Bird's-Eye-View Scene Graph for Vision-Language Navigation

Rui Liu Xiaohan Wang Wenguan Wang* Yi Yang
ReLER, CCAI, Zhejiang University
https://github.com/DefaultRui/BEV-Scene-Graph

Abstract

Vision-language navigation (VLN), which entails an agent to navigate 3D environments following human instructions, has shown great advances. However, current agents are built upon panoramic observations, which hinders their ability to perceive 3D scene geometry and easily leads to ambiguous selection of panoramic view. To address these limitations, we present a BEV Scene Graph (BSG), which leverages multi-step BEV representations to encode scene layouts and geometric cues of indoor environment under the supervision of 3D detection. During navigation, BSG builds a local BEV representation at each step and maintains a BEV-based global scene map, which stores and organizes all the online collected local BEV representations according to their topological relations. Based on BSG, the agent predicts a local BEV grid-level decision score and a global graph-level decision score, combined with a subview selection score on panoramic views, for more accurate action prediction. Our approach significantly outperforms state-of-the-art methods on REVERIE, R2R, and R4R, showing the potential of BEV perception in VLN.

1. Introduction

Vision-language navigation (VLN) task [1] requires an agent to navigate through a 3D environment [2] to a target location, according to natural language instructions. Existing work has made great advances in cross-modal reasoning [3–8], path planning [9–13], and auxiliary tasks for pretraining [14–18]. Their core ideas are learning to relate the language instructions to panoramic images of the environment. Though straightforward, these approaches heavily rely on 2D panoramic observations. As a result, they lack the capacity to preserve scene layouts and 3D structure, which are critical for navigation decision-making in embodied scenes. Moreover, indoor environments [2, 19–21] are characterized by substantial occlusion [22–24], posing challenges for the agent to accurately identify the objects and landmarks referenced by the instructions [1, 25].

*Corresponding author: Wenguan Wang.

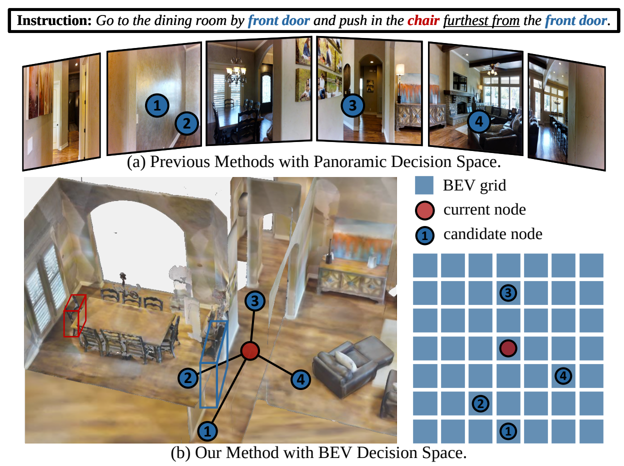


Figure 1: For panoramic view (a), two candidate nodes (①&②) correspond to the same image leading to ambiguity. For Bird's-Eye-View (b), they are represented by discriminative grids (■).

For example (Fig. 1(a)), given the instruction "*Go to the dining room by front door and push in the chair furthest from the front door*", previous approaches [3–5, 14, 15, 17, 26–30] formulate VLN as a sequential text-to-image grounding problem by matching navigable candidate nodes with adjacent panoramic views. At each time step, given a set of subviews captured from different directions, the agent selects a navigable direction as the next step for navigation. However, this strategy tends to introduce ambiguity, when the agent needs to discriminate between multiple candidate nodes corresponding to the same subview. In addition, the agent struggles to ground the associated objects and explore their spatial relation in 3D scene, such as identifying "*the chair furthest from the front door*". Consequently, relying solely on panoramic view presents difficulties in both comprehensive scene perception and efficient navigation.

To address the challenges encountered by panoramic methods, Bird's-Eye-View (BEV) perception emerges as a viable solution, employing discriminative grid representations to model the 3D environment. Meanwhile, BEV grid representation effectively captures spatial context and scene layouts [31, 32], facilitating both perception [33–36] and planning [37–40]. Building upon these insights, we present a BEV Scene Graph (BSG), which harnesses the power of

BEV representation to construct an informative navigation graph. During navigation, the agent collects local BEV representations at each navigable node. A global scene graph is established by connecting these BEV representations topologically. At each step, the agent makes an informed decision by predicting a BEV grid-level decision score and a BSG graph-level decision score, combined with a subview selection score on panoramic views [13, 28, 41].

Specifically, the agent acquires multi-view observations at each step and performs view transformation [35, 42–44] on the corresponding image features. Later, a 3D detection head [44–46] is employed on these BEV representations to predict oriented bounding boxes, encoding object-level geometric and semantic information. During navigation, the node embeddings of BSG are represented by neighboring BEV grids. Then they are updated by querying the overlap region between BEV representations from different steps.

Previous semantic maps in robot navigation, including occupancy grids [47–50] and learnable spatio-semantic representations [51–55], have only provided top-down information without crucial 3D object information. Differently, BSG leverages the BEV representations to achieve consistency between 3D perception and decision-making while encoding geometric context. Our approach is evaluated on three benchmarks (*i.e.*, REVERIE [25], R2R [1], R4R [56]). For the referring expression comprehension in REVERIE, BSG outperforms the state-of-the-art method [28] by $\mathbf{5.14}\%$ and $\mathbf{3.21}\%$ in SR and RGS on the val unseen split, respectively. BSG also achieves $\mathbf{4}\%$ and $\mathbf{3}\%$ improvement in SR and SPL on the test split of R2R, respectively. The impressive results shed light on the promises of BEV perception in VLN task.

2. Related Work

Vision-Language Navigation (VLN). VLN task [1] has drawn significant attention in embodied AI domain. Early work typically adopts recurrent neural networks with cross-modal attention [1, 3, 5, 57, 58]. Later, various techniques have been developed to improve VLN, including: **i)** using more powerful vision-and-language embedding methods based on pre-trained transformer models [14, 15, 17, 18, 28–30, 59–61]; **ii)** exploiting more supervisory information from environment augmentation [62–64], instruction generation [3, 5, 65–67], and other auxiliary tasks [4, 9, 16, 68, 69]; **iii)** designing more efficient action planning and learning strategies by incorporating self-correction [11, 57], global action space [12, 26, 41, 70, 71], map building [13, 28, 41, 55], knowledge prompts [8, 72, 73], or ensemble of IL [1] and RL [4, 74]; and **iv)** developing more large-scale benchmarks [2, 25, 70, 75–81] and platforms [2, 79–81].

However, existing work heavily relies on panoramic subviews for navigation, suffering from the limitations of 2D perspective view. These limitations, including occlusion and a narrow field of subview, introduce ambiguity in action prediction, thereby hindering efficient navigation. In contrast, we leverage BEV representations to facilitate navigation decision-making through view transformation. These BEV representations encode geometric context of environment under the supervision of BEV-based 3D detection.

Map Representation for Navigation. To achieve accurate navigation, it is critical to develop an efficient representation of surrounding environments. In robot navigation, classical SLAM-based approaches build a map based on geometry and plan the path on this semantic-agnostic map [48, 50, 82, 83]. These approaches are built upon sensors and thus highly susceptible to measurement noises [49, 54]. To explore semantic information, learnable semantic map [10, 47–52, 54, 84] is proposed using the learnable spatial representations from a top-down view. These two types of metric maps focus on dense representations with explicit location information of environment. Moreover, topological maps [12, 13, 28, 41] are developed to model the relationship among sparse nodes in the environment, mitigating the burden of heavy computation. In addition, some efforts build topo-metric maps to combine the advantages of metric and topological maps [55, 85–87].

Existing map-based methods neglect the role of 3D perception for navigation. In contrast, BSG encodes scene layouts and geometric cues by 3D detection for comprehensive scene understanding, eventually facilitating path planning.

Perceptual Organization of 3D Scenes. Scene representation should provide information about both object semantics and layout composition [24, 88–93]. For indoor scene understanding, visual representation can take various forms, including an RGB image and depth map [19–21], voxel grids [94], and point clouds [95, 96]. As pointed by [24], structural representation [6, 97, 98] also plays a significant role, as it models the spatial relationships among different objects. Therefore, modeling visual and structural properties is critical for scene understanding. Recently, BEV feature provides a unified representation for perception and motion planning [31, 32, 37–40].

Motivated by the recent efforts that achieve learnable projection between BEV plane and perspective view [33–35, 42, 43, 99–103], we collect oriented 3D bounding boxes in Matterport3D dataset [2] and perform camera-based BEV perception for embodied amodal detection [22, 104], as opposed to previous point cloud-based detection [105–107]. Under the supervision of 3D detection, we employ BEV feature to establish scene representations that effectively capture object-level geometry information for navigation.

