BEVBert: Topo-Metric Map Pre-training for Language-guided Navigation

Dong An[1,4] Yuankai Qi[2] Yangguang Li[3] Yan Huang[1] Liang Wang[1] Tieniu Tan[1,5] Jing Shao[3]
[1]Institute of Automation, Chinese Academy of Sciences [2]University of Adelaide
[3]SenseTime Research [4]School of Future Technology, UCAS [5]Nanjing University

Abstract

*Existing approaches for vision-and-language navigation (VLN) are mainly based on cross-modal reasoning over discrete views. However, this scheme may hamper an agent's spatial and numerical reasoning because of incomplete objects within a single view and duplicate observations across views. A potential solution is mapping discrete views into a unified birds's-eye view, which can aggregate partial and duplicate observations. Existing metric maps could achieve this goal, but they suffer from less expressive semantics (*e.g.* usually predefined labels) and limited map size, which weakens an agent's language grounding and long-term planning ability. Inspired by the robotics community, we introduce hybrid topo-metric maps into VLN, where a topological map is used for long-term planning and a metric map for short-term reasoning. Beyond mapping with more expressive deep features, we further design a pre-training framework via the hybrid map to learn language-informed map representations, which enhances cross-modal grounding and facilitates the final language-guided navigation goal. Extensive experiments demonstrate the effectiveness of the map-based route for VLN, and the proposed method sets the new state-of-the-art on three VLN benchmarks.*

1. Introduction

Interaction with a service robot using natural language is a long-standing goal. Towards this goal, vision-and-language navigation (VLN) has been proposed and drawn increasing research interest [5, 26, 67]. Given a natural language instruction, a VLN agent is expected to interpret and follow the instruction to reach the desired location. Despite progress, it remains challenging to develop a robust VLN system.

Most existing approaches for VLN are based on cross-modal reasoning over discrete views to predict actions. Despite progress, they cannot sufficiently deal with spatial and numerical reasoning in complex environments, due to incomplete and duplicate observations in discrete views as shown in Fig. 1 (a), where it is highly challenging to reason about "the second bedroom to the bookcase" because it finds duplicate "bedrooms" across views as well as "bedroom" and "bookcase" fall in different views.

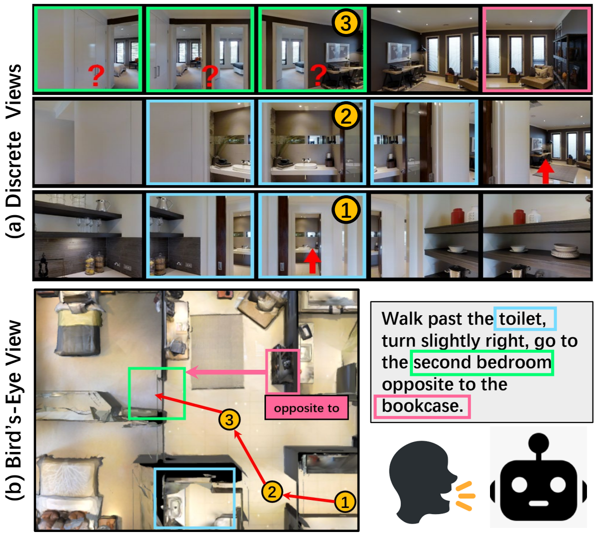

(a) Discrete Views
3
2
1
(b) Bird's-Eye View
opposite to
3
2
1
Walk past the toilet, turn slightly right, go to the second bedroom opposite to the bookcase.

Figure 1. (a) Incomplete observations within a single view and duplicates across views may confuse the agent. (b) Mapping discrete views into a unified space can solve the problem, thus facilitating spatial and numerical reasoning.

A potential solution is to map discrete views into a unified space as shown in Fig. 1 (b), which explicitly aggregates incomplete observations and removes duplicates. Though existing metric mapping techniques [4, 7, 10, 24, 37] may achieve this goal, they suffer two major limitations. First, they commonly rely on inexpressive representations (*e.g.* semantic labels) to construct a metric map, which misses object attributes and thus is insufficient to interpret instructions like "the brown sofa" accurately. Second, the grid-based map consists of dense cells and is typically restricted to a small scale to avoid heavy computation. Therefore, they are not suitable for long-term planning, such as backtracking to a previously visited location.

Considering graph-based topological maps have been shown critical for long-term planning [11, 13, 15], a natural idea is to combine them together. Thus, in this paper, we propose a hybrid semantic topo-metric map-based method named BEVBert. Beyond existing hybrid maps in robotics [6, 25, 44, 61], our metric map is built upon deep visual features such as CLIP [69], and we further design a hybrid-map pre-training framework for learning rich language-informed map representations to benefit complicated cross-modal reasoning and therefore ease the VLN task. Furthermore, to mitigate the partial observability in navigation (*e.g.*, some areas cannot be observed due to occlusions) , we investigate two solutions from historical and imaginary perspectives. The former employs a topology-guided map update strategy to incorporate visual clues from multi-hop histories flexibly. And the latter leverages a cross-modal imagination task to equip an agent with the ability to extrapolate unobserved areas from partial visual-linguistic clues. The proposed BEVBert presents an elegant balance between long-term planning and short-term reasoning, and clearly outperforms state-of-the-art methods on three VLN benchmarks (R2R, RxR, REVERIE).

We summarize the contributions of this work as follows:

- To the best of our knowledge, this is the first time to investigate a hybrid semantic topo-metric map scheme in the context of VLN. The proposed hybrid-map-based pre-training unleashes the great potential of map-based methods for VLN.
- To alleviate the partial observability problem, we devise a topology-guided map update strategy to flexibly incorporate multi-hop histories, and propose a cross-modal imagination task to enable the agent to imagine unobserved areas based on partial observation clues.
- Extensive experiments defend the map-based route for VLN. BEVBert achieves state-of-the-art on three benchmarks (*e.g.*, in test-unseen splits, 73% SR and 62% SPL on R2R dataset, 64.4% SR and 54.2% SDTW on RxR dataset).

2. Related Work

Maps for Navigation. Works on navigation have a long tradition of using SLAM [21] to construct metric maps [9, 10, 22]. A metric map is a grid-based representation consisting of dense cells that delicately represent the environment. To avoid heavy computation, common practices restrict the map size [24,37], which can be inadequate for long-term planning. Therefore, topological maps based on sparse nodes are proposed for long-term modeling and planning [11,13,17,48,76]. But the drawback is short-term reasoning within the condensed nodes [15]. In robotics, topo-metric maps are proposed to combine their advantages [6, 25, 44, 61]. However, they mainly focus on traditional robotic tasks, and the adaptations to advanced navigation tasks leave unexplored. We introduce hybrid maps into VLN and investigate the language-involved adaptations.

