for state estimation:

$$\boldsymbol{p}_t^{3d} = \text{Softmax}\big(\text{MLP}(\widetilde{\boldsymbol{F}}_t^{3d})\big) \in [0, 1]^{X \times Y \times Z}. \tag{4}$$

Efficient Height-aware Group. The computational and memory efficiency of Eq. (3) is compromised due to the resolution of $\boldsymbol{F}_t^{3d}$ ($XYZ \gg L$). Considering the similarity and sparsity of information along the height direction [97, 99], we partition $\boldsymbol{F}_t^{3d}$ into several uniform groups $\{\boldsymbol{F}_{t,z}^{g} \in \mathbb{R}^{D_e \times XY}\}_{z=1}^{Z}$ along this axis. Then we apply MLT to each group, and Eq. (3) is reformulated as:

$$\begin{aligned} \widetilde{\boldsymbol{F}}_{t,i}^{g} &= \text{MLT}\big([\boldsymbol{E}; \boldsymbol{F}_{t,i}^{g}]\big) \in \mathbb{R}^{D_e \times XY}, \\ \widetilde{\boldsymbol{F}}_t^{3d} &= \{\widetilde{\boldsymbol{F}}_{t,z}^{g}\}_{z=1}^{Z} \in \mathbb{R}^{D_e \times XYZ}, \end{aligned} \tag{5}$$

where the weights of MLT are shared among different groups. The updated features from different groups $\{\widetilde{\boldsymbol{F}}_{t,z}^{g}\}_{z=1}^{Z}$ are aggregated along the height to leverage complementary information. For ease of notation, the symbol

$\widetilde{\boldsymbol{F}}_t^{3d}$ is slightly reused for the gathered 3D representations. Then, $\boldsymbol{p}_t^{3d}$ is calculated by Eq. (4).

3.3. Action Prediction

For action prediction across the entire explored scene, a topological graph $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t\}$ is constructed and updated online to represent *episodic memory* during navigation. Specifically, $\mathcal{V}_t$ denotes the observed viewpoints, *i.e.*, all visited viewpoints and their candidate viewpoints. These viewpoints are encoded by compressing previous VERs. The edge $\mathcal{E}_t$ denotes the navigable connections between these viewpoints. To predict the next step, our agent combines both the volume state estimation and episodic navigation memory for decision-making.

Mapping Volume State to Action. As our agent navigates on the horizontal plane to reach the adjacent candidate viewpoints $\{v_n^*\}_{n=1}^{N_t}$, we map the volume state space into 2D space to align with this movement pattern. Specifically, we average $\boldsymbol{p}_t^{3d}$ along the height (z-axis) axis as $\boldsymbol{p}_t^h \in \mathbb{R}^{X \times Y}$. Then we sum probability values in the neighborhood of $\{v_{t,n}^*\}_{n=0}^{N_t}$ (v_0^* for the current viewpoint, *i.e.*, [STOP]), and normalize them as the local action probabilities:

$$\boldsymbol{p}_t^{2d} = \Big\{ \sum \boldsymbol{p}_t^h(x_n, y_n) \Big\}_{n=0}^{N_t} \in [0,1]^{N_t+1}, (x_n, y_n) \in \Omega_n, \quad (6)$$

where $\boldsymbol{p}_t^h(x_n, y_n)$ is the value at the coordinate (x_n, y_n), and Ω_n is the neighborhood of v_n^* in the horizontal plane. In the training stage, a heatmap [102] with a Gaussian kernel is used to supervise this action prediction (§3.5).

Global Action Prediction. The *episodic memory* module is used to store past environment representations and allows easy access to them. To memory the environment representations efficiently, we use the neighboring pillar [91] representations $\boldsymbol{F}_{t,n}^p \in \mathbb{R}^{D_e \times |\Omega_n| \times Z}$, corresponding to the current observed viewpoints $\{v_n^*\}_{n=0}^{N_t}$ at step t:

$$\boldsymbol{F}_{t,n}^p = \{\widetilde{\boldsymbol{F}}_t^{3d}(x_n, y_n, z_n)\}_{(x_n,y_n)\in\Omega_n, z_n \in [1,Z]} \quad (7)$$

where $\widetilde{\boldsymbol{F}}_t^{3d}(x_n, y_n, z_n)$ is the representation at position (x_n, y_n, z_n) of $\widetilde{\boldsymbol{F}}_t^{3d}$ (Eq. 5). $\{\bar{\boldsymbol{F}}_{t,n}^p \in \mathbb{R}^{D_e}\}_{n=0}^{N_t}$ is obtained by average pooling as the corresponding node embeddings, which are then incorporated into $\mathcal{G}_t$. For previously observed nodes, we compute the average of their features.