3. Approach

Task Setup. We illustrate our approach using R2R [1] task, where the environment is discretized as a set of navigable nodes and navigability edges. The agent observes the sur-

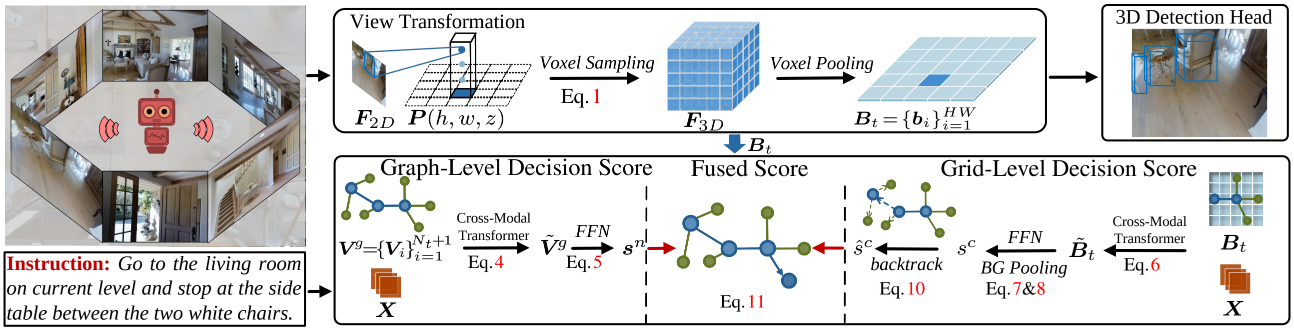


Figure 2: Overview of BSG. View transformation is first employed to project the multi-view images into BEV plane (§3.1). Then, BEV feature is encoded using 3D detection (§3.3). Through the integration of BEV representations during navigation, we predict a graph-level decision score on BSG and a grid-level decision score based on BEV. These scores are fused to facilitate effective decision-making (§3.2).

roundings at each node and finds a route to the target location, specified by the instruction $\mathcal{X}=\{x_l\}_{l=1}^{L}$ with L words.

Panoramic Methods. Previous VLN agents [4, 5, 28, 30] are built as panoramic view selectors [3] where navigable candidate nodes are represented by adjacent observations from different viewing angles. However, the adjacency rule in panoramic navigation will cause multiple candidate nodes to correspond to the same panoramic view, thus introducing ambiguity in action prediction (Fig. 1(a)). In addition, the geometric cues of 3D environment cannot be captured by visual features of 2D panoramic views, such as occluded objects [22, 104, 108, 109] and scene layouts [39, 110].

Our Idea. To overcome the above limitations, we utilize BEV features as geometry-enhanced visual representations, supervised by BEV-based 3D detection. Then, we construct BEV Scene Graph (BSG) online using BEV features (Fig. 1(b)). With BSG, the agent effectively predicts the next step on candidate nodes, which are represented by discriminate BEV grids. Before detailing BEV detection (§3.3), we first introduce how to build BSG (§3.1) and how to predict decision score for action prediction(§3.2).

3.1. BSG Construction

During navigation, the agent collects local BEV representations of surrounding environment online, and constructs a global scene graph gradually. Specifically, at time step t, BSG is denoted as $\mathcal{G}_t=\{\mathcal{V}_t,\mathcal{E}_t\}$, where each node $v\in\mathcal{V}_t$ incorporates observed information (Fig. 3), corresponding to each navigable location in the environment.

View Transformation. At current location v^*, the agent acquires multi-view camera images[1]. We perform *voxel sampling* [31, 32, 35, 44, 111] on each image feature $\boldsymbol{F}_{2D}\in\mathbb{R}^{H_cW_c\times D}$ to construct 3D voxel feature $\boldsymbol{F}_{3D}\in\mathbb{R}^{HWZ\times D}$, where H_cW_c and HW are the spatial dimensions of image feature and BEV plane, respectively. Predefined 3D reference points $\boldsymbol{P}\in\mathbb{R}^{HWZ}$ are used to query the image feature via *cross-attention* for voxel feature (Fig. 2), where HWZ denotes the number of reference points:

$$\boldsymbol{F}_{3D}(h,w,z)=\text{CrossAtt}\big(\boldsymbol{P}(h,w,z),\boldsymbol{F}_{2D}(h_i,w_i)\big). \tag{1}$$

Then, $\boldsymbol{F}_{3D}$ is squeezed down to BEV space by *voxel pooling* as $\boldsymbol{B}=\{\boldsymbol{b}_i\}_{i=1}^{HW}\in\mathbb{R}^{HW\times D}$, where each grid cell contains a D-sized latent vector, representing the corresponding region in environment. Then, BEV feature is connected with a 3D detection head (*cf.* §3.3) to predict bounding boxes, providing the agent with object-level geometry information.

Node Representation from BEV Grids. At the start of navigation (*i.e.*, $t=0$), BSG $\mathcal{G}_0$ is initialized with the node set $\mathcal{V}_0$ and its associated BEV feature $\boldsymbol{B}_0$ (Fig. 3). It is noted that there is an overlapping region Ω^{o} between $\boldsymbol{B}_t$ and $\boldsymbol{B}_{t+1}$, since the perception range is greater than the moving step. At time step $t+1$, the same spatial region will be captured by different BEV grid features from Ω^{o}. Then, we execute temporal modeling on $\boldsymbol{B}_t$ and $\boldsymbol{B}_{t+1}$ to integrate history information, thereby facilitating the representation of stationary objects [35, 112, 113]. In particular, we adopt *cross-attention* [114] on the grid features to update $\boldsymbol{B}_{t+1}$:

$$\tilde{\boldsymbol{b}}_{j,t+1}=\text{CrossAtt}\big(\boldsymbol{b}_{i,t},\boldsymbol{b}_{j,t+1}\big),\ i,j\in\Omega^{\text{o}}. \tag{2}$$

Since local scene information is captured by corresponding BEV features, we construct node representations of BSG by incorporating the features of surrounding BEV grids, which are identified by nearest neighbor search [115, 116]. At step $t+1$, for current node v^* and its navigable candidate nodes $\{v_k^+\}_{k=1}^{K_{t+1}}\in\mathcal{V}_{t+1}$, we *average* the BEV grid features of corresponding neighborhood Ω_t^{n}:

$$\boldsymbol{V}_{t+1}=\text{Ave}\big(\{\boldsymbol{b}_{i,t+1}\}_{i\in\Omega_t^{\text{n}}}\big). \tag{3}$$

Each node representation $\boldsymbol{V}_{t+1}\in\mathbb{R}^{D}$ attends to a certain area. For the candidate nodes that have been observed (or visited) multiple times, we *average* the previous representations as its node embedding [28, 41]. After updating BSG, we preserve $\boldsymbol{B}_{t+1}$ for subsequent action prediction (§3.2).

[1]As there are no specific camera parameters available for panoramic images from the simulator [1], we utilize the images captured by raw camera with intrinsic and extrinsic parameters [2]. Both types of images encompass identical visual content (see §9.1 for details).

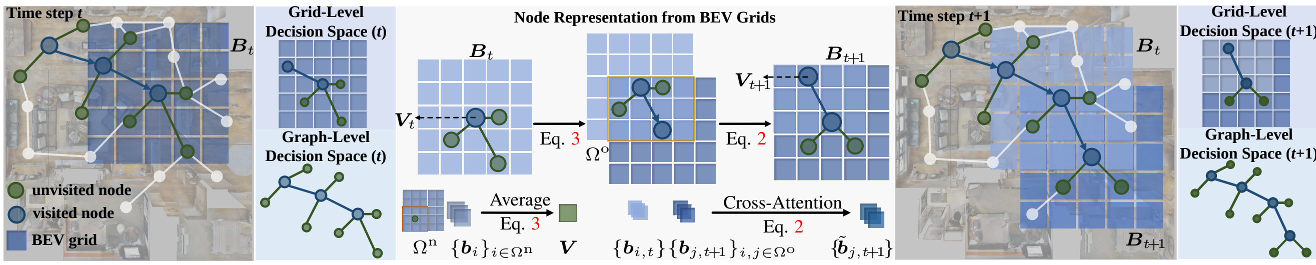


Figure 3: The node embeddings of BSG are represented by BEV grids in their neighborhood. From step t to t+1, BSG is updated using temporal modeling (§3.1). Both global graph-level and local grid-level decision space are also used for accurate action prediction (§3.2).

3.2. BEV-based Navigation Action Prediction

With the current BSG $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t\}$ and navigation instruction $\mathcal{X}$, the agent predicts next step by combining grid-level decision score on BEV feature $\boldsymbol{B}_t$ and graph-level decision score on BSG $\mathcal{G}_t$. Following [12, 28], we add a hallucination "stop" node to existing N_t $(=|\mathcal{V}_t|)$ nodes.

BSG-based Graph-level Decision Score. The word embeddings $\boldsymbol{X} \in \mathbb{R}^{L \times D}$ and node embeddings $\boldsymbol{V}^g = \{\boldsymbol{V}_n\}_{n=1}^{N_t+1} \in \mathbb{R}^{(N_t+1) \times D}$ are fed into a *cross-modal* encoder [117] with several *cross-attention* and *self-attention* layers to model the relations between instruction and graph representations:

$$\tilde{\boldsymbol{V}}^g = \{\tilde{\boldsymbol{V}}_n\}_{n=1}^{N_t+1} = \text{CrossMod}\big([\boldsymbol{V}^g, \boldsymbol{X}]\big), \tag{4}$$

where $[\cdot]$ indicates the concatenation operation. After that, we adopt a feed-forward network (*FFN*) to predict the global graph-level decision score $s^n \in \mathbb{R}^{N_t+1}$ of $\mathcal{G}_t$:

$$\boldsymbol{s}^g = \{s_n^g\}_{n=1}^{N_t+1} = \text{FFN}\big(\tilde{\boldsymbol{V}}^g\big). \tag{5}$$

BEV-based Grid-Level Decision Score. Grid-level decision score on $\boldsymbol{B}_t$ is crucial for the agent to understand the 3D scene and learn effective navigation policies. A similar *cross-modal* transformer [117] is used to mine fine-grained visual clues and object-related textual information from the instructions, such as "*front door*" and "*the chair furthest from the front door*":

$$\tilde{\boldsymbol{B}}_t = \{\tilde{\boldsymbol{b}}_i\}_{i=1}^{HW} = \text{CrossMod}\big([\boldsymbol{B}_t, \boldsymbol{X}]\big). \tag{6}$$