Vision-and-Language Navigation. VLN has drawn widely research interest in recent years [5, 26, 29, 45, 47, 67, 80, 88]. Popular approaches are based on cross-modal reasoning over discrete views [19, 58, 66, 79, 81]. Under the view-based scope, a variety of topics are well explored, such as extra object clues [1, 23, 31, 33, 60, 85], data augmentation [20, 50, 54, 62, 74, 75, 78], memory mechanism [14, 63, 76], pre-training [27, 28, 59, 65, 82]. Some recent methods [13, 15, 17, 76] reveal the potential of topological maps for long-term planning (*e.g.*, efficient backtrack to a previous location). A few attempts at metric maps [4, 24, 37] have drawbacks in language grounding or long-term modeling, thus leading to sub-optimal performance. We address the above limitations via a hybrid scheme and a map-based pre-training framework.

Visual Representation in Vision Language Pre-training. Existing approaches for vision language pre-training fall into image-based, object-based, and grid-based. Image-based methods [52, 69] extract an overall feature for an image, yet neglect details, thus drawback on fine-grained language grounding. Object-based methods [51, 57, 73, 84] represent an image with dozens of objects identified by external detectors [3, 46, 70]. The challenge is that objects can be redundant and limited in predefined categories. Grid-based methods [34, 35, 38, 41, 83] directly use image grid features for pre-training, thus enabling multi-grained vision-language alignments. Most VLN pre-training are image-based [14,27,28,59,68], which rely on discrete views. We introduce grid-based methods into VLN through metric maps, where the model can learn via multi-grained room layouts.

3. Method

Problem Definition. Given an instruction $\mathbf{W} = \{\mathbf{w}_l\}_{l=1}^{L}$ with L words, an agent is required to follow the instruction to traverse on a connectivity graph $\mathbf{G}^*$ to reach the target location. The agent perceives discrete panoramic views at each step t comprised of K RGB images $\mathbf{V}_t = \{\mathbf{v}_{t,k}\}_{k=1}^{K}$, where k denotes the direction. It also perceives associated depth images $\mathbf{D}_t = \{\mathbf{d}_{t,k}\}_{k=1}^{K}$ and camera poses $\mathbf{P}_t = \{\mathbf{p}_{t,k}\}_{k=1}^{K}$. The simulator provides these inputs, while in other settings, they can be estimated by SLAM systems, etc [4,21]. Given observations $\mathbf{O}_t = \{\mathbf{V}_t, \mathbf{D}_t, \mathbf{P}_t\}$, the goal of VLN is to learn a policy $\pi(\mathbf{a}_t|\mathbf{W}, \mathbf{O}_t; \theta)$ parametrized by θ to predict action $\mathbf{a}_t$.

Overview. We first convert discrete views into a hybrid map with deep learning features (Sec. 3.1). Then we use a pre-training framework to learn language-informed map representations with several proxy tasks (Sec. 3.2 and 3.3). After pre-training, the same network is fine-tuned on specific tasks for action prediction (Sec. 3.4).

3.1. Hybrid Topo-Metric Mapping

As illustrated in Fig. 2, we present our hybrid mapping pipeline and a bird's-eye view diagram. Note that the topological map is updated over time, while the metric map is temporarily constructed. Details are shown as follows. **Image Processing.** For the t-th step panoramic RGB images $\mathbf{V_t}$, we

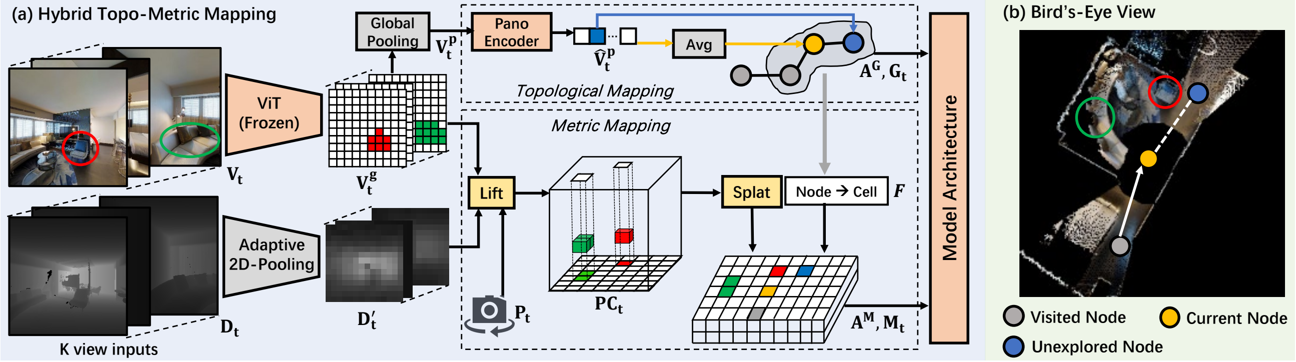


Figure 2. Illustration of hybrid mapping at step t. Topological map $\mathbf{G}_t$ is updated over time by adding new nodes and edges, where nodes are represented via panoramic view features $\mathbf{V}_t^p$. Metric map $\mathbf{M}_t$ is temporarily constructed by 'lift' and 'splat' of multi-camera grid features $\mathbf{V}_t^g$. The one-hop neighbors of the current node (including the current node) in global action space $\mathbf{A}^G$ are projected ('Node→Cell') onto cells as local action space $\mathbf{A}^M$ for short-term action reasoning.

first use a pretrained vision transformer (ViT) [18] to extract view features $\mathbf{V}_t^p = \{\mathbf{v}_{t,k}^p | \mathbf{v}_{t,k}^p \in \mathcal{R}^D\}_{k=1}^K$ and downsized grid features $\mathbf{V}_t^g = \{\mathbf{v}_{t,k}^g | \mathbf{v}_{t,k}^g \in \mathcal{R}^{H' \times W' \times D}\}_{k=1}^K$. The associated depth images are downsized to the same scale as $\mathbf{D}_t^{'} = \{\mathbf{d}_{t,k}^{'} | \mathbf{d}_{t,k}^{'} \in \mathcal{R}^{H' \times W'}\}_{k=1}^K$.

Topological Mapping. The topological map $\mathbf{G}_t$ is a graph-based representation consisting of nodes and edges $\mathbf{G}_t = \{\mathbf{N}_t, \mathbf{E}_t\}$, and is updated over time to model long-term dependence. Nodes $\mathbf{N}_t$ fall into three categories: visited node ●, current node ● and unexplored node ●. Edges $\mathbf{E}_t$ represent spatial relations among these nodes (*e.g.* orientations, connectivity). For simplicity, we demonstrate a single-step map update in Fig. 2. A pano encoder (a two-layer transformer) is first used to obtain contextual view embeddings $\hat{\mathbf{V}}_t^p$, and then use these embeddings to update current and observed unexplored nodes. For node ● $\mathbf{n}_i \in \mathbf{N}_t$, since it has access to the panorama, its representation updates to be $\mathbf{n}_i = \text{Avg}(\hat{\mathbf{V}}_t^p) \in \mathcal{R}^D$. The partially observed ● is represented by the corresponding view embedding $\hat{\mathbf{v}}_{t,k}^p$ in the panorama.