The episodic memory $\mathcal{G}_t$, which includes the observed viewpoints, offers a global action space $\mathcal{A}_t^* \in \mathbb{R}^{|\mathcal{V}_t|}$. This enables our agent to change its current navigation state easily by 'jumping' directly to another viewpoint, which may be even observed several steps ago. The global action probabilities on $\mathcal{G}_t$ are calculated as:

$$\begin{aligned} \hat{\boldsymbol{G}}_t &= \text{MLT}([\boldsymbol{E}; \boldsymbol{G}_t]) \in \mathbb{R}^{D_e \times |\mathcal{V}_t|}, \\ \boldsymbol{p}_t^g &= \text{Softmax}(\text{MLP}(\hat{\boldsymbol{G}}_t)) \in [0,1]^{|\mathcal{V}_t|}, \end{aligned} \quad (8)$$

where $\boldsymbol{G}_t$ denotes the node embeddings of $\mathcal{G}_t$. The ultimate action probabilities are given as:

$$\begin{aligned} \hat{\boldsymbol{p}}_t^{2d} &= [\boldsymbol{p}_t^{2d\to g}; \boldsymbol{p}_t^{2d}] \in [0,1]^{|\mathcal{V}_t|}, \\ \hat{\boldsymbol{p}}_t^g &= W_g \boldsymbol{p}_t^g + (1 - W_g)\hat{\boldsymbol{p}}_t^{2d} \in [0,1]^{|\mathcal{V}_t|}, \end{aligned} \quad (9)$$

where $\boldsymbol{p}_t^{2d\to g} \in \mathbb{R}^{|\mathcal{V}_t|-(N_t+1)}$ denotes the probabilities of global backtracking and we use the same value as the local [STOP] probability in Eq. (6); W_g is a learnable weight.

State Transition and Memory Update. After executing the action in $\mathcal{A}_t^*$, our agent reaches the next viewpoint $v_{t+1,0}^*$, and will iteratively: (1) encode its current observation as $\boldsymbol{F}_{t+1}^{3d}$ through Eq. (1); (2) update its volume state as $s_{t+1} = (x_{t+1}, y_{t+1}, z_{t+1})$; (3) add the node embeddings of $\{v_{t+1,n}^*\}_{n=0}^{N_{t+1}}$ into the episodic memory $\mathcal{G}_{t+1}$; and (4) predict the next step with the updated episodic memory (*i.e.*, $\mathcal{G}_{t+1}$) and volume state (*i.e.*, s_{t+1}). Our agent repeats the above process until it chooses the [STOP] action or reaches the maximum step limit.

3.4. Annotation Generation

A multi-task learning framework is proposed to extract and encode the VERs (§3.1). To achieve this, we generate annotations on Matterport3D dataset [7] for 3D occupancy prediction, object detection, and room layout estimation. We design a *room-object-voxel* pipeline to automatically generate these annotations. This pipeline leverages existing LiDAR point labels without additional human annotations (more details in Appendix). We utilize the egocentric observations with multi-view images as input.

Room Layout. The room layout in our context specifies the positions, orientations, and heights of the walls, relative to the camera center. It aims to reconstruct cuboid room shapes within the Manhattan world [18]. Given that the horizontal plane is aligned on the $x-z$ axis, we parameterize the layout with center coordinate, width, length, height, and rotation. In contrast to directly operating on a single panoramic image [105, 106], we use the embodied observations with multi-view images as input.

Object Detection. Based on the room layout, we collect the surrounding objects if the agent locates in a room. In a non-closed environment, we collect information about nearby objects based on their distance from the agent. For each object, the eigenvectors of its vertices are used to define an oriented bounding box that tightly encloses the object [35]. Considering some objects may disappear from view due to occlusion but still exist in the environment (referred to as permanence [98]), we also include them in the analysis.

Point Accumulation for Occupancy. To generate voxel labels for occupancy [77, 95], we accumulate the sparse LiDAR points and utilize 3D boxes. Given dense background and object points, we first voxelize the 3D space and label each voxel based on the majority vote of labelled points in that voxel. Due to the limited number of LiDAR points, we leverage the Nearest Neighbors algorithm to generate dense labels for remaining voxels by searching the nearest

semantic label. Moreover, the agent infers the complete 3D occupancy of each object (amodal perception), including regions that are not directly observed. This attribute enables the agent to predict the entire object instead of only visible surfaces [98].

Statistics. For high-resolution labels, we define $120\times120\times35$ voxel grids in world coordinates, where the scene voxel size equals to 0.1 m (see more details about multi-resolution labels in Appendix). We annotate over 50 billion voxels and 16 classes, including 11 foreground objects and 5 background stuffs. It comprises about $100k$ annotated bounding boxes and 1,500 room layouts within the scenes. These annotations follow the same *train/val/test* splits as R2R [3]. There are 61 scenes for *train/val seen*, 11 scenes for *val unseen*, and 18 scenes for *test*.