Then the instruction-aware representations $\tilde{\boldsymbol{B}}_t$ is used to predict local grid-level decision score $s^l \in \mathbb{R}^{HW}$ by *FFN*:

$$\boldsymbol{s}^l = \{s_i^l\}_{i=1}^{HW} = \text{FFN}\big(\tilde{\boldsymbol{B}}_t\big). \tag{7}$$

We propose a distance-dependent weighted pooling to convert the grid-level score s^l to local candidate score $s^c \in \mathbb{R}^{K_t+1}$ (containing the stop node) [1, 25]. For k-th navigable candidate node, the score is calculated as follows:

$$s_k^c = \sum\nolimits_{i \in \Omega_k^{\text{n}}} W_{k,i} s_i^l, \tag{8}$$

where Ω_k^{n} is the grid neighborhood of k-th candidate node (*cf.* Eq. 3), and $\boldsymbol{W}_k = [W_{k,i}]_{i=1}^{|\Omega_k^{\text{n}}|}$ is a truncated *Bivariate Gaussian* weight, as the contribution of BEV grids to candidate nodes is considered contingent on relative distance:

$$W_{k,i} = \hat{g}(\Delta x_{k,i}, \Delta y_{k,i}), \tag{9}$$

where $(\Delta x_{k,i}, \Delta y_{k,i})$ is the relative coordinates of the i-th BEV grid center to k-th candidate node coordinates, $\hat{g}(\cdot)$ is normalized Bivariate Gaussian probability $\mathcal{N}(\boldsymbol{\mu}_{x,y}, \boldsymbol{\sigma}_{x,y})$, $\boldsymbol{\mu}_{x,y}$ is the mean vector, and $\boldsymbol{\sigma}_{x,y}$ is the covariance matrix.

Fused Action Prediction. To fuse the global graph-level decision score and local grid-level decision score, a backtracking strategy [12, 28] is adopted to convert the local score $s^c \in \mathbb{R}^{K_t+1}$ into global space $\hat{s}^c \in \mathbb{R}^{N_t+1}$. Specifically, when navigating to the nodes that are not connected to the current node, we assume the agent needs to backtrack through the visited candidate nodes as:

$$\hat{s}^c = \begin{cases} s_{\text{back}}, & \text{if backtrack}, \\ s^c, & \text{otherwise}. \end{cases} \tag{10}$$

More specifically, we compute a backtracking score for unconnected nodes in $\mathcal{V}_t$ by summing the decision scores of visited candidate nodes as s_{back}. Then, a weight W_f is employed to fuse the local and global decision scores:

$$s_n = W_f \hat{s}_n^c + (1 - W_f) s_n^g. \tag{11}$$

Using the fused prediction, BSG can complement existing works [12, 28, 41] with global action space. We will adopt a previous method [28] as basic agent for experiment (*cf.* §4).

3.3. BEV Representation Encoding

BEV detection endows the agent with awareness of object-level geometry information, facilitating more accurate action prediction [118, 119]. In this section, we learn 3D object detection on the top of BEV feature (see §3.1) [33–35]. Accordingly, the details on collecting a Matterport3D-based detection dataset for embodied amodal perception [22, 23, 120], called Matterport3D^2, will be presented. We also introduce the details of detection head.

Multi-view Image Acquisition. To enable an agent to perceive the surroundings through camera, we build a new 3D detection dataset Matterport3D^2 on multi-view images captured by camera [2], which differs from the previous whole-scene detection [19–21, 110] based on point clouds [95, 96].

During navigation, the agent revolves around the direction of gravity to capture the RGB images in 90 building-scale scenes. The original dataset [2] provides information on the object center and segments throughout the entire scene.

Amodal Perception for Embodied Agent. Apart from recognizing the semantics and shapes for visible part of the object, the ability to perceive the whole of an occluded object (*i.e.*, amodal perception) [22, 104, 108, 109] is also significant for navigation. Since occlusion frequently occurs in the indoor scenes, embodied amodal perception aids the agent in comprehending the persistence of scene layouts that objects possess extents and continue to exist even when they are occluded. We consider the occlusion relationship between objects on center visibility criterion, *i.e.*, an object is considered to be visible if its center is not occluded. To determine the visibility of objects in each image from multi-views, we project the object center onto the multi-view image planes and ascertain whether it is located within the camera frustum [21, 121]. Specifically, we establish the transformation from 3D world coordinates to pixel coordinates in the image using the intrinsic and extrinsic parameters of the camera. Then, we obtain a group of corresponding objects for the multi-view images (more details are shown in Appendix).

3D Oriented Bounding Box Generation. We spatially register all objects into an egocentric coordinate system at each panoramic viewpoint. To annotate the objects, we utilize a custom algorithm (*cf.* Appendix) which automatically generates 3D oriented bounding boxes (OBB) for 17 categories of indoor objects, as opposed to the axis-aligned bounding box (AABB) annotations with a fixed yaw angle of zero in the original dataset [2]. OBB surrounds the outline of the objects more tightly than AABB, resulting in more accurate route planning for VLN (see Table 10). In addition, the amodal detection on Matterport3D[2] follows the same train/val/test splits as previous VLN tasks [1, 25].

Bipartite Matching for BEV Detection. We construct the 3D detection head [33–35] upon BEV features $\boldsymbol{B}$ on Matterport3D[2]. A bipartite matching loss [44–46, 122] is used to establish a correspondence between the groundtruth and box prediction, which consists of a focal loss [123] for class labels and a $L1$ loss for bounding box regression. We evaluate different BEV methods [34, 35, 42, 44] for indoor detection (see Table 8). Note that BSG is not constrained to any specific BEV model, allowing for seamless integration of advanced BEV frameworks for VLN.

3.4. Implementation Details

For ease of training, we employ a separate training strategy of the BEV detection and navigation policy networks, as the initial perception module cannot offer a correct feedback (or rewards) to the navigation policy [22, 23]. Therefore, BSG utilizes BEV features encoded by BEVFormer [35]. Following recent VLN practice [14, 15, 17, 29], pretraining and finetuning paradigm is adopted on a basic model [28] equipped with BSG. In this section, we will mainly introduce the details of BSG branch and present the detailed results in Table 4 (see Appendix for more details).

Voxel Sampling. For view transformation, we introduce the *voxel sampling* here (Eq. 1). The default size of BEV queries is 11×11 with four reference points (*i.e.*, $Z=4$) for each query, and the perception ranges are $[-5.0\ m, 5.0\ m]$ for x and y axes. Considering the practical height of camera and rooms in [2], the predefined height anchors are uniformly sampled from $[-1.0\ m, 2.0\ m]$ for z axis. The number of neighboring grids for node embedding is 9 (Eq. 3).

BSG Architecture. Following the recent transformer-based methods [14, 17, 18, 28–30], the pretrained LXMERT [117] is utilized for initialization. We use 9, 2, and 4 transformer layers in the text encoder and cross-modal encoder (Eq. 4&6), respectively. We keep the other parameters consistent with prior works [28, 117]. During the finetuning process, the similar structure variants in the cross-modal encoder are adopted as previous studies [17, 28]. The fused weight W_f in Eq. 11 is set to 0.5. For the Bivariate Gaussian weight (Eq. 9), $\boldsymbol{\mu}_{x,y}$ is the zero vector, and $\boldsymbol{\sigma}_{x,y}$ is the diagonal matrix with diagonal elements of 2. We set the weight of 0.7 for OCM and 0.3 for [28].

Pretraining. For the R2R [1] and R4R [56], we adopt the *Masked Language Modeling* (MLM) [60, 114] and *Single-step Action Prediction* (SAP) [17, 30] as auxiliary tasks in the pretraining stage. For REVERIE [25], an additional *Object Grounding* (OG) [28, 124] is used for object reasoning. During the pretraining stage, we train the model with a batch size of 32 for 100k iterations, using Adam optimizer [125] with 1e-4 learning rate. Four RTX 3090 GPUs are used for network training, and only one pretraining task is adopted at each mini-batch with the same sampling ratio.

Finetuning. Following standard protocol [17, 28], we finetune the pretrained network with a mixture of *teacher-forcing* [126] and *student-forcing* on different VLN datasets. On REVERIE, the OG loss [28, 124] is also employed for finetuning, and a predefined weight 0.20 is adopted to balance navigation and object grounding. Moreover, we set the learning rate to 1e-5 and batch size to 8 with 25k iterations.

Inference. Once trained, the agent is capable of route planning while considering object context and scene layouts (§3.3). During the testing phase, we update BSG online (§3.1) and predict a fused action score (§3.2). The agent is forced to stop if it exceeds the maximum action steps [1].

4. Experiment

We first provide the results on VLN benchmarks (§4.1). To verify efficacy of core model designs, we conduct a set of diagnostic studies (§4.2). For comprehensive analysis, we investigate the impact of BEV perception on VLN (§4.3).

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 [4]	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-MATTN [25]	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 [124]	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 [30]	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 [29]	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 [17]	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 [18]	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
TD-STP [71]	–	–	–	–	–	–	–	39.48	34.88	27.32	21.16	16.56	–	40.26	35.89	27.51	19.88	15.40
DUET [28]	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
LANA [67]	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
Ours	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**

Table 1: Quantitative comparison results on REVERIE [25]. '–': unavailable statistics. See §4.1 for more details.

4.1. Performance on VLN

Datasets. The experiments are conducted on three datasets. REVERIE [25] contains high-level instructions describing target locations and objects, with a focus on grounding remote target objects. R2R [1] contains 7,189 shortest-path trajectories, each associated with three step-by-step instructions. The dataset is split into *train*, *val seen*, *val unseen*, and *test unseen* sets with 61, 56, 11, and 18 scenes, respectively. R4R [56] is an extended variant of R2R by concatenating two adjacent trajectories with longer instructions.