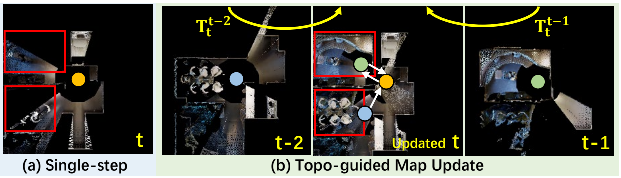
Metric Mapping. A grid-based metric map $\mathbf{M}_t \in \mathcal{R}^{U \times V \times D}$ is constructed temporarily at each step, in which each cell contains a D-sized latent feature representing a small region of the environment. We define $\mathbf{M}_t$ as an egocentric local map, and we place the agent at the central cell $(\lfloor \frac{U}{2} \rfloor, \lfloor \frac{V}{2} \rfloor)$. Inspired by MapNet [30] and LSS [64], we use 'lift' (inverse pinhole camera projection) and 'splat' (orthographic projection) to integrate multi-camera grid features $\mathbf{V}_t^g$ into $\mathbf{M}_t$. Specifically, the grid features are first 'lift' into 3D semantic point clouds $\mathbf{PC}_t$ by shooting rays from the camera center along the associated depths $\mathbf{D}_t^{'}$. Then, 'splat' and discretize features $\mathbf{PC}_t$ into corresponding cells of $\mathbf{M}_t$, using elementwise average pooling to handle feature collisions in a cell. We simplify the procedure as:

$$\mathbf{PC}_t = \text{Lift}(\mathbf{V}_t^g, \mathbf{D}_t^{'}; \mathbf{P}_t), \quad \mathbf{M}_t = \text{Splat}(\mathbf{PC}_t). \tag{1}$$

Topo-guided Map Update. $\mathbf{M}_t$ obtained by Eq. 1 is a single-step metric map, which is under partial observability as shown in Fig. 3 (a). A typical solution is employing a recurrent network to incorporate histories and update the map [4, 30]. However, this recurrent scheme may lead to biased terms due to accumulating observations. To solve this problem, we devise a topology-guided map update (TMU) strategy, which can flexibly incorporate multi-hop histories from visited nodes as shown in Fig. 3 (b). Specifically, for the current node $\mathbf{n}_i$, we first combine point cloud features within its κ-hop visited nodes. Then, these features are aligned to the same coordinate system and 'splat' as the updated map. Formally, we supplement Eq. 1 as:

$$\overline{\mathbf{PC}}_t = \bigcup_{\mathbf{h}(i,j) \le \kappa} \mathbf{T}_i^j \cdot \mathbf{PC}_{\mathbf{n}_j}, \quad \mathbf{M}_t = \text{Splat}(\overline{\mathbf{PC}}_t) \tag{2}$$

where $\mathbf{h}(i,j)$ returns the hop between nodes $\mathbf{n}_i$ and $\mathbf{n}_j$, and $\mathbf{T}_i^j$ represents the coordinate alignment operation. Note that we cache point cloud features in each visited node.



(a) Single-step
(b) Topo-guided Map Update

Figure 3. A diagram of map update. (a) A single-step metric map is under partial observability. (b) Flexibly update the map with multi-hop histories. ($\mathbf{T}_t^*$ denote coordinate alignments).

Global and Local Action Space. We apply a global action space $\mathcal{A}^G$ for long-term planning on the topological map. For short-term action reasoning, we project the one-hop neighbors of the current node (including the current node) in $\mathcal{A}^G$ onto the metric map as local action space $\mathcal{A}^M$. Formally:

$$\mathcal{A}^G = \{i | i \in \bigcup_{t'=1}^{t} \mathcal{A}_{t'}\}, \quad \mathcal{A}^M = \{(u,v) | (u,v) \in \mathcal{F}(\mathcal{A}_t)\} \tag{3}$$

where $\mathcal{A}_t$ denotes the t-th step adjacent navigable nodes provided by the simulator, i and (u,v) denote indexs of nodes and cells, $\mathcal{F}$ denotes the 'Node→Cell' operation.

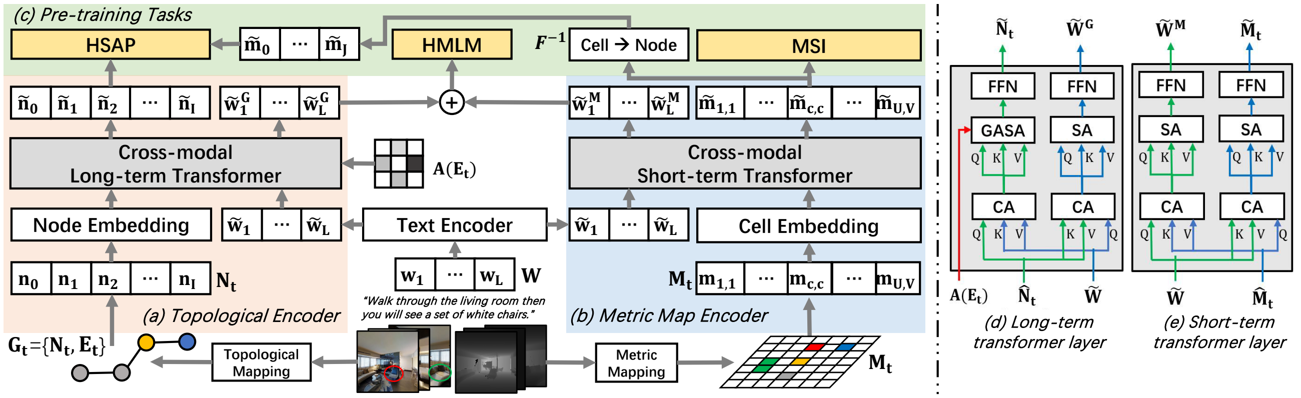

(c) Pre-training Tasks
HSAP
$\tilde{m}_0$ ⋯ $\tilde{m}_J$
HMLM
F^{-1}
Cell → Node
MSI
$\tilde{n}_0$ $\tilde{n}_1$ $\tilde{n}_2$ ⋯ $\tilde{n}_I$
$\tilde{w}_1^G$ ⋯ $\tilde{w}_L^G$
$\tilde{w}_1^M$ ⋯ $\tilde{w}_L^M$
$\tilde{m}_{1,1}$ ⋯ $\tilde{m}_{c,c}$ ⋯ $\tilde{m}_{U,V}$
Cross-modal Long-term Transformer
A(E_t)
Cross-modal Short-term Transformer
Node Embedding
$\tilde{w}_1$ ⋯ $\tilde{w}_L$
Text Encoder
$\tilde{w}_1$ ⋯ $\tilde{w}_L$
Cell Embedding
n_0 n_1 n_2 ⋯ n_I N_t
w_1 ⋯ w_L W
M_t $m_{1,1}$ ⋯ $m_{c,c}$ ⋯ $m_{U,V}$
"Walk through the living room then you will see a set of white chairs."
(a) Topological Encoder
(b) Metric Map Encoder
$G_t=\{N_t, E_t\}$
Topological Mapping
Metric Mapping
M_t
$\tilde{N}_t$ $\tilde{W}^G$ $\tilde{W}^M$ $\tilde{M}_t$
FFN FFN FFN FFN
GASA SA SA SA
Q K V
CA CA CA CA
A(E_t) $\hat{N}_t$ $\tilde{W}$ $\tilde{W}$ $\hat{M}_t$
(d) Long-term transformer layer
(e) Short-term transformer layer