3.5. Implementation Details

Initially, the environment encoder (§3.1) is introduced for VER through *coarse-to-fine* extraction. Then multi-task learning is performed across multiple 3D perception tasks, including 3D occupancy prediction, 3D layout estimation, and 3D detection. During navigation, our agent is equipped with the frozen environment encoder and predicts the next step. Following recent VLN practice [14, 16, 33], both offline pretraining and finetuning are adopted. In this section, we will mainly introduce the details of architecture and training (see more details in Appendix).

Environment Representation Learning. For the multi-view images, we adopt ViT-B/16 [20] pretrained on ImageNet to extract features. The number of 3D volume queries is $15\times15\times4$. For each query, it is projected to sample 2D views according to intrinsic and extrinsic parameters of the camera. We set the perception range as $[-6m, 6m]$ for $x-y$ axis and $[-1.5m, 2m]$ for the height (z axis). We adopt six layers of CVA (Eq. 1) for 2D-3D sampling, and then use $M=3$ cascade deconvolutions for up-sampling (Eq. 2). The feature dimension is 768 (*i.e.*, D_i=D_w=D_e=768).

Navigation Network. MLT with 4 layers is initialized from [73] for state estimation (Eq. 4) and global action prediction (Eq. 8), respectively. The range of neighborhood for each candidate is set as $|\Omega_n| = 9$. The standard deviation of the Gaussian kernel for the heat map is set as 3.0. Based on this heat map, the focal loss [52] is used to supervise the local action prediction (Eq. 6). We also use a cross-entropy loss for the global action prediction (Eq. 9).

Pretraining. For R2R [3] and R4R [39], Masked Language Modeling [14, 42] and Single-step Action Prediction [14, 33] are adopted as auxiliary tasks on offline-sampled instruction-route pairs [30]. For REVERIE [64], an additional Object Grounding (OG) [16, 53] is used for object reasoning. During pretraining, we train the model with a batch size of 64 for $100k$ iterations, using Adam [43] optimizer with 1e-4 learning rate. Only one task is adopted at each mini-batch with the same sampling ratio.

Models	R2R							
	val unseen				*test unseen*			
	TL↓	NE↓	SR↑	SPL↑	TL↓	NE↓	SR↑	SPL↑
Seq2Seq [3]	8.39	7.81	22	–	8.13	7.85	20	18
SF [23]	–	6.62	35	–	14.82	6.62	35	28
EnvDrop [74]	10.70	5.22	52	48	11.66	5.23	51	47
AuxRN [103]	–	5.28	55	50	–	5.15	55	51
Active [81]	20.60	4.36	58	40	21.60	4.33	60	41
RecBERT [33]	12.01	3.93	63	57	12.35	4.09	63	57
HAMT [14]	11.46	2.29	66	61	12.27	3.93	65	60
SOAT [60]	12.15	4.28	59	53	12.26	4.49	58	53
SSM [82]	20.7	4.32	62	45	20.4	4.57	61	46
CCC [84]	–	5.20	50	46	–	5.30	51	48
HOP [65]	12.27	3.80	64	57	12.68	3.83	64	59
DUET [16]	13.94	3.31	72	60	14.73	3.65	69	59
LANA [89]	12.0	–	68	62	12.6	–	65	60
TD-STP [100]	–	3.22	70	63	–	3.73	67	61
BSG [55]	14.90	2.89	74	62	14.86	3.19	73	62
BEVBert [1]	14.55	2.81	75	64	–	3.13	73	62
Ours	14.83	2.80	**76**	**65**	15.23	**2.74**	**76**	**66**

Table 1. Quantitative results on R2R [3] (more details in §4.1).

Finetuning. Following the standard protocol [1, 16, 94], we finetune the navigation network using Dagger [69] techniques. In addition, the OG loss [1, 16, 53] is employed on REVERIE. In this stage, we set the learning rate to 1e-5 and batch size to 8 with $20k$ iterations.

Inference. During the testing phase, the agent receives the multi-view images and encodes them as VERs through the frozen environment encoder (§3.1). Based on VERs, the agent performs volume state estimation (§3.2) and models episodic memory (§3.3). By combining both of them, the agent predicts the next step accurately until stops.

Reproducibility. Our model is implemented in PyTorch and trained on eight RTX 4090 GPUs with a 24GB memory per-card. Testing is conducted on the same machine.

4. Experiment

4.1. Performance on VLN

Datasets. The experiments are conducted on three datasets. R2R [3] contains 7,189 trajectories sampled from 90 real-world indoor scenes. It consists of $22k$ human-annotated navigational instructions. The dataset is split into *train*, *val seen*, *val unseen*, and *test unseen* sets, which mainly focus on the generalization capability in unseen environments. REVERIE [64] contains high-level instructions which describe target locations and objects, with a focus on grounding remote target objects. R4R [39] is an extended version of R2R with longer trajectories.