Evaluation Metric. Following the standard setting [1, 3, 17] of VLN task, we use five metrics for evaluation, *i.e.*, Success Rate (SR), Trajectory Length (TL), Oracle Success Rate (OSR), Success rate weighted by Path Length (SPL), and Navigation Error (NE). Two additional evaluation metrics, Remote Grounding Success rate (RGS) and Remote Grounding Success weighted by Path Length (RGSPL), are used for REVERIE [25, 28, 30]. For R4R [17, 41, 56], Coverage weighted by Length Score (CLS), normalized Dynamic Time Warping (nDTW), and Success rate weighted nDTW (SDTW) are adopted (more details in Appendix).

Performance on REVERIE [25]. Table 1 compares our model with the recent state-of-the-art VLN models on REVERIE dataset. We find that our model outperforms previous approaches across all the evaluation metrics on the three splits. Notably, on the *val unseen* split, our model outperforms the previous best model DUET [28] by $\mathbf{5.14}\%$ on SR, $\mathbf{1.86}\%$ on SPL and $\mathbf{3.21}\%$ on RGS. On the more challenging *test unseen* split, we improve over the baseline by $\mathbf{3.94}\%$ on SR, $\mathbf{2.64}\%$ on SPL, and $\mathbf{1.27}\%$ on RGS. This demonstrates the effectiveness of our architecture design.

Performance on R2R [1]. Table 2 presents the comparison results on R2R dataset. We can find that our approach sets new state-of-the-arts for most metrics. For instance, on *val unseen*, our model yields SR and SPL of $\mathbf{74}$ and $\mathbf{62}$, respectively, while those for the baseline method [28] are 72 and 60. Our approach improves the performance of DUET by solid margins on *test unseen* (*i.e.*, 69→$\mathbf{73}$ for SR, 59→$\mathbf{62}$ for SPL). In addition, it also shows significant performance gains in terms of NE (*i.e.*, 3.65→$\mathbf{3.19}$).

Models	R2R							
	val unseen				*test unseen*			
	TL↓	NE↓	SR↑	SPL↑	TL↓	NE↓	SR↑	SPL↑
Seq2Seq [1]	8.39	7.81	22	–	8.13	7.85	20	18
SF [3]	–	6.62	35	–	14.82	6.62	35	28
EnvDrop [5]	10.70	5.22	52	48	11.66	5.23	51	47
AuxRN [16]	–	5.28	55	50	–	5.15	55	51
PREVALENT [15]	10.19	4.71	58	53	10.51	5.30	54	51
RelGraph [6]	9.99	4.73	57	53	10.29	4.75	55	52
Active Perception [26]	20.60	4.36	58	40	21.60	4.33	60	41
RecBERT [30]	12.01	3.93	63	57	12.35	4.09	63	57
HAMT [17]	11.46	2.29	66	61	12.27	3.93	65	60
SOAT [27]	12.15	4.28	59	53	12.26	4.49	58	53
EGP [12]	–	4.83	56	44	–	5.34	53	42
GBE [70]	–	5.20	54	43	–	5.18	53	43
SSM [41]	20.7	4.32	62	45	20.4	4.57	61	46
CCC [66]	–	5.20	50	46	–	5.30	51	48
HOP [18]	12.27	3.80	64	57	12.68	3.83	64	59
LANA [67]	12.0	–	68	62	12.6	–	65	60
TD-STP [71]	–	3.22	70	63	–	3.73	67	61
DUET [28]	13.94	3.31	72	60	14.73	3.65	69	59
Ours	14.90	**2.89**	**74**	**62**	14.86	**3.19**	**73**	**62**

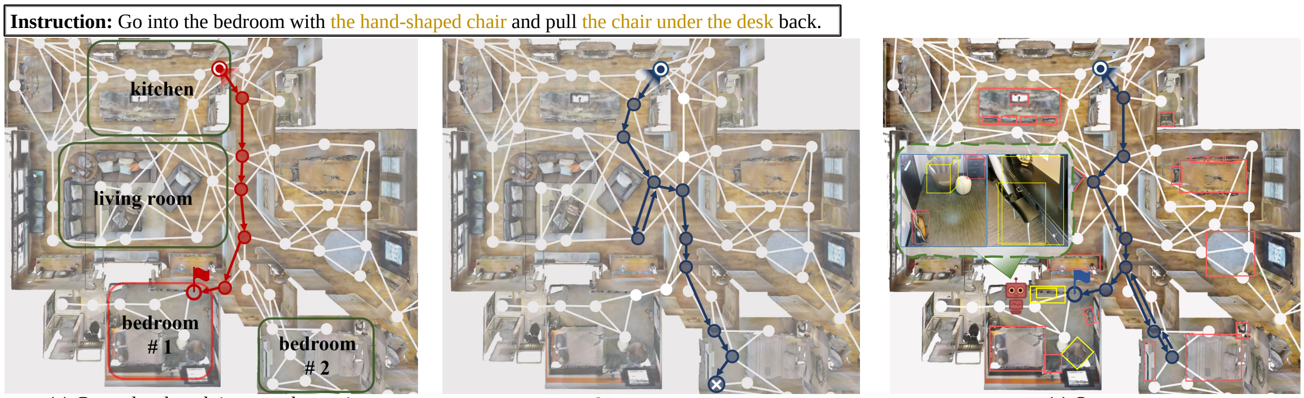
Table 2: Quantitative results on R2R [1] (more details in §4.1).

Models	R4R *val unseen*				
	NE↓	SR↑	CLS↑	nDTW↑	SDTW↑
SF [3]	8.47	24	30	–	–
RCM [4]	–	29	35	30	13
EGP [12]	8.00	30	44	37	18
SSM [41]	8.27	32	53	39	19
RelGraph [6]	7.43	36	41	47	34
RecBERT [30]	6.67	44	51	45	30
HAMT [17]	6.09	45	58	50	32
LANA [67]	–	43	60	52	32
Ours	6.12	**47**	59	**53**	**34**

Table 3: Quantitative results on R4R [56] (more details in §4.1).

Performance on R4R [56]. Table 3 shows results on R4R dataset. Our approach outperforms others in most metrics and leads to a promising gain on SR (*i.e.*, 45→$\mathbf{47}$).

Visual Results. As shown in Fig. 4, "bedroom" is a critical landmark for instruction execution. There are two



(a) Groundtruth path in a top-down view (b) Basic agent (c) Our agent

Figure 4: A representative visual result on REVERIE dataset [25] (§4.1). There are two bedrooms and it is difficult to distinguish between them. The basic agent in (b) steps into the bedroom #2 and ends in failure. With BSG, our agent in (c) returns back to the correction direction and succeeds according to the object context and scene layouts.

#	Models	REVERIE			R2R	
		SR↑	SPL↑	RGS↑	SR↑	SPL↑
1	Basic agent [28]	46.98	33.73	32.15	71.52	60.36
2	BEV Branch	39.03	25.73	25.09	65.56	52.21
3	***w/o.*** detection	49.25	32.44	33.21	72.65	60.20
4	Full model	**52.12**	**35.59**	**35.36**	**73.73**	**62.33**

Table 4: Ablation study of overall design on *val unseen* of REVERIE [25] and R2R [1]. See §4.2 for more details.

#	$\lvert\Omega^n\rvert$	REVERIE			R2R	
		SR↑	SPL↑	RGS↑	SR↑	SPL↑
1	4	51.33	34.34	34.86	72.89	62.07
2	9	**52.12**	**35.59**	**35.36**	**73.73**	**62.33**
3	16	51.71	34.11	34.54	73.26	61.99

Table 5: Ablation study of node embeddings on *val unseen* of REVERIE [25] and R2R [1]. See §4.2 for more details.

Updating	REVERIE			R2R	
	SR↑	SPL↑	RGS↑	SR↑	SPL↑
w/o. BEV updating	50.30	34.05	35.05	72.29	60.77
w. BEV updating	**52.12**	**35.59**	**35.36**	**73.73**	**62.33**

Table 6: Ablation study of *BEV updating* on *val unseen* of REVERIE [25] and R2R [1]. See §4.2 for more details.

#	Models	Decision Space		REVERIE			R2R	
		Graph	Grid	SR↑	SPL↑	RGS↑	SR↑	SPL↑
1	Basic agent [28]	–	–	46.98	33.73	32.15	71.52	60.36
2	Variants	✓		50.18	33.94	33.66	73.02	60.76
3			✓	48.25	34.34	34.02	72.79	61.54
4*			✓	51.27	34.56	35.20	73.10	61.88
5	Full model	✓	✓	**52.12**	**35.59**	**35.36**	**73.73**	**62.33**

Table 7: Ablation study of fused decision-making on *val unseen* of REVERIE [25] and R2R [1]. '$*$' denotes using uniform weight instead of *Bivariate Gaussian* (Eq. 8). More details in §4.2.

bedrooms in the environment, which have different objects and geometric context (Fig. 4(a)). However, the basic agent [28] navigates a wrong bedroom #2 and finally fails (Fig. 4(b)). In Fig. 4(c), the BSG enables our agent to perceive the object-aware 3D information, finding "the chair under the desk" and "hand-shaped chair" to accomplish the task (more visual results are shown in Appendix).

4.2. Diagnostic Experiment

To assess the efficacy of essential components of BSG, detailed ablation studies are conducted and the results of *val unseen* split of REVERIE [25] and R2R [1] are shown.