Figure 4. The pre-training model takes the topo-metric map $(\mathbf{G}_t, \mathbf{M}_t)$ and an instruction $\mathbf{W}$ as inputs. The topological encoder performs cross-modal long-term reasoning over $\mathbf{G}_t$, while the metric map encoder performs short-term reasoning over $\mathbf{M}_t$. Outputs of the two encoders are fed into pre-training tasks to learn navigation-oriented language-informed map representations ($\tilde{\mathbf{n}}_*$ and $\tilde{\mathbf{m}}_{*,*}$). Before hybrid action prediction (HSAP), the 'Cell→Node' operation is used to align cells and nodes into the same action space.

3.2. Pre-training Model

We present the model architecture in Fig. 4. It takes the hybrid map $(\mathbf{G}_t, \mathbf{M}_t)$ and an instruction $\mathbf{W}$ as inputs. A topological encoder and a metric map encoder first perform cross-modal reasoning, respectively. Their outputs are combined and fed into three pre-training tasks to learn navigation-oriented language-informed map representations.

3.2.1 Text Encoder

Each word embedding in the instruction $\mathbf{W}$ is added with a position embedding and a text type embedding. Then, all embeddings are fed into a 9-layer transformer to obtain contextual word representations $\tilde{\mathbf{W}} = \{\tilde{\mathbf{w}}_l\}_{l=1}^{L}$.

3.2.2 Topological Encoder

Node Embedding. Each node feature $\mathbf{n}_i \in \mathbf{N}_t$ is added with a location embedding and a navigation step embedding. The location embedding is calculated by the relative orientation and distance of each node to the current node, and the step embedding is the latest visited time step for visited nodes and 0 for unexplored nodes. A zero-vector 'stop' node $\mathbf{n}_0$ is padded to denote the stop action. The resulted node embeddings are $\hat{\mathbf{N}}_t = \{\hat{\mathbf{n}}_i\}_{i=0}^{I}$.

Cross-modal Long-term Transformer. We adopt a 4-layer transformer to model node-level vision-language relations. The architecture of each layer is detailed in Fig. 4 (d). Within each layer, we first model inter-modal relations with cross-attention (CA) and then model intra-modal relations with self-attention (SA). We use graph-aware self-attention (GASA) to introduce topology for node relation modeling:

$$\mathrm{GASA}(\mathbf{X}) = \mathrm{Softmax}(\frac{\mathbf{X}\mathbf{W}_q(\mathbf{X}\mathbf{W}_k)^\top}{\sqrt{\mathbf{D}}} + \mathbf{A}(\mathbf{E}_t))\mathbf{X}\mathbf{W}_v \quad (4)$$

where $\mathbf{W}_* \in \mathcal{R}^{D \times D}$ are parameters and bias are omitted, and $\mathbf{A}(\mathbf{E}_t)$ denotes the spatial affinity matrix, which composes of pair distances among all observed nodes. The outputs are node-instruction-associated representations $\tilde{\mathbf{N}}_t = \{\tilde{\mathbf{n}}_i\}_{i=0}^{I}$ and $\tilde{\mathbf{W}}^G = \{\tilde{\mathbf{w}}_l^G\}_{l=1}^{L}$.

3.2.3 Metric Map Encoder

Cell Embedding. Each cell feature $\mathbf{m}_{u,v} \in \mathbf{M}_t$ is added with a position embedding $\mathbf{p}_{u,v}$ and a navigability embedding $\mathbf{n}_{u,v}$. To capture the relations between the agent and surrounding room layouts, we design an egocentric polar position embedding for each cell:

$$\mathbf{p}_{u,v} = [\cos(\theta_{u,v}), \sin(\theta_{u,v}), \mathrm{dis}_{u,v}] \quad (5)$$

where $\theta_{u,v}$ and $\mathrm{dis}_{u,v}$ denote the relative heading and normalized distance of a cell to the map center (agent position). We empirically found it is better than a learnable [40] or 2D position embedding [18]. Navigability embeddings are set to 1 for cells that lie in the local action space $\mathcal{A}^M$, and 0 otherwise. Both position and navigability embeddings are linearly transformed to D-dimension. The resulted cell embeddings are $\hat{\mathbf{M}}_t = \{\hat{\mathbf{m}}_{u,v}\}_{u=1,v=1}^{U,V}$.

Cross-modal Short-term Transformer. Similar to Sec. 3.2.2, we adopt a 4-layer transformer to model cell-level vision-language relations, with the encoded instruction $\tilde{\mathbf{W}}$ and all cell embeddings $\hat{\mathbf{M}}_t$. The short-term transformer layer is detailed in Fig. 4 (e). Each layer consists of a cross-attention (CA) layer for intra-modal relations and a self-attention (SA) layer for fine-grained (cell-level) relations of room layouts. It can benefit reasoning about spatial relations, such as *"go into the hallway second to the right from the stairs"*. The outputs are instruction-cell-associated representations $\tilde{\mathbf{W}}^M = \{\tilde{\mathbf{w}}_l^M\}_{l=1}^{L}$ and $\tilde{\mathbf{M}}_t = \{\tilde{\mathbf{m}}_{u,v}\}_{u=1,v=1}^{U,V}$.

3.3. Pre-training Tasks

Given an offline expert trajectory $\mathbf{\Gamma} = \langle \mathbf{O}_1, ..., \mathbf{O}_t \rangle$ and the associated instruction $\mathbf{W}$ sampled from the training set $\mathcal{D}$, we devise three pre-training tasks to learn navigation-oriented language-informed map representations.