Evaluation Metrics. For R2R, Success Rate (SR), Trajectory Length (TL), Oracle Success Rate (OSR), Success rate weighted by Path Length (SPL), and Navigation Error (NE) are used. For REVERIE, Remote Grounding Success rate

Models	REVERIE																	
	val seen						*val unseen*						*test unseen*					
	TL↓	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑	TL↓	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑	TL↓	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑
RCM [88]	10.70	29.44	23.33	21.82	16.23	15.36	11.98	14.23	9.29	6.97	4.89	3.89	10.60	11.68	7.84	6.67	3.67	3.14
FAST-M [64]	16.35	55.17	50.53	45.50	31.97	29.66	45.28	28.20	14.40	7.19	7.84	4.67	39.05	30.63	19.88	11.61	11.28	6.08
SIA [53]	13.61	65.85	61.91	57.08	45.96	42.65	41.53	44.67	31.53	16.28	22.41	11.56	48.61	44.56	30.80	14.85	19.02	9.20
RecBERT [33]	13.44	53.90	51.79	47.96	38.23	35.61	16.78	35.02	30.67	24.90	18.77	15.27	15.86	32.91	29.61	23.99	16.50	13.51
Airbert [28]	15.16	48.98	47.01	42.34	32.75	30.01	18.71	34.51	27.89	21.88	18.23	14.18	17.91	34.20	30.28	23.61	16.83	13.28
HAMT [14]	12.79	47.65	43.29	40.19	27.20	25.18	14.08	36.84	32.95	30.20	18.92	17.28	13.62	33.41	30.40	26.67	14.88	13.08
HOP [65]	13.80	54.88	53.76	47.19	38.65	33.85	16.46	36.24	31.78	26.11	18.85	15.73	16.38	33.06	30.17	24.34	17.69	14.34
DUET [16]	13.86	73.86	71.75	63.94	57.41	51.14	22.11	51.07	46.98	33.73	32.15	23.03	21.30	56.91	52.51	36.06	31.88	22.06
TD-STP [100]	–	–	–	–	–	–	–	39.48	34.88	27.32	21.16	16.56	–	40.26	35.89	27.51	19.88	15.40
BEVBert [1]	–	76.18	73.72	65.32	57.70	51.73	–	56.40	51.78	36.37	34.71	24.44	–	57.26	52.81	36.41	32.06	22.09
GridMM [94]	–	–	–	–	–	–	23.20	57.48	51.37	36.47	34.57	24.56	19.97	59.55	53.13	36.60	34.87	23.45
LANA [89]	15.91	74.28	71.94	62.77	59.02	50.34	23.18	52.97	48.31	33.86	32.86	22.77	18.83	57.20	51.72	36.45	32.95	22.85
BSG [55]	15.26	78.36	76.18	66.69	61.56	54.02	24.71	58.05	52.12	35.59	35.36	24.24	22.90	62.83	56.45	38.70	33.15	22.34
Ours	16.13	**80.49**	75.83	66.19	**61.71**	**56.20**	23.03	**61.09**	**55.98**	**39.66**	33.71	23.70	24.74	62.22	**56.82**	**38.76**	33.88	23.19

Table 2. Quantitative comparison results on REVERIE [64]. '–': unavailable statistics (see §4.1 for more details).

Models	R4R *val unseen*				
	NE↓	SR↑	CLS↑	nDTW↑	SDTW↑
SF [3]	8.47	24	30	–	–
RCM [88]	–	29	35	30	13
EGP [19]	8.00	30	44	37	18
SSM [82]	8.27	32	53	39	19
RelGraph [32]	7.43	36	41	47	34
RecBERT [33]	6.67	44	51	45	30
HAMT [14]	6.09	45	58	50	32
BSG [55]	6.12	47	59	53	34
LANA [89]	–	43	60	52	32
Ours	6.10	**47**	**61**	**54**	33

Table 3. Quantitative results on R4R [39] (more details in §4.1).

(RGS), and Remote Grounding Success weighted by Path Length (RGSPL) are also employed for object grounding. For R4R, Coverage weighted by Length Score (CLS), normalized Dynamic Time Warping (nDTW), and Success rate weighted nDTW (SDTW) are used.

Performance on R2R. Table 1 compares our model with the state-of-the-art models on R2R. As we find that our model yields SR of **76**% and SPL of **66**% on *text unseen*, which leads to promising gains of **3**% and **4**% over BEVBert [1], respectively. It verifies that using VER to represent the environment leads to better decision-making.

Performance on REVERIE. Table 2 presents the comparison results on REVERIE. Compared with the recent state-of-the-art VLN agent [55], our agent improves SR and SPL by **3.86**% and **4.07**% on the *val unseen* split. This highlights the effectiveness of our architecture design.

Performance on R4R. Table 3 shows results on R4R. Our approach outperforms others in most metrics with a promising gain on nDTW (*i.e.*, **1**%). This suggests the *episodic memory* module is able to retrieve the long-time context.

Visual Results. Fig. 4 depicts one exemplar navigation episode from *val unseen* set of R2R. In this complex environment, there are many rooms with different objects and 3D layout. From the visualization of 3D occupancy prediction at the key steps, we find the geometric details and semantics can be captured well by VER. The room layout estimation can help the agent to understand "enter the bedroom". Finally, our agent finds the "bed" and accomplishes the instruction successfully.