Overall Design. We first investigate the effectiveness of our overall design. The results presented in row #1, #2, and #4 of Table 4 indicate that adding BEV branch leads to a promising gain over the basic agent [28] across all metrics. From row #3 and #4, we improve the model by using additional detection loss $\mathbf{2.87}\%$ on SR of REVERIE, $\mathbf{3.15}\%$ on RGS of REVERIE, and $\mathbf{2.13}\%$ on SPL of R2R.

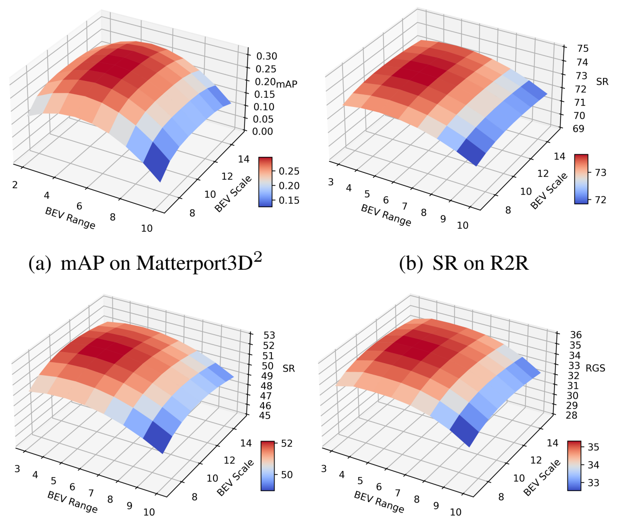
Neighborhood for Node Embeddings. We next validate the design of node embeddings. For each navigable candidate node, we employ its neighboring grid representations to construct the node embeddings (*cf.* Eq. 3). In Table 5, it can be observed that insufficient neighboring grids, as seen in rows #1 and #2, cannot represent the node well for navigation. On the other hand, from row #2 and #3, selecting too many neighboring grids can impact the discriminability of node embeddings due to a large number of overlap between candidate neighborhoods (see Fig. 3).

BEV Updating Strategy. At each step, we update BEV features by *cross-attention*, and then use the modified BEV grids to revise node embeddings (*cf.* Eq. 2). In Table 6, the variant of model that does not include *BEV updating* leads to inferior performance compared to full model.

Fused Decision-Making. The results in row #1, #2, and #3 of Table 7 suggest that both graph and grid-level decision space of BSG facilitate the navigation (*cf.* §3.2). From

BEV Models	Matterport3D^2		REVERIE			R2R	
	mAP↑	mAR↑	SR↑	SPL↑	RGS↑	SR↑	SPL↑
LSS [42]	0.188	0.270	50.83	34.43	33.19	72.25	61.30
BEVDepth [34]	0.252	0.443	51.06	34.35	34.13	72.77	61.38
BEVFormer [35]	**0.299**	**0.488**	**52.12**	**35.59**	**35.36**	**73.73**	**62.33**

Table 8: Ablation study of different BEV models on *val unseen* of REVERIE [25] and R2R [1]. See §4.3 for more details.



(a) mAP on Matterport3D^2 (b) SR on R2R

(c) SR on REVERIE (d) RGS on REVERIE

Figure 5: Ablation study of BEV scale and perception range on *val unseen* of REVERIE [25] and R2R [1] (more details in §4.3).

row #4, using *Bivariate Gaussian* weights results in better performance compared to assigning uniform weights, as it takes into account the varying contribution of each BEV grid to the node based on the relative distances.

4.3. Analysis on BEV Encoding

In this section, we present the detection results on *val unseen* of Matterport3D^2. For evaluation, we utilize mean Average Precision (mAP) and mean Average Recall (mAR), with Intersection over Union (IoU) thresholds of 0.50, following standard protocols [19, 20, 127, 128]. Then, we provide a quantitative analysis of how BEV detection affects VLN performance, including different types of BEV models (*depth prediction* [34, 42] and *voxel sampling* [35]) and the ablation study on the superior model [35].

Different BEV Models. We first compare several representative open-source BEV models [34, 35, 42], which are divided into two aspects based on different view transformations. BEVFormer [35] utilizes voxel sampling to encode 2D features to 3D space (*cf.* Eq. 1), while LSS [42] and BEVDepth [34] employ 2D features to predict depth information and then lift these features to 3D space. Note that BEVDepth [34] requires explicit depth information as additional supervision. As listed in Table 8, BEVFormer [35] outperforms all other methods with $\mathbf{0.299}$ mAP and $\mathbf{0.488}$ mAR. We adopt BEVFormer [35] as our BEV baseline. Moreover, our performance can be further improved with more advanced BEV models.

#	Z	Matterport3D^2		REVERIE			R2R	
		mAP↑	mAR↑	SR↑	SPL↑	RGS↑	SR↑	SPL↑
1	2	0.260	0.443	51.49	35.07	**36.27**	72.81	60.50
2	4	**0.299**	**0.488**	**52.12**	**35.59**	35.36	**73.73**	**62.33**
3	8	0.266	0.438	50.98	33.56	35.34	72.30	60.44

Table 9: Ablation study of reference points on *val unseen* of REVERIE [25] and R2R [1]. See §4.3 for more details.

Annotation	Matterport3D^2		REVERIE			R2R	
	mAP↑	mAR↑	SR↑	SPL↑	RGS↑	SR↑	SPL↑
AABB	0.266*	0.491*	49.25	32.44	34.14	73	60
OBB	0.299	0.488	**52.12**	**35.59**	**35.36**	**74**	**62**

Table 10: Ablation study of OBB and AABB on *val unseen* of REVERIE [25] and R2R [1]. '$*$' denotes the detection performance on AABB annotations. See §4.3 for more details.

BEV Scale and Perception Range. We next delve into the core parameters of our BEV, *i.e.*, scale and perception range (*cf.* Eq. 1). The results are summarized in Fig. 5. We find that different scales and perception ranges will affect detection accuracy (*cf.* Eq. 1). Since node representations are associated with BEV features (*cf.* §3.1), better detection performance can bring more gain to navigation. The selection of an appropriate perception range should take into account both dimensions and structure of indoor environment.

Reference Points. Table 9 presents a comprehensive analysis of the number of reference points proposed in §3.1. Reference points enable the sampling of multi-view features and their integration into BEV feature (*cf.* §3.1).

OBB *vs* AABB for Perception and Navigation. The oriented bounding box (OBB) is more commonly used in 3D perception of real-world scenarios, such as collision detection [129, 130] and grasp detection [131–133], compared to the axis-aligned box (AABB). In Table 10, using the OBB, the agent's perception performance is better as it provides accurate orientation and scale information (*cf.* §3.3), resulting in the improved performance in all navigation tasks.

5. Conclusion

Scene understanding is crucial for intelligent navigation in 3D environments. However, current VLN agents rely solely on panoramic observations, lacking the capacity to preserve 3D layouts and geometric cues, and hence limiting their planning ability. In this paper, we propose a BEV scene graph (BSG) for 3D perception-based VLN, that enables the agent to perceive the scene and access the object layouts. By fusing BSG-based action score and BEV grid-level action score, our approach achieves promising results. This highlights the great potential of BEV perception in VLN.

Bird's-Eye-View Scene Graph for Vision-Language Navigation
Supplementary Material

This document provides more details of our approach and additional experimental results, which are organized as follows:

- *Model details (§6)*
- *Experimental setups (§7)*
- *Additional results and visualization (§8)*
- *Additional analysis of Matterport3D[2] (§9)*
- *Discussion (§10)*

6. Model Details

In our model, BEV Scene Graph (BSG) is proposed to enable discriminative decision space (*c.f.* §3.2) based on BEV feature. However, to align with the discrete environments present in the VLN simulator [1, 25], it is necessary to convert the action space into nodes (Fig. 6). Consequently, BSG can serve as a valuable complement to existing works [12, 28, 41] that focus on panoramic decision space (*c.f.* §6.2). Specifically, our approach incorporates a panoramic branch [28] . We will give more details on how to train this combined model in §6.4.

6.1. Different Decision Space

Low-level Decision Space. The early research [1] employed a low-level visuomotor control, which constrained the action space to six actions corresponding to left, right, up, down, forward, and stop. Specifically, the forward action means the agent need to move to the closest reachable viewpoint. The left, right, up and down actions are defined to move the camera by 30 degrees. Nonetheless, such a visuomotor control posed challenges for the agent to follow instructions accurately and required the agent to memorize extensive sequential inputs.

Panoramic Decision Space. To enable high-level action reasoning, panoramic decision space [3] involves discretizing panoramic view of the surrounding environment into 36 view angles (12 headings $\times$ 3 elevations with 30 degree intervals). At each location, the agent is limited to a few navigable directions that correspond to these panoramic views. Most existing works [1, 3, 9, 17, 28, 30] adopt this decision space. However, due to the adjacency rule (Fig.1(a)), multiple candidate nodes may correspond to the same panoramic view, resulting in ambiguity during route planning.

BEV Grid Decision Space. To address the aforementioned constraints, we introduce a grid-level decision space from

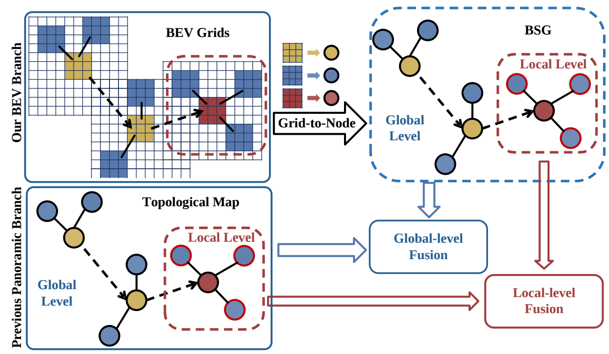


Figure 6: Integrating our framework with previous approaches.

bird's eye view. Each candidate node corresponds to specific BEV grids (Fig. 1(b)). The node embedding is represented by its neighboring grid features (Fig. 6).