Hybrid Masked Language Modeling (HMLM). HMLM is modified from the commonly used Masked Language

Modeling (MLM) proxy task in BERT pre-training [40]. For VLN, HMLM aims to recover masked words $\mathbf{W}_m$ via reasoning over the surrounding words $\mathbf{W}_{\backslash m}$ and a topometric map. Precisely, we first randomly mask out input tokens of the instruction with a 15% probability. Then, we use the procedure in Sec. 3.1 to construct the hybrid map $(\mathbf{G}_t, \mathbf{M}_t)$ at step t. This task is optimized by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{HMLM}}(\theta) = -\mathbb{E}_{(\mathbf{W},\mathbf{\Gamma})\sim\mathcal{D}} \log \mathcal{P}_\theta(\mathbf{W}_m|\mathbf{W}_{\backslash m}, \mathbf{G}_t, \mathbf{M}_t) \qquad (6)$$

As shown in Fig. 4 (c), before the HMLM head, We sum the latent word representations $(\tilde{\mathbf{W}}^G, \tilde{\mathbf{W}}^M)$ to learn both long-term and short-term reasoning.

Hybrid Single Action Prediction (HSAP). HSAP is designed to benefit the downstream goal: predicting navigation actions. To learn a long-term and short-term aware navigation policy, we integrate action predictions from $\tilde{\mathbf{N}}_t$ and $\tilde{\mathbf{M}}_t$. Since the two maps have different action spaces, we first use the 'Cell→Node' operation $\mathcal{F}^{-1}$ in Fig. 4 (c) to align them. We denote the aligned cells as $\{\tilde{\mathbf{m}}_j | j \in \mathcal{A}^{G'}\}$, where $\mathcal{A}^{G'}$ is a subset of global action space $\mathcal{A}^G$. Two separate feed-forward networks (FFN) are used to predict node-level and cell-level scores, and we gated fuse the scores (conditioned on the agent state):

$$\mathbf{s}_i^G = \text{FFN}(\tilde{\mathbf{n}}_i), \quad \mathbf{s}_i^M = \text{FFN}(\tilde{\mathbf{m}}_i) \qquad (7)$$

$$\mathbf{s}_i = \begin{cases} \delta_t \mathbf{s}_i^G + (1-\delta_t)\mathbf{s}_i^M, \text{ if } i \in \mathcal{A}^G \cap \mathcal{A}^{G'} \\ \mathbf{s}_i^G, \text{ otherwise} \end{cases} \qquad (8)$$

where we use the padded node $\tilde{\mathbf{n}}_0$ and central cell $\tilde{\mathbf{m}}_{c,c}$ to represent the state, thus $\delta_t = \text{Sigmoid}(\text{FFN}([\tilde{\mathbf{n}}_0; \tilde{\mathbf{m}}_{c,c}])$. The task is optimized via a cross-entropy loss over the fused scores and teacher action $\mathbf{a}_t^*$:

$$\mathcal{L}_{\text{HSAP}}(\theta) = -\mathbb{E}_{(\mathbf{W},\mathbf{\Gamma},\mathbf{a}_t^*)\sim\mathcal{D}} \log \mathcal{P}_\theta(\mathbf{a}_t^*|\mathbf{W}, \mathbf{G}_t, \mathbf{M}_t) \qquad (9)$$

Masked Semantic Imagination (MSI). MSI is an imagination-based task to mitigate the partial observability issue. The agent is expected to extrapolate unobserved regions of the metric map by reasoning over partial visual-linguistic clues (see Fig. 5). We first randomly mask out cells of the metric map $\mathbf{M}_t$ with a 15% probability to simulate and magnify the partial observability. Then, the MSI head in Fig. 4 (c) forces the agent to predict semantics of masked regions $\mathbf{S}$ conditioned on the language-informed map representation $\tilde{\mathbf{M}}_t$. Each cell of the discretized map may contain multiple semantics; therefore, the task is formulated as a multi-label classification problem and optimized via a binary cross-entropy loss:

$$\begin{aligned} \mathcal{L}_{\text{MSI}}(\theta) = -\mathbb{E}_{(\mathbf{W},\mathbf{\Gamma})\sim\mathcal{D}} \sum_i^C [&\mathbf{S}_i \log \mathcal{P}_\theta(\mathbf{S}_i|\mathbf{W}, \mathbf{M}_{t,\backslash m}) \\ &+ (1-\mathbf{S}_i)\log(1-\mathcal{P}_\theta(\mathbf{S}_i|\mathbf{W}, \mathbf{M}_{t,\backslash m}))] \end{aligned} \qquad (10)$$

where $\mathbf{S}_i$ corresponds to the i-th semantic class ($C = 40$), and we obtain these labels from Matterport3D dataset [8].

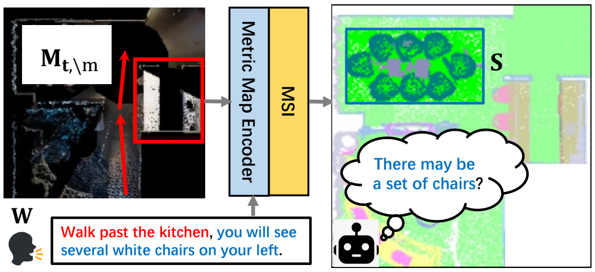

$\mathbf{M}_{t,\backslash m}$
Metric Map Encoder
MSI
S
There may be a set of chairs?
W
Walk past the kitchen, you will see several white chairs on your left.

Figure 5. MSI enables extrapolation of latent representations of unobserved regions by reasoning over partial visual-linguistic clues.

3.4. Training and Inference

Training. As standard practices in transformer-based VLN methods [14, 27, 28, 59], we first mix the three tasks in Sec. 3.3 to learn a pre-trained model based on offline expert data. To avoid overfitting to expert experience, we alternately run 'teacher-forcing' and 'student-forcing' to finetune the pre-trained model. The 'teacher-forcing' is equivalent to Eq. 9, where the agent follows an expert path. The 'student-forcing' generates paths by on-policy action sampling and supervised by pseudo labels [5, 15], which consist of the optimal action of each step, *e.g.*, a navigable node that has the shortest distance to the final target. More details are in the appendix.

Inference. At each time step during testing, the agent constructs a hybrid map as described in Sec. 3.1, and then performs cross-modal reasoning over the map as explained in Sec. 3.2. Following the single-run setting of VLN, the next action is greedily selected based on the predicted scores in Eq. 8. The agent will stop if it selects the 'stop' node or exceeds the maximum action steps.

4. Experiments

We evaluate the proposed method on three popular VLN datasets: R2R [5], RxR [47] and REVERIE [67]. R2R and RxR focus on low-level instruction following, which give detailed instructions. RxR is much more challenging than R2R due to much longer instructions and non-shortest trajectory from starting point to ending point. By contrast, REVERIE is a goal-oriented navigation task using concise high-level instructions, focusing more on knowledge exploitation than instruction following.

Navigation Metrics. As in [2,5,36], we adopt the following navigation metrics. Trajectory Length (TL): average path length in meters; Navigation Error (NE): average distance in meters between the final and target location; Success Rate (SR): the ratio of paths with NE less than 3 meters; Oracle SR (OSR): SR given oracle stop policy; SR penalized by Path Length (SPL); Normalize Dynamic Time Wrapping (NDTW): the fidelity between the predicted and annotated paths and NDTW penalized by SR (SDTW).