Models	REVERIE			R2R	
	SR↑	SPL↑	RGS↑	SR↑	SPL↑
w/o Volume State	52.31	34.91	32.75	72.71	61.13
w/o Episodic Memory	49.33	33.71	31.36	68.21	61.70
Full model	**55.98**	**39.66**	**33.71**	**75.80**	**65.37**

Table 4. Ablation study of overall design on *val unseen* of REVERIE [64] and R2R [3] (see §4.2 for more details).

4.2. Diagnostic Experiment

To thoroughly test the efficacy of crucial components of our model, we conduct a series of diagnostic studies on *val unseen* split of REVERIE and R2R.

Overall Design. We first investigate the effectiveness of our core design. The results in Table 4 indicate that adding *Volume State* leads to a substantial performance gain (*i.e.*, **3.67**% on SR). After using *Episodic Memory*, a higher score (*i.e.*, 31.36% → **33.71**% on RGS) is achieved.

Neighborhood Range of Viewpoints. We use the neighborhood of each viewpoint in the state space for local action prediction (Eq. 6). Moreover, corresponding pillar representations of the neighborhood are also used for node embeddings of the episodic memory (Eq. 7). From Table 5, the limited range of neighborhood is insufficient to represent the candidate viewpoint for navigation (*e.g.*, **75.80**% → 73.75% of SR on R2R). However, too large neighborhood range will contain irrelevant information, leading to inferior performance (*e.g.*, **55.98**%→53.49% of SR on REVERIE).

4.3. Analysis on 3D Representation Learning

Evaluation Metric. Following standard protocols, we employ Intersection over Union (IoU) to evaluate the occu-

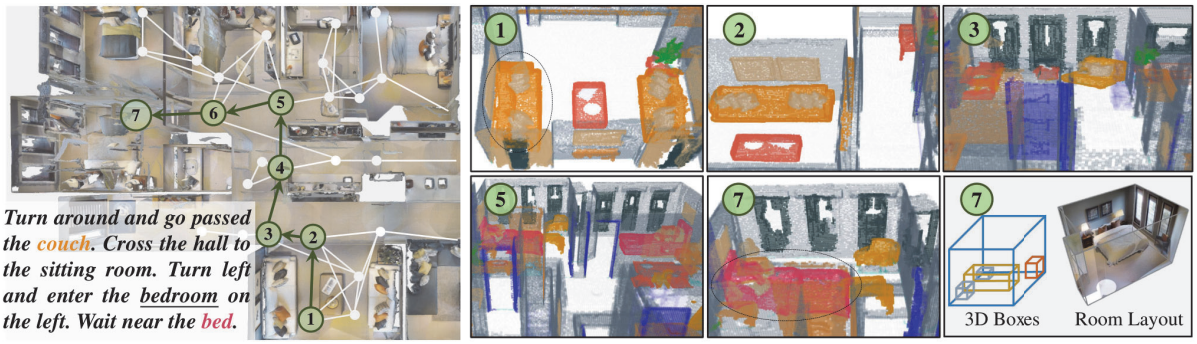

Figure 4. A representative visual result on *val unseen* of R2R [3]. We first visualize the 3D occupancy prediction at the key steps. In addition, we provide the prediction of 3D boxes and 3D room layout at step (7). We find that VER can capture the geometric details of 'couch' and the structure of 'bedroom'. With VER, our agent easily finds the 'bed' and succeeds. See §4.1 for more details.

$\lvert\Omega_n\rvert$	REVERIE			R2R	
	SR↑	SPL↑	RGS↑	SR↑	SPL↑
4	53.30	35.90	**34.16**	73.75	62.77
9	**55.98**	**39.66**	33.71	**75.80**	**65.37**
16	52.18	36.37	33.14	73.82	63.49
25	53.49	35.87	33.46	74.63	63.24

Table 5. Ablation study of neighborhood range on *val unseen* of REVERIE [64] and R2R [3] (see §4.2 for more details).

Models	Occupancy		Detection		Layout
	IoU↑	mIoU↑	mAP↑	mAR↑	3D IoU↑
BEVFormer [51]	20.38	8.97	27.30	43.88	62.71
OccNet [78]	22.12	10.66	29.91	47.15	64.04
Ours	**24.31**	**12.93**	**33.57**	**51.60**	**66.45**

Table 6. Quantitative results on 3D occupancy, 3D detection, and room layout prediction (see §4.3 for more details).

Up-Sampling	3D Perception			R2R	
	mIoU↑	mAP↑	3D IoU↑	SR↑	SPL↑
w/o Coarse-to-Fine	12.39	32.95	**66.57**	–	–
Trilinear Interpolation	11.03	29.42	63.45	75.14	64.30
3D Deconvolution	**12.93**	**33.57**	66.45	**75.80**	**65.37**

Table 7. Ablation study of *Coarse-to-Fine Extraction* on occupancy prediction (mIoU), 3D detection (mAP), room layout (3D IoU), and *val unseen* set of R2R [3] (see §4.3 for more details).