6.2. Complementary to Existing Methods

As shown in Fig. 6, our method predicts the next step action by fusing both global and fine-scale local decision-making strategies (see §3.2). Specifically, for the topological level, our model predicts the global score on all the navigable nodes, including previously visited and observed nodes, which are similar to previous works [12, 28, 41, 55]. Meanwhile, for the local level, the local score are for all navigable nodes of the current node, but our model first predicts the BEV grid-level score in the local level then converts to the score of navigable nodes to making a more accurate prediction. Thus, our model can be easily combined with existing work based on panoramic features as shown in Fig. 6. In this paper, we explore the complementary nature of our model with a recent state-of-the-art method [28], which also predicts the global and local score at each step.

6.3. Detailed Network Architecture on REVERIE

Object Prediction. For REVERIE [25], an agent is required to identify an object at each step where additional candidate object annotations are provided. To enable fine-grained perception, we incorporate an object prediction module into local branch. Specifically, we adopt the ViT-B/16 pretrained on ImageNet to extract the features of M objects at t-th step $O_t = \{o_m | o_m \in \mathbb{R}^{768}\}_{m=1}^{M}$, and add orientation feature [9, 28] with $\sin$ and $\cos$ values for heading and elevation angles. Then these object features are concatenated with BEV features as visual features, and we adopt a cross-modal transformer on visual and textual features to obtain contextual representations. Finally, grid-level decision score and object score are predicted by FFN.

6.4. Pretraining Objectives

For R2R [1] and R4R [56], we adopt Masked Language Modeling (MLM) [60, 114], Masked Region Classification (MRC) [14, 15, 134, 135], and Single-step Action Prediction with Progress Monitoring (SAP-PM) [9, 17, 30, 57] as auxiliary tasks in the pretraining stage. For REVERIE [25], an additional Object Grounding (OG) [28, 124] are used for object reasoning and grounding, and the sample ratio is MLM:MRC:SAP-PM:OG=1:1:1:1. All the auxiliary tasks are based on the input pair $(\mathcal{X}, \mathcal{G}_t, \mathcal{T}_t)$, where $\mathcal{X}$ is the textual embedding, $\mathcal{G}_t$ is BSG built at time step t, and $\mathcal{T}_t$ is topological map of complementary method [28] with panoramic visual feature V_t (*c.f.* §6.2).

MLM. The task aims to learn grounded language representations in VLN task and cross-modal alignment. It masks some percentage of the input tokens at random, and then predicts those masked tokens based on contextual words and [60]. We randomly mask out one of the word tokens in $\mathcal{X}$ with the probability of 15% [17, 28], and the final hidden representations corresponding to the [*mask*] token are fed into an output softmax over the instruction vocabulary:

$$\mathcal{L}_{\mathrm{MLM}} = -\log p(x_i|\mathcal{X}_{\setminus i}, \mathcal{G}_t, \mathcal{T}_t), \tag{12}$$

where x_i is the textual embedding of the masked token, $\mathcal{X}_{\setminus i}$ is the masked instruction. We average output embedding of two textual encoders of panoramic branch and BEV branch, and minimize the negative log-likelihood of original words.

MRC. This task predicts the semantic labels of masked observation features given instructions and neighboring observations [17]. We only use this task for panoramic branch, and keep other settings consistent with [17, 28].

SAP-PM. We employs imitation learning to predict the next action [15, 17, 28]. Specifically, we sample a map-action pair $(\mathcal{G}_t, \mathcal{T}_t, \mathcal{A}_t)$ from the groundtruth trajectory at the t-th step, and then the loss of panoramic branch is as follows:

$$\mathcal{L}_{\mathrm{SAP}} = \sum\nolimits_{t=1}^{T} -\log p(a_t|\mathcal{X}, \mathcal{T}_t). \tag{13}$$

For our BEV branch, we employ an additional progress monitoring task [9, 57] to reflect the navigation progress:

$$\mathcal{L}_{\mathrm{SAP-PM}} = \sum\nolimits_{t=1}^{T} -\log p(a_t|\mathcal{X}, \mathcal{G}_t) + (y_t^{pm} - p_t^{pm})^2, \tag{14}$$

where y_t^{pm} is the normalized distance of length from the current location to the goal as in Eq.(12). We use a weight of 0.5 to balance $\mathcal{L}_{\mathrm{SAP}}$ and $\mathcal{L}_{\mathrm{SAP-PM}}$.

OG. The goal of this task is to predict the best matching object among a set of candidate objects at the current viewpoint [28, 124]. The loss is as follows:

$$\mathcal{L}_{\mathrm{OG}} = -\log p(o_i|\mathcal{X}, \mathcal{G}_t, \mathcal{T}_t), \tag{15}$$

where o_i is the groundtruth object, and we average the matching score of panoramic branch and BEV branch.

6.5. Finetuning Objectives

Since reinforcement learning reward makes the agent pay more attention on shortest paths rather than path fidelity with instruction [17], we alternatively use Teacher-Forcing (TF) and Student-Forcing (SF) for action prediction as behavior cloning (BC):

$$\begin{aligned} \mathcal{L}_{\mathrm{TF}} &= \sum\nolimits_{t=1}^{T} -\log p(a_t|\mathcal{X}, \mathcal{G}_t, \mathcal{T}_t), \\ \mathcal{L}_{\mathrm{SF}} &= \sum\nolimits_{t=1}^{T} -\log p(a_t^*|\mathcal{X}, \mathcal{G}_t^*, \mathcal{T}_t^*), \end{aligned} \tag{16}$$

where $\mathcal{G}_t$ and $\mathcal{T}_t$ are maps built online following the expert trajectory, $\mathcal{G}_t^*$ and $\mathcal{T}_t^*$ are following the sampling trajectory, and a_t^* is supervised by the pseudo interactive demonstrator in [28, 136]. On REVERIE, the OG loss is also employed for finetuning, and we adopt a predefined weight $\alpha = 0.20$ to balance them:

$$\mathcal{L} = \alpha\mathcal{L}_{\mathrm{TF}} + \mathcal{L}_{\mathrm{SF}} + \mathcal{L}_{\mathrm{OG}}. \tag{17}$$

7. Experimental Setups

7.1. Evaluation Metrics

VLN. Following the standard setting [1, 3, 17] of R2R, there are several metrics for evaluation: (1) Success Rate (SR) considers the percentage of final positions less than 3 m away from the goal location. (2) Trajectory Length (TL) measures the total length of agent trajectories. (3) Oracle Success Rate (OSR) is the success rate if the agent can stop at the closest point to the goal along its trajectory. (4) Success rate weighted by Path Length (SPL) is a trade-off between SR and TL. (5) Navigation Error (NE) refers to the shortest distance between agent's final position and the goal location. For REVERIE [25, 28, 30], there are two additional metrics. (6) Remote Grounding Success rate (RGS) is the success rate of finding the target object. (7) Remote Grounding Success weighted by Path Length (RGSPL) uses the ratio between the length of the ground-truth path and the agent's path to normalize RGS. For R4R [17, 41, 56], three metrics are used for instruction fidelity. (8) Coverage weighted by Length Score (CLS) is the product of the path coverage and length score of the agent's path with respect to reference path. (9) Normalized Dynamic Time Warping (nDTW) and (10) Success rate weighted normalized Dynamic Time Warping (SDTW) measure the order consistency of agent trajectories.

7.2. Training Details

VLN. During the pretraining stage, we train the combined model with a batch size of 32 for 100k iterations. We then finetune the model with the batch size of 8 for 25k iterations. On REVERIE [25], we select the best epoch by SPL on *val unseen*. On R2R and R4R [1, 56], the best model is selected according to the sum of SR and SPL on *val unseen*. For fair comparison, the same synthesize instructions in [28] by a speaker model [3] are also used for REVERIE.

3D Detection. For BEVFormer [35], a static model without using history BEV features is used for 3D detection. We adopt ViT-B/16 [137] pretrained on ImageNet as the backbone. The size of the image features are $1280 \times 1024 \times 768$, and we don't utilize the multi-scale features in previous work [33–35]. We train this BEV encoder with detection head [35, 44] using AdamW with a weight decay of 0.01 for 500 epoches, a learning rate of 1×10^{-4}.

For LSS [42] and BEVDepth [34], we use ResNet-50 as the image backbone and the image size is processed to 256×704. We don't adopt image or BEV data augmentations. AdamW is used as an optimizer with a learning rate set to 2×10^{-4} and batch size set to 48. All experiments are trained for 24 epochs.

8. Additional Results and Visualization

VLN. To compare the differences between the two datasets, we also show an example with the same groundtruth path but different instructions in Fig. 7. It shows that detailed instructions in R2R provide additional information that enables a more accurate navigation strategy.

3D Detection. Table 12 present the detection results on *test unseen* in Matterport3D^2. For evaluation, we utilize Average Precision (AP) and Average Recall (AR) with Intersection over Union (IoU) thresholds of 0.25 and 0.50, following established protocols [19, 20, 127, 128]. We find that it has good detection performance on larger objects, such as "bed" and 'sofa' with 0.535 and 0.394 for AP in Table 12. However, detecting small objects like 'picture' and 'plant' presents more difficulty since they are almost flat. The detection performance on Matterport3D^2 can be further improved in the future.