Object Grounding Metrics. As in [67], we use Remote Grounding Success (RGS) and RGSPL (RGS penalized by

Path Length) to evaluate the capacity of grounding objects. All metrics are the higher the better, except for TL and NE.

4.1. Implementation Details

Image Processing and Mapping. We resize and central crop RGB images to 224×224, and use ViT-B/16-CLIP [69] to extract grid features and view features. The scale of grid features is 14×14 (outputs before the MLP head of ViT), and we downsize depth images into the same scale. The depth images and extra semantic annotations used in MSI are obtained in Habitat Simulator [71]. We set the metric map scale as 21×21, and each cell represents a square region with a side length of 0.5m (the entire map is thus $10.5m \times 10.5m$).
Model Architecture. Hyperparameters of our model are the same with LXMERT [73] (*e.g.* the hidden layer size is 768). In the pre-training stage, we use pre-trained LXMERT for initialization on R2R and REVERIE datasets, and pre-trained RoBerta [55] is used for the multilingual RxR dataset. REVERIE provides additional object annotations for the final object grounding task, and BEVBert adaptation to this dataset is presented in the appendix.
Training Details. For all datasets, we first offline pre-train BEVBert with batch size 64 for 100k iterations using 4 NVIDIA Tesla A100 GPUs ($\sim$10 hours). We use the Prevalent [28], RxR-Markey [78] and REVERIE-Spk [15] synthetic instructions as data augmentation on R2R, RxR and REVERIE respectively. We choose a pre-trained model with the best zero-shot performance (*e.g.*, SR + SPL on R2R, SR + NDTW on RxR, SR + RGS on REVERIE) as initialization for downstream fine-tuning. Then, we use alternative 'teacher-forcing' and 'student-forcing' to online fine-tune the model in the simulator, with batch size 16 for 40k iterations on 4 NVIDIA Tesla A100 GPUs ($\sim$20 hours). The best iterations are selected by best performance on validation unseen splits. More details are in the appendix.

4.2. Ablation Study

We conduct ablation study on the R2R val unseen split, and results of essential metrics are highlighted in grey.
1) Comparison of map variants. We analyze different variants of maps in Tab. 1. In row 1, we only use topological maps for action predictions. Though it achieves decent SR (65.52%), the TL is exceptionally long, thus leading to low SPL (43.04%), which may attributed to its less-accurate cross-modal grounding ability that makes the agent walk a long way. In contrast, metric maps are only used in row 2 and row 3. The TL decreases significantly, but the navigation performance is poor (*e.g.* 61.54% OSR and 52.15% SR). This may be caused by that the agent stops too early and lacks the long-term planning capacity to correct mistakes. In row 4 and row 5, the navigation performance increases remarkably when applying the proposed hybrid topo-metric maps (*e.g.* 74.88% SR and 63.60% SPL). It indicates that hybrid maps could combine the advantages of the above two maps and enable the agent to make long-term and short-term balanced decisions.

Map	Depth	#	TL	NE↓	OSR↑	SR↑	SPL↑
Topological	-	1	26.90	3.59	84.72	65.52	43.04
Metric	estimated	2	14.37	4.76	62.03	51.46	45.47
	sensing	3	13.71	4.74	61.54	52.15	45.85
Hybrid	estimated	4	13.61	2.88	82.63	74.67	63.63
	sensing	5	14.55	2.81	83.65	74.88	63.60

Table 1. Comparison of map variants. Depths images are only used in metric mapping. We denote ground-truth and estimated depths as 'sensing' and 'estimated' respectively.

2) Are depth sensors necessary? We adopt in-domain pre-trained RedNet [39] for depth estimation, and then investigate the dependence on depth sensors of BEVBert. As shown in Tab. 1 (row 2 vs. row 3, row 4 vs. row 5), the performance of BEVBert does not drop severely when applying estimated depths for metric mapping, which presents that our approach does not highly rely on accurate depth sensing. The main reason is that metric maps are constructed in feature space, where features are projected guided by rough grid depths (*e.g.* 14×14). We believe BEVBert has the potential to be extended in large-scale training with synthetic environments [42, 43], where depth sensors are unavailable.
3) The effect of pre-training tasks. In Tab. 2, we present the results of different proxy tasks in the pre-training stage. BEVBert, with the generic HMLM task, can achieve decent performance (*e.g.* 73.52% SR and 60.13% SPL). However, it has the highest TL, thus leading to lower SPL compared to the other two variants. On the other hand, the TL decreases, and SPL increases significantly after applying the HSAP task, *e.g.*, SPL from 60.13% to 63.03%. It indicates that action prediction tasks are beneficial to learn action-informed map representations for the downstream navigation goal. Furthermore, in row 3, the proposed MSI task increases the navigation performance successfully (*e.g.* SR from 74.03% to 74.88% and SPL from 63.03% to 63.60%) and reduces the stop error (NE from 3.03% to 2.81%). The potential reason is that the agent learns to imagine over partially observed environments with MSI, which helps generalize unseen environments.

Proxy Tasks	#	TL	NE↓	OSR↑	SR↑	SPL↑
HMLM	1	16.26	3.09	83.82	73.52	60.13
HMLM + HSAP	2	14.50	3.03	82.67	74.03	63.03
HMLM + HSAP + MSI	3	14.55	2.81	83.65	74.88	63.60

Table 2. The effect of pre-training tasks.

4) Hop of topology-guided map update. In Tab. 3, we present the effect of update hop in Eq. 2. When $\kappa = 0$, the metric map degenerates into 'single-step map', and it has the worst performance because of the lack of historical observations. The performance improves (*e.g.* SPL from 62.37% to 63.60%) when incorporating visual clues from 1-hop visited nodes, which mitigates partial observability.

However, no more improvement as κ goes up. Because 1-hop update is enough for a small-scale local map, a higher hop may introduce noisy observations for making the current decision.

κ	#	TL	NE↓	OSR↑	SR↑	SPL↑
0	1	14.43	3.01	82.12	73.73	62.37
1	2	14.55	2.81	83.65	74.88	63.60
2	3	14.89	2.81	84.29	75.18	62.71

Table 3. The effect of hop in TMU.

5) Scale and size of metric maps. In Tab. 4, we evaluate the scale and size of metric maps and report the short-term transformer flops. We observe an upward trend in navigation performance as the map size increases (row 2 vs. row 1) because the agent could perceive environments in the broader scope. The agent performs slightly better when the cell size decreases (row 3 vs. row 2), contributing to a better perception of minor objects. We also investigate the larger map scale in row 4, but the navigation performance does not increase obviously. Our approach uses topological maps for long-term planning; thus, large-scale metric maps only bring marginal benefit. On the other hand, a larger metric map brings heavy computation (*e.g.*, flops of the short-term transformer are approximately quadratic w.r.t. the map scale). Therefore, row 3 is our default setting.