Multi-task Learning			3D Perception			R2R	
Occ.	Obj.	Room.	mIoU↑	mAP↑	3D IoU↑	SR↑	SPL↑
✓			12.09	–	–	74.90	63.82
✓	✓		12.14	32.64	–	75.21	64.79
	✓	✓	–	33.11	64.58	74.03	63.51
✓		✓	11.37	–	63.29	74.97	64.66
✓	✓	✓	**12.93**	**33.57**	**66.45**	**75.80**	**65.37**

Table 8. Ablation study of *Multi-task Learning* on occupancy prediction (mIoU), 3D detection (mAP), room layout estimation (3D IoU), and *val unseen* set of R2R [3] (see §4.3 for more details).

pancy prediction quality, regardless of the semantic labels. The mean IoU (mIoU) of 15 classes is also used to assess the performance of semantic occupancy. For 3D object detection, we utilize mean Average Precision (mAP) and mean Average Recall (mAR) with IoU thresholds of 0.50. For room layout, we adopt 3D IoU for cuboid layout evaluation.

Performance on 3D Tasks. In Table 6, our network (§3.1) outperforms other methods [51, 78] by a significant margin (**2.19**% on IoU of occupancy, **3.66**% on mAP of 3D detection, and **2.41**% on 3D IoU of room layout estimation). The mIoU of occupancy is also exhibits improvement (**2.27**%), underscoring the network's proficiency in capturing both scene geometry and fine-grained semantics.

Coarse-to-Fine Extraction. Table 7 lists the scores with different up-sampling operations (Eq. 2). Our approach improves the performance by solid margins (*e.g.*, 11.03% → **12.93**% for 3D occupancy, 75.14% → **75.80**% on SR of R2R). This verifies the efficacy of our design of the coarse-to-fine extraction and learnable up-sampling operations.

Multi-task Learning. Table 8 reports performance comparison with different perception tasks (§3.1). We find that multi-task learning yields a substantial performance gain. This suggests these 3D perception tasks are complementary to each other in capturing geometric and semantic properties of scenes, further facilitating the decision-making.

5. Conclusion

In this paper, we propose a Volumetric Environment Representation (VER), which aggregates the perspective features into structured 3D cells. Through coarse-to-fine feature extraction, we can efficiently perform several 3D perception tasks. Based on this comprehensive representation, we develop the volume state for local action prediction and the episodic memory for retrieving the global context. We demonstrate that our agent achieves promising performance on VLN benchmarks (R2R, REVERIE, and R4R).

Acknowledgment. This work was supported by the National Natural Science Foundation of China (No. 62372405) , the Fundamental Research Funds for the Central Universities (No. 226-2022-00051), and CCF-Tencent Open Fund.

References

[1] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. In *ICCV*, 2023. 1, 2, 4, 6, 7

[2] Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *arXiv:2304.03047*, 2023. 1, 2

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018. 1, 2, 3, 6, 7, 8

[4] Peter Anderson, Ayush Shrivastava, Devi Parikh, Dhruv Batra, and Stefan Lee. Chasing ghosts: Instruction following as bayesian state tracking. In *NeurIPS*, 2019. 1, 2, 4

[5] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv:2303.12712*, 2023. 2

[6] Vincent Cartillier, Zhile Ren, Neha Jain, Stefan Lee, Irfan Essa, and Dhruv Batra. Semantic mapnet: Building allocentric semantic maps and representations from egocentric views. In *AAAI*, 2021. 2

[7] 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 *3DV*, 2017. 5

[8] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *ICLR*, 2019. 2

[9] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *CVPR*, 2020. 2

[10] Jinyu Chen, Wenguan Wang, Si Liu, Hongsheng Li, and Yi Yang. Omnidirectional information gathering for knowledge transfer-based audio-visual navigation. In *ICCV*, 2023. 1

[11] Kevin Chen, Junshen K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *CVPR*, 2021. 1, 2

[12] Peihao Chen, Dongyu Ji, Kunyang Lin, Runhao Zeng, Thomas Li, Mingkui Tan, and Chuang Gan. Weakly-supervised multi-granularity map learning for vision-and-language navigation. In *NeurIPS*, 2022. 1, 2

[13] Peihao Chen, Xinyu Sun, Hongyan Zhi, Runhao Zeng, Thomas H Li, Gaowen Liu, Mingkui Tan, and Chuang Gan. a^2 nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models. In *NeurIPS Workshop*, 2023. 2

[14] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, 2021. 2, 6, 7

[15] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Learning from unlabeled 3d environments for vision-and-language navigation. In *ECCV*, 2022. 2

[16] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, 2022. 1, 2, 6, 7

[17] Ian Cherabier, Johannes L Schonberger, Martin R Oswald, Marc Pollefeys, and Andreas Geiger. Learning priors for semantic 3d reconstruction. In *ECCV*, 2018. 4