9. Additional Analysis of Matterport3D^2

9.1. Detailed Annotation Process

Images of Skybox from Simulator. For each panorama in original Matterport3D [2], the acquisition equipment rotates around the direction of gravity to six distinct orientations, stopping at each to acquire three 1280×1024 photos from three RGB cameras pointing up, horizontal, and down, respectively. Consequently, each panorama view contains 6×3 raw images. In the VLN task, most previous works [3, 9, 25, 28, 30] use the split "skybox" images [2] for panoramic viewing. These "skybox" images are generated by stitching the raw 6×3 images. Then, Matterport3D Simulator [1, 3] in the VLN task splits the skybox-based panoramic view into 12×3 images with the pre-defined size of 640×480 (*c.f.* §3.2). However, this approach does not produce an explicit view transformation matrix.

Raw Camera Images. In order to use accurate camera internal and external parameters for projection in 3D detection[2], we acquire the six raw color images at each viewpoint from the horizontal view for Matterport3D^2 dataset. Multi-view perspective images captured by camera can access to the original transformation matrix. Given the camera parameters, the resolution of raw camera image is also fixed. Thus we have to use 1280×1024 resolution (see §3.1). Specifically, we use the undistorted color images and undistorted camera parameters.

Oriented Bounding Boxes. Although original dataset [2] provides the axis-aligned bounding boxes, they do not provide accurate annotations for 3D detection. Thus, to conform with standard protocols [128, 138], we annotate the oriented bounding boxes (OBB) under LiDAR coordinate system [33, 35][3], which surrounding the outline of the objects more tightly than the axis-aligned bounding boxes. We apply Principal Component Analysis (PCA) to the x and y coordinates of segments in each object, as each object consists of many annotated segments.

9.2. Detailed Dataset Statistics

In Table 11, we present the detailed statistics of our Matterport3D^2 dataset. At each viewpoint, there are six multi-view images (*c.f.* §9.1). However, since we need to filter the objects at each viewpoint, we only collect the multi-view images of viewpoints that have objects. We use the same *train seen*, *val unseen*, and *test unseen* splits as existing VLN datasets [1, 25].

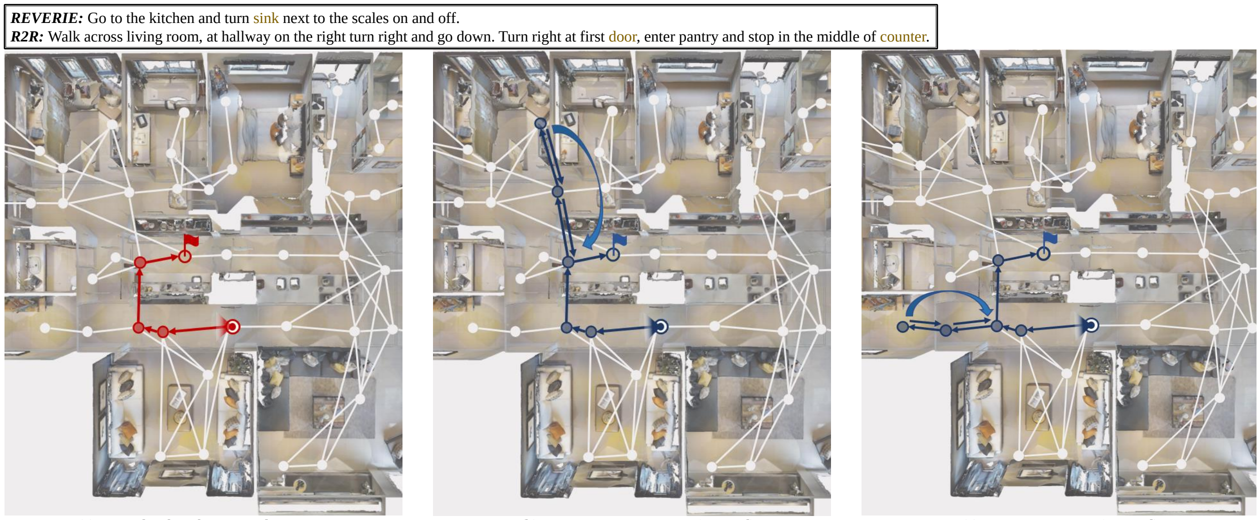
10. Discussion

Asset License and Consent. In this study, we explore vision-language navigation using famous datasets, i.e. Matterport3D [2], R2R [1], and REVERIE [25], that are all publicly available for academic purposes. All the code is released under the MIT license. We implement all models on the MMDetection3D codebase. MMDetection3D codebase (https://github.com/open-mmlab/mmdetection3d) is released under Apache 2.0 license.

Broader Impact. Our work introduces BEV feature for VLN with BSG. Our approach not only achieves a promising improvement of model performance, but also enhances the decision-making by providing grid-level decision score. Furthermore, Matterport3D^2 dataset, which includes oriented bounding boxes for indoor 3D detection, will contribute to future research in the community. It should be noted that our navigation agents are developed and evaluated in virtual simulated environments. Since we primarily trained the model in a static environment where all objects are relatively stationary, deploying the algorithm on a

[2] https://github.com/niessner/Matterport/blob/master/data_organization.md

[3] https://mmdetection3d.readthedocs.io/en/latest/tutorials/coord_sys_tutorial.html



(a) Groundtruth path in a top-down view (b) Our agent on REVERIE (succeed) (c) Our agent on R2R (succeed)

Figure 7: Visual results with the same groundtruth path on REVERIE and R2R dataset.

Split	viewpoints	chair	door	table	picture	cabinet	cushion	window	sofa	bed	chest	plant	sink	toilet	monitor	lighting	shelving	appliances	overall
train seen	3463	14665	18394	5511	8493	3632	5534	13918	1056	1100	2215	1875	1831	605	1745	8171	2629	847	92221
val unseen	439	1634	2456	863	1388	726	1491	1501	176	97	211	223	179	48	72	762	380	107	12314
test unseen	829	2388	4105	1009	2492	1223	1411	2365	289	285	277	1063	601	228	323	1469	547	357	20432

Table 11: Statistics of Matterport3D^2 dataset.

Classes	AP_{25}	AR_{25}	AP_{50}	AR_{50}
cabinet	0.522	0.676	0.348	0.551
door	0.451	0.649	0.279	0.516
picture	0.152	0.334	0.053	0.186
cushion	0.489	0.659	0.281	0.505
window	0.413	0.570	0.251	0.434
shelving	0.501	0.629	0.320	0.501
sofa	0.663	0.765	0.394	0.581
lighting	0.257	0.486	0.103	0.308
plant	0.587	0.729	0.352	0.566
sink	0.486	0.654	0.265	0.486
table	0.487	0.668	0.306	0.525
bed	0.691	0.740	0.535	0.649
toilet	0.529	0.645	0.306	0.456
chair	0.542	0.695	0.374	0.579
appliances	0.504	0.613	0.346	0.507
chest	0.447	0.607	0.247	0.448
monitor	0.413	0.570	0.264	0.446
Overall	0.478	0.629	0.295	0.485

Table 12: Results on Matterport3D^2 *test unseen*.

real-world robot may result in collisions with moving objects and cause harm to individuals. Therefore, further research and development should be conducted to ensure safe deployment in real-world scenarios, such as adding more speed sensors to avoid collisions and including additional environments to study potential damage risks.

References

[1] 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, 4, 5, 6, 7, 8, 9, 10, 11

[2] 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. 1, 2, 3, 4, 5, 11

[3] 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, 3, 6, 9, 10, 11

[4] 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, 3, 6

[5] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL*, 2019. 1, 2, 3, 6

[6] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. In *NeurIPS*, 2020. 2, 6

[7] Wanrong Zhu, Yuankai Qi, Pradyumna Narayana, Kazoo Sone, Sugato Basu, Xin Wang, Qi Wu, Miguel P. Eckstein, and William Yang Wang. Diagnosing vision-and-language navigation: What really matters. In *NAACL*, 2022.

[8] Bingqian Lin, Yi Zhu, Zicong Chen, Xiwen Liang, Jianzhuang Liu, and Xiaodan Liang. Adapt: Vision-language navigation with modality-aligned action prompts.

In *CVPR*, 2022. 1, 2

[9] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *ICLR*, 2019. 1, 2, 9, 10, 11

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

[11] 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 *CVPR*, 2019. 2

[12] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. In *NeurIPS*, 2020. 2, 4, 6, 9

[13] 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

[14] 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. 1, 2, 5, 10

[15] 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. 1, 2, 5, 6, 10

[16] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *CVPR*, 2020. 2, 6

[17] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, 2021. 1, 2, 5, 6, 9, 10

[18] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. Hop: history-and-order aware pre-training for vision-and-language navigation. In *CVPR*, 2022. 1, 2, 5, 6

[19] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. 1, 2, 4, 8, 11

[20] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 8, 11

[21] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *NeurIPS*, 2021. 1, 2, 4, 5

[22] Jianwei Yang, Zhile Ren, Mingze Xu, Xinlei Chen, David J Crandall, Devi Parikh, and Dhruv Batra. Embodied amodal recognition: Learning to move to perceive objects. In *ICCV*, 2019. 1, 2, 3, 4, 5

[23] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied question answering in photorealistic environments with point cloud perception. In *CVPR*, 2019. 4, 5

[24] Akshay Gadi Patil, Supriya Gadi Patil, Manyi Li, Matthew Fisher, Manolis Savva, and Hao Zhang. Advances in data-driven analysis and synthesis of 3d indoor scenes. *arXiv preprint arXiv:2304.03188*, 2023. 1, 2

[25] 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, 4, 5, 6, 7, 8, 9, 10, 11

[26] Hanqing Wang, Wenguan Wang, Tianmin Shu, Wei Liang, and Jianbing Shen. Active visual information gathering for vision-language navigation. In *ECCV*, 2020. 1, 2, 6

[27] 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

[28] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, 2022. 2, 3, 4, 5, 6, 7, 9, 10, 11

[29] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *ICCV*, 2021. 5, 6

[30] 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. 1, 2, 3, 5, 6, 9, 10, 11

[31] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Enze Xie, Zhiqi Li, Hanming Deng, Hao Tian, et al. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *arXiv preprint arXiv:2209.05324*, 2022. 1, 2, 3

[32] Yuexin Ma, Tai Wang, Xuyang Bai, Huitong Yang, Yuenan Hou, Yaming Wang, Yu Qiao, Ruigang Yang, Dinesh Manocha, and Xinge Zhu. Vision-centric bev perception: A survey. *arXiv preprint arXiv:2208.02797*, 2022. 1, 2, 3

[33] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 2, 4, 5, 11

[34] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *AAAI*, 2023. 5, 8, 11

[35] 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, 5, 8, 11

[36] Linyan Huang, Huijie Wang, Jia Zeng, Shengchuan Zhang, Liujuan Cao, Rongrong Ji, Junchi Yan, and Hongyang Li. Geometric-aware pretraining for vision-centric 3d object detection. *arXiv preprint arXiv:2304.03105*, 2023. 1

[37] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu,

and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *ICCV*, 2021. 1, 2

[38] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022.