Scale*	Cell Size	Map Size	Flops	#	NE↓	OSR↑	SR↑	SPL↑
11×11	$0.5m^2$	$5.5m^2$	4.5G	1	2.98	81.61	73.27	63.07
11×11	$1.0m^2$	$11.0m^2$	4.5G	2	2.82	83.01	74.58	63.37
21×21	$0.5m^2$	$10.5m^2$	15.2G	3	2.81	83.65	74.88	63.60
31×31	$0.5m^2$	$15.5m^2$	32.7G	4	2.83	83.23	74.84	64.88

Table 4. The effect of metric maps scale and size. *Scales are set to odd to ensure the agent is at the central cell.

6) Visual features. BEVBert achieves better performance with CLIP pre-trained features as shown in Tab. 5. Imagenet features may lack diverse visual concepts, because they are learned by a one-hot classification task that focuses on salient regions of images. In contrast, CLIP features are learned by large-scale image-text matching, where visual grid features are informed by diverse linguistic concepts, which can be more suitable for metric mapping.

Features	#	TL	NE↓	OSR↑	SR↑	SPL↑
ViT-B/16-ImageNet [16]	1	15.90	2.91	83.44	74.03	61.86
ViT-B/16-CLIP [69]	2	14.55	2.81	83.65	74.88	63.60

Table 5. Comparison of different visual features.

4.3. Comparison with State of the Art

We compare BEVBert with the state of the arts on unseen splits, and more results (seen splits) are in the appendix.

R2R. Tab. 6 compares BEVBert against state-of-the-art (SOTA) methods on the R2R dataset in the single-run setting (greedy search, no pre-exploitation [79]). Our approach beats SOTA methods on all evaluation metrics. For example, on the val unseen split, BEVBert outperforms the previous best model DUET [15] by 3% on SR and 4% on SPL. BEVBert also generalizes well on the test unseen split[1], where we improve over DUET by 4% on SR and 3% on SPL. Compared with Chasing [4], which also uses depths for metric mapping, our performance is noteworthy (*e.g*. 75% SR vs. 35% SR on the val unseen split). The improvements mainly come from the hybrid map, which balances long-term planning and short-term reasoning, while Chasing is based on pure metric maps leading to non-ideal long-term planning capacity (*e.g.* backtracking when navigation error). Moreover, Chasing is trained from scratch, while BEVBert is pre-trained through the proposed framework leading to better language-guided grounding capacity.

	Val Unseen				Test Unseen			
Methods	NE↓	OSR↑	SR↑	SPL↑	NE↓	OSR↑	SR↑	SPL↑
Human	-	-	-	-	1.61	90	86	76
Seq2Seq [5]	7.81	28	21	-	7.85	27	20	-
SF [19]	6.62	45	36	-	6.62	-	35	28
Chasing [4]	7.20	44	35	31	7.83	42	33	30
RCM [79]	6.09	50	43	-	6.12	50	43	38
SM [58]	5.52	56	45	32	5.67	59	48	35
EnvDrop [74]	5.22	-	52	48	5.23	59	51	47
AuxRN [87]	5.28	62	55	50	5.15	62	55	51
NvEM [1]	4.27	-	60	55	4.37	66	58	54
SSM [76]	4.32	73	62	45	4.57	70	61	46
AirBert [27]	4.10	-	62	56	4.13	-	62	57
SEvol [12]	3.99	-	62	57	4.13	-	62	57
RecBert [32]	3.93	-	63	57	4.09	70	63	57
EnvMix [54]	3.89	-	64	58	3.87	72	65	59
HAMT [14]	3.65	-	66	61	3.93	72	65	60
TD-STP [86]	3.22	76	70	63	3.73	72	67	61
DUET [15]	3.31	81	72	60	3.65	76	69	59
BEVBert (Ours)	2.81	84	75	64	3.13	81	73	62

Table 6. Comparison with SOTA methods on R2R dataset.

	Val Unseen				Test Unseen			
Methods	NE↓	SR↑	NDTW↑	SDTW↑	NE↓	SR↑	NDTW↑	SDTW↑
Human	-	-	-	-	0.9	93.9	79.5	76.9
LSTM [47]	10.9	22.8	38.9	18.2	12.0	21.0	36.8	16.9
EnvDrop+ [72]	-	42.6	55.7	-	-	38.3	51.1	32.4
CLEAR-C [49]	-	-	-	-	-	40.3	53.7	34.9
HAMT [14]	-	56.5	63.1	48.3	6.2	53.1	59.9	45.2
EnvEdit* [50]	-	62.8	68.5	54.6	5.1	60.4	64.6	51.8
BEVBert†(ours)	4.6	64.1	63.9	52.6	-	-	-	-
BEVBert (ours)	4.0	68.5	69.6	58.6	4.8	64.4	65.4	54.2

Table 7. Comparison with SOTA methods on RxR dataset. *Ensemble of three agents. †Without Marky synthetic instructions [78].

RxR. Tab. 7 represents the results on the RxR dataset. RxR is more challenging because of more descriptions of visual entities and spatial relations than R2R. With the unified fine-grained metric map representation, BEVBert is skilled at these complex instructions and achieves considerate improvement on RxR. For instance, on the val unseen split, BEVBert improves over a previous ensemble-based method EnvEdit [50] by 5.7% on SR, 1.1% on NDTW, and 4% on SDTW. It also generalizes well on the test unseen split[2], where we improve over EnvEdit by 4% on SR, 0.8% on NDTW and 2.4% on SDTW. We also note that fidelity metrics (NDTW, SDTW) improve less than the accuracy met-

[1] https://eval.ai/web/challenges/challenge-page/97/leaderboard/270

[2] https://ai.google.com/research/rxr/

ric (SR). This is because the global action space specially designed for backtracking leads to longer path length and affects path fidelity (NDTW). We also report the results of BEVBert without Marky synthetic instructions [78] for a fair comparison. Compared to EnvEdit, BEVBert only leads on SR, but the improvements over the SOTA single-model HAMT [14] are notable (*e.g.* increasing SR by 7.6%, NDTW by 0.8% and SDTW by 4.3% on val unseen split).

Methods	Val Unseen			Test Unseen		
	SR↑	RGS↑	RGSPL↑	SR↑	RGS↑	RGSPL↑
Human	-	-	-	81.51	77.84	51.44
FAST [67]	14.40	7.84	4.67	19.88	11.28	6.08
SIA [53]	31.53	22.41	11.56	30.80	19.02	9.20
RecBert [32]	30.67	18.77	15.27	29.61	16.50	13.51
AirBert [27]	27.89	18.23	14.18	30.26	16.83	13.28
HAMT [14]	32.95	18.92	17.28	30.40	14.88	13.08
TD-STP [86]	34.88	21.16	16.56	35.89	19.88	15.40
DUET [15]	46.98	32.15	23.03	52.51	31.88	22.06
BEVBert (Ours)	51.78	34.71	24.44	52.81	32.06	22.09

Table 8. Comparison with SOTA methods on REVERIE dataset.