[18] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *ICCV*, 1999. 5

[19] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. In *NeurIPS*, 2020. 1, 2, 7

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 6

[21] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006. 2

[22] Alberto Elfes. Occupancy grids: A stochastic spatial representation for active robot perception. *arXiv:1304.1098*, 2013. 2

[23] 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 *NeurIPS*, 2018. 1, 2, 6

[24] Chen Gao, Xingyu Peng, Mi Yan, He Wang, Lirong Yang, Haibing Ren, Hongsheng Li, and Si Liu. Adaptive zone-aware hierarchical planner for vision-language navigation. In *CVPR*, 2023. 2

[25] Sourav Garg, Niko Sünderhauf, Feras Dayoub, Douglas Morrison, Akansel Cosgun, Gustavo Carneiro, Qi Wu, Tat-Jun Chin, Ian Reid, Stephen Gould, et al. Semantics for robotic mapping, perception and interaction: A survey. *Foundations and Trends® in Robotics*, 8(1–2):1–224, 2020. 2

[26] Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *CVPR*, 2022. 1, 2

[27] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *ACL*, 2022. 2

[28] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *ICCV*, 2021. 2, 7

[29] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *NeurIPS*, 2018. 2

[30] 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 *CVPR*, 2020. 6

[31] Joao F Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *CVPR*, 2018. 2

[32] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. In *NeurIPS*, 2020. 1, 2, 7

[33] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *CVPR*, 2021. 2, 6, 7

[34] Yicong Hong, Yang Zhou, Ruiyi Zhang, Franck Dernoncourt, Trung Bui, Stephen Gould, and Hao Tan. Learning navigational visual representations with semantic map supervision. In *CVPR*, 2023. 1, 2

[35] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *3DV*, 2016. 5

[36] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, 2023. 3

[37] Jingyang Huo, Qiang Sun, Boyan Jiang, Haitao Lin, and Yanwei Fu. Geovln: Learning geometry-enhanced visual representation with slot attention for vision-and-language navigation. In *CVPR*, 2023. 2

[38] 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. In *ICPR*, 2022. 1, 2

[39] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. In *ACL*, 2019. 2, 6, 7

[40] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *CVPR*, 2020. 3

[41] Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. In *CVPR*, 2023. 2

[42] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2, 6

[43] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[44] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2016. 1

[45] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. In *ICCV*, 2021. 2

[46] Obin Kwon, Jeongho Park, and Songhwai Oh. Renderable neural radiance map for visual navigation. In *CVPR*, 2023. 2

[47] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Hanming Deng, et al. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *IEEE TPAMI*, 2023. 2, 3

[48] Jialu Li and Mohit Bansal. Panogen: Text-conditioned panoramic environment generation for vision-and-language navigation. In *NeurIPS*, 2023. 2

[49] Jialu Li, Hao Tan, and Mohit Bansal. Envedit: Environment editing for vision-and-language navigation. In *CVPR*, 2022. 2

[50] Yunzhu Li, Shuang Li, Vincent Sitzmann, Pulkit Agrawal, and Antonio Torralba. 3d neural scene representations for visuomotor control. In *CoRL*, 2022. 2

[51] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 2, 3, 4, 8

[52] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 4, 6

[53] Xiangru Lin, Guanbin Li, and Yizhou Yu. Scene-intuitive agent for remote embodied visual grounding. In *CVPR*, 2021. 6, 7

[54] Chong Liu, Fengda Zhu, Xiaojun Chang, Xiaodan Liang, Zongyuan Ge, and Yi-Dong Shen. Vision-language navigation with random environmental mixup. In *ICCV*, 2021. 2

[55] Rui Liu, Xiaohan Wang, Wenguan Wang, and Yi Yang. Bird's-eye-view scene graph for vision-language navigation. In *ICCV*, 2023. 1, 2, 6, 7

[56] Shice Liu, Yu Hu, Yiming Zeng, Qiankun Tang, Beibei Jin, Yinhe Han, and Xiaowei Li. See and think: Disentangling semantic scene completion. In *NeurIPS*, 2018. 4

[57] Yu Lu, Ruijie Quan, Linchao Zhu, and Yi Yang. Zero-shot video grounding with pseudo query lookup and verification. *IEEE TIP*, 33:1643–1654, 2024. 2

[58] 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 *ECCV*, 2020. 2

[59] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[60] Abhinav Moudgil, Arjun Majumdar, Harsh Agrawal, Stefan Lee, and Dhruv Batra. Soat: A scene-and object-aware transformer for vision-and-language navigation. In *NeurIPS*, 2021. 6

[61] Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. In *ICLR*, 2018. 1

[62] Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. In *ICCV*, 2021. 1

[63] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 3

[64] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, 2020. 1, 2, 6, 7, 8

[65] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. Hop: history-and-order aware pretraining for vision-and-language navigation. In *CVPR*, 2022. 2, 6, 7