[39] Haimei Zhao, Jing Zhang, Sen Zhang, and Dacheng Tao. Jperceiver: Joint perception network for depth, pose and layout estimation in driving scenes. In *ECCV*, 2022. 3

[40] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Goal-oriented autonomous driving. In *CVPR*, 2023. 1, 2

[41] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *CVPR*, 2021. 2, 3, 4, 6, 9, 10

[42] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 2, 5, 8, 11

[43] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021. 2

[44] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, 2022. 2, 3, 5, 11

[45] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *CVPR*, 2016.

[46] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 5

[47] Alberto Elfes. Occupancy grids: A stochastic spatial representation for active robot perception. In *Proceedings of the Sixth Conference on Uncertainty in AI*, 1990. 2

[48] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *ICLR*, 2019. 2

[49] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *ICLR*, 2020. 2

[50] Santhosh K Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman. An exploration of embodied visual exploration. *IJCV*, 129:1616–1649, 2021. 2

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

[52] 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

[53] 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.

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

[55] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Topo-metric map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022. 2, 9

[56] 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, 5, 6, 10

[57] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *CVPR*, 2019. 2, 10

[58] Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. Talk2nav: Long-range vision-and-language navigation with dual attention and spatial memory. *IJCV*, 129:246–266, 2021. 2

[59] 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 *EMNLP*, 2019. 2

[60] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 5, 10

[61] Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. In *ICCV*, 2021. 2

[62] 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

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

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

[65] 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 *ECCV*, 2020. 2

[66] 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. 6

[67] Xiaohan Wang, Wenguan Wang, Jiayi Shao, and Yi Yang. Lana: A language-capable navigator for instruction following and generation. In *CVPR*, 2023. 2, 6

[68] Haoshuo Huang, Vihan Jain, Harsh Mehta, Alexander Ku, Gabriel Magalhaes, Jason Baldridge, and Eugene Ie. Transferable representation learning in vision-and-language navigation. In *ICCV*, 2019. 2

[69] Dong An, Yuankai Qi, Yan Huang, Qi Wu, Liang Wang, and Tieniu Tan. Neighbor-view enhanced model for vision and language navigation. In *ACM MM*, 2021. 2

[70] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun

Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *CVPR*, 2021. 2, 6
[71] 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
[72] Yi Yang, Yueting Zhuang, and Yunhe Pan. Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies. *FITEE*, 22(12):1551–1558, 2021. 2
[73] Xiangyang Li, Zihan Wang, Jiahao Yang, Yaowei Wang, and Shuqiang Jiang. Kerm: Knowledge enhanced reasoning for vision-and-language navigation. In *CVPR*, 2023. 2
[74] Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *ECCV*, 2018. 2
[75] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, 2020. 2
[76] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, 2020.
[77] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, 2018.
[78] Hanqing Wang, Wei Liang, Luc V Gool, and Wenguan Wang. Towards versatile embodied navigation. In *NeurIPS*, 2022.
[79] 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 *ICCV*, 2019. 2
[80] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3D): 1000 large-scale 3d environments for embodied AI. In *NeurIPS Datasets and Benchmarks*, 2021.
[81] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. In *NeurIPS*, 2021. 2
[82] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002. 2
[83] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998. 2
[84] 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. 2
[85] Jose-Luis Blanco, Juan-Antonio Fernández-Madrigal, and Javier Gonzalez. Toward a unified bayesian approach to hybrid metric–topological slam. *IEEE Transactions on Robotics*, 24(2):259–270, 2008. 2
[86] Clara Gomez, Marius Fehr, Alex Millane, Alejandra C Hernandez, Juan Nieto, Ramon Barber, and Roland Siegwart. Hybrid topological and 3d dense mapping through autonomous exploration for large indoor environments. In *ICRA*, 2020.
[87] Shun Niijima, Ryusuke Umeyama, Yoko Sasaki, and Hiroshi Mizoguchi. City-scale grid-topological hybrid maps for autonomous mobile robot navigation in urban area. In *IROS*, 2020. 2
[88] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013. 2
[89] Hema Koppula, Abhishek Anand, Thorsten Joachims, and Ashutosh Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *NeurIPS*, 2011.
[90] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.
[91] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *CVPR*, 2020.
[92] Aming Wu, Rui Liu, Yahong Han, Linchao Zhu, and Yi Yang. Vector-decomposed disentanglement for domain-invariant object detection. In *ICCV*, 2021.
[93] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, 2020. 2
[94] 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
[95] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2, 4
[96] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2, 4
[97] Siddhartha Chaudhuri, Daniel Ritchie, Jiajun Wu, Kai Xu, and Hao Zhang. Learning generative models of 3d structures. In *Computer Graphics Forum*, 2020. 2
[98] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *CVPR*, 2020. 2
[99] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *ICRA*, 2023. 2
[100] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. In *NeurIPS*, 2022.
[101] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d

object detection? In *ICCV*, 2021.

[102] Xiaoyang Guo, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Liga-stereo: Learning lidar geometry aware representations for stereo-based 3d detector. In *ICCV*, 2021.

[103] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3d object detection. In *CVPR*, 2020. 2

[104] Guanqi Zhan, Weidi Xie, Andrew Zisserman, and Coop Medianet Innovation Center. A tri-layer plugin to improve occluded detection. In *BMVC*, 2022. 2, 3, 5

[105] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *CVPR*, 2021. 2

[106] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, 2016.

[107] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018. 2

[108] D Wilkes and JK Tsotsos. Active object recognition. In *CVPR*, 1992. 3, 5

[109] Stephen E Palmer. *Vision science: Photons to phenomenology*. MIT press, 1999. 3, 5

[110] Yao-Hung Hubert Tsai, Hanlin Goh, Ali Farhadi, and Jian Zhang. Towards multimodal multitask scene understanding models for indoor mobile agents. In *ICRA*, 2023. 3, 4

[111] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, 2022. 3

[112] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 3

[113] Xinzhu Ma, Wanli Ouyang, Andrea Simonelli, and Elisa Ricci. 3d object detection from images for autonomous driving: a survey. *arXiv preprint arXiv:2202.02980*, 2022. 3

[114] 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. 3, 5, 10

[115] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions On Graphics*, 38(5):1–12, 2019. 3

[116] Yue Wang and Justin M Solomon. Object dgcnn: 3d object detection using dynamic graphs. In *NeurIPS*, 2021. 3

[117] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 4, 5

[118] William B Shen, Danfei Xu, Yuke Zhu, Leonidas J Guibas, Li Fei-Fei, and Silvio Savarese. Situational fusion of visual representation for visual navigation. In *ICCV*, 2019. 4

[119] Sinan Tan, Mengmeng Ge, Di Guo, Huaping Liu, and Fuchun Sun. Self-supervised 3d semantic representation learning for vision-and-language navigation. *arXiv preprint arXiv:2201.10788*, 2022. 4

[120] David Nilsson, Aleksis Pirinen, Erik Gärtner, and Cristian Sminchisescu. Embodied visual active learning for semantic segmentation. In *AAAI*, 2021. 4

[121] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 5

[122] 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. 5

[123] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5

[124] Xiangru Lin, Guanbin Li, and Yizhou Yu. Scene-intuitive agent for remote embodied visual grounding. In *CVPR*, 2021. 5, 6, 10

[125] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[126] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *NeurIPS*, 2016. 5

[127] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 8, 11

[128] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 8, 11

[129] Arthur Gregory, Ming C Lin, Stefan Gottschalk, and Russell Taylor. A framework for fast and accurate collision detection for haptic interaction. In *ACM SIGGRAPH*, 2005. 8

[130] Gino van den Bergen. Efficient collision detection of complex deformable models using aabb trees. *Journal of graphics tools*, 2(4):1–13, 1997. 8

[131] Kai Huebner, Steffen Ruthotto, and Danica Kragic. Minimum volume bounding box decomposition for shape approximation in robot grasping. In *ICRA*, 2008. 8

[132] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *ICRA*, 2015.

[133] Xinwen Zhou, Xuguang Lan, Hanbo Zhang, Zhiqiang Tian, Yang Zhang, and Narming Zheng. Fully convolutional grasp detection network with oriented anchor box. In *IROS*, 2018. 8

[134] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 10

[135] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020. 10

[136] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011. 10

[137] 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. 11
[138] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 11