[66] Ruijie Quan, Linchao Zhu, Yu Wu, and Yi Yang. Holistic lstm for pedestrian trajectory prediction. *IEEE TIP*, 30: 3229–3239, 2021. 2

[67] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 4

[68] Luis Roldao, Raoul De Charette, and Anne Verroust-Blondet. 3d semantic scene completion: A survey. *IJCV*, 130(8):1978–2005, 2022. 2, 3, 4

[69] Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011. 6

[70] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 2, 3, 4

[71] Yucheng Suo, Fan Ma, Linchao Zhu, and Yi Yang. Knowledge-enhanced dual-stream zero-shot composed image retrieval. In *CVPR*, 2024. 2

[72] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014. 1, 2

[73] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 2, 6

[74] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL*, 2019. 2, 6

[75] Sinan Tan, Mengmeng Ge, Di Guo, Huaping Liu, and Fuchun Sun. Self-supervised 3d semantic representation learning for vision-and-language navigation. *arXiv:2201.10788*, 2022. 1, 2

[76] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002. 2, 4

[77] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *arXiv:2304.14365*, 2023. 2, 4, 5

[78] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *ICCV*, 2023. 3, 8

[79] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2

[80] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *CVPR*, 2020. 2

[81] Hanqing Wang, Wenguan Wang, Tianmin Shu, Wei Liang, and Jianbing Shen. Active visual information gathering for vision-language navigation. In *ECCV*, 2020. 6

[82] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *CVPR*, 2021. 1, 2, 6, 7

[83] Hanqing Wang, Wei Liang, Luc V Gool, and Wenguan Wang. Towards versatile embodied navigation. In *NeurIPS*, 2022. 2

[84] Hanqing Wang, Wei Liang, Jianbing Shen, Luc Van Gool, and Wenguan Wang. Counterfactual cycle-consistent learning for instruction following and generation in vision-language navigation. In *CVPR*, 2022. 2, 6

[85] Hanqing Wang, Wei Liang, Luc Van Gool, and Wenguan Wang. Dreamwalker: Mental planning for continuous vision-language navigation. In *ICCV*, 2023. 2

[86] Hanqing Wang, Wenguan Wang, Wei Liang, Steven CH Hoi, Jianbing Shen, and Luc Van Gool. Active perception for visual-language navigation. *IJCV*, 131(3):607–625, 2023. 2

[87] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, Xihui Liu, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *CVPR*, 2024. 3

[88] 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 *CVPR*, 2019. 2, 7

[89] Xiaohan Wang, Wenguan Wang, Jiayi Shao, and Yi Yang. Lana: A language-capable navigator for instruction following and generation. In *CVPR*, 2023. 6, 7

[90] Xiaohan Wang, Wenguan Wang, Jiayi Shao, and Yi Yang. Learning to follow and generate instructions for language-capable navigation. *IEEE TPAMI*, 2023. 2

[91] Yue Wang, Alireza Fathi, Abhijit Kundu, David A Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *ECCV*, 2020. 5

[92] Yuqi Wang, Yuntao Chen, Xingyu Liao, Lue Fan, and Zhaoxiang Zhang. Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. In *CVPR*, 2024. 4

[93] Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *ICCV*, 2023. 2

[94] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *ICCV*, 2023. 1, 2, 6, 7

[95] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, 2023. 4, 5

[96] Aming Wu, Rui Liu, Yahong Han, Linchao Zhu, and Yi Yang. Vector-decomposed disentanglement for domain-invariant object detection. In *ICCV*, 2021. 2

[97] Yiming Wu, Ruixiang Li, Zequn Qin, Xinhai Zhao, and Xi Li. Heightformer: Explicit height modeling without extra data for camera-only 3d object detection in bird's eye view. *arXiv:2307.13510*, 2023. 4

[98] Zhenjia Xu, Zhanpeng He, Jiajun Wu, and Shuran Song. Learning 3d dynamic scene representations for robot manipulation. In *CoRL*, 2020. 3, 5, 6

[99] Jiahui Zhang, Hao Zhao, Anbang Yao, Yurong Chen, Li Zhang, and Hongen Liao. Efficient semantic scene completion network with spatial group convolution. In *ECCV*, 2018. 4

[100] Yusheng Zhao, Jinyu Chen, Chen Gao, Wenguan Wang, Lirong Yang, Haibing Ren, Huaxia Xia, and Si Liu. Target-driven structured transformer planner for vision-language navigation. In *ACM MM*, 2022. 2, 6, 7

[101] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *AAAI*, 2024. 2

[102] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv:1904.07850*, 2019. 5

[103] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *CVPR*, 2020. 6

[104] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 4

[105] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *CVPR*, 2018. 5

[106] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. Manhattan room layout reconstruction from a single 360 image: A comparative study of state-of-the-art methods. *IJCV*, 129:1410–1431, 2021. 2, 3, 5