REVERIE. BEVBert also generalizes to the goal-oriented REVERIE dataset as shown in Tab. 8, where the agent is required to follow a concise instruction to find the target object. On the val unseen split, BEVBert surpasses the previous best model DUET [15] by 4.80% on SR, 2.64% on SPL, 2.56% on RGS, and 1.41% on RGSPL. We also note improvements on the test unseen split[3] are not prominent compared to DUET. We attribute it to the distribution shit between the val unseen and test unseen splits.

4.4. Quantitative and Qualitative Analysis

Quantitative Analysis. We investigate the performance of BEVBert and SOTA methods on spatial or numerical related instructions in Fig. 6. As the number of special tokens in each instruction goes up, the performance of all models shows downward trends, which indicates spatial and numerical reasoning capacity is a bottleneck of existing approaches. However, our BEVBert performs better than second best methods, especially on the RxR dataset which involves more spatial and numerical descriptions. It presents the better spatial and numerical reasoning capacity of our approach.

Qualitative Analysis. We visualize the predicted trajectories of BEVBert and the SOTA DUET [15] in Fig. 7. DUET uses view-level tokens for local reasoning and has non-ideal spatial reasoning capacity. For example, it does not follow the instruction strictly (*e.g.* "go between the kitchen counters", "walk behind the couch") and leads to incorrect endpoints. In contrast, BEVBert could interpret these complex descriptions and make correct decisions.

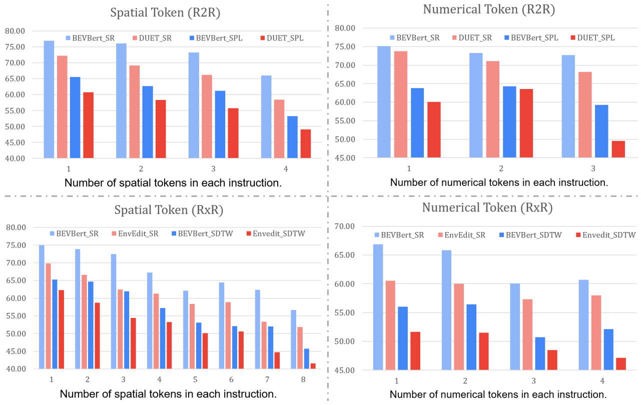
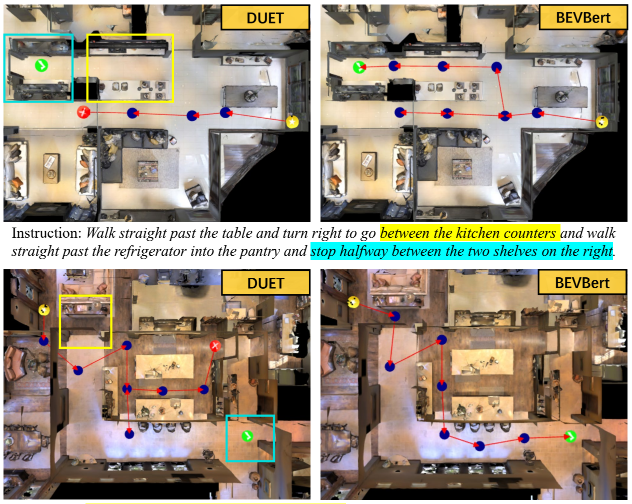


Figure 6. Comparison of navigation performance on spatial and numerical related instructions (BEVBert vs. DUET [15] SR (light color) and SPL (dark color) on R2R val unseen split, BEVBert vs. EnvEdit [50] SR and SDTW on RxR val unseen split).



Instruction: *Walk straight past the table and turn right to go between the kitchen counters and walk straight past the refrigerator into the pantry and stop halfway between the two shelves on the right.*

Instruction: *Walk behind the couch towards the kitchen. Enter the kitchen. Walk towards the doors that go outside. Turn left when you reach the doors. Walk down the hallway past the kitchen. Stop where the four hallways intersect.*

Figure 7. Predictions of DUET [15] and BEVBert on R2R-unseen. Yellow and green circles denote the start and target locations, respectively, and the red circles represent incorrect endpoints.

5. Conclusion

In this paper, we present a novel mapping-based vision language pre-training model, BEVBert, to improve an embodied agent's spatial perception and reasoning capacity. We further propose a topology-guided metric mapping and a masked semantic imagination scheme to mitigate the partial observability issue. Extensive experiments demonstrate the effectiveness of the proposed method, and BEVBert sets the new state-of-the-art.

Limitations and future works. BEVBert relies on the cross-attention and self-attention operation on the entire metric map, which is computationally expensive. Designing more efficient operators for map reasoning is the future direction. We hope this work can encourage the VLN community to increase research on semantic mapping in the future.

[3] https://eval.ai/web/challenges/challenge-page/606/leaderboard/1683

References

[1] Dong An, Yuankai Qi, Yan Huang, Qi Wu, Liang Wang, and Tieniu Tan. Neighbor-view enhanced model for vision and language navigation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5101–5109, 2021. 2, 7, 16

[2] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 5

[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018. 2

[4] Peter Anderson, Ayush Shrivastava, Devi Parikh, Dhruv Batra, and Stefan Lee. Chasing ghosts: Instruction following as bayesian state tracking. *Advances in neural information processing systems*, 32, 2019. 1, 2, 3, 7, 16

[5] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018. 1, 2, 5, 7, 13, 14, 15, 16, 17

[6] 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. 1, 2

[7] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 964–972, 2021. 1

[8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE, 2017. 5, 13, 15

[9] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations*, 2019. 2

[10] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020. 1, 2

[11] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12875–12884, 2020. 1, 2

[12] Jinyu Chen, Chen Gao, Erli Meng, Qiong Zhang, and Si Liu. Reinforced structured state-evolution for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15450–15459, 2022. 7, 16

[13] Kevin Chen, Junshen K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11276–11286, 2021. 1, 2

[14] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34:5834–5847, 2021. 2, 5, 7, 8, 14, 16, 17

[15] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16537–16547, 2022. 1, 2, 5, 6, 7, 8, 13, 15, 16, 17

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7, 14

[17] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 33:20660–20672, 2020. 2

[18] 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 *International Conference on Learning Representations*, 2020. 3, 4

[19] 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. *Advances in Neural Information Processing Systems*, 31, 2018. 2, 7, 16

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

[21] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015. 2

[22] Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, and Alessandro Saffiotti. Robot task planning using semantic maps. *Robotics and autonomous systems*, 56(11):955–966, 2008. 2

[23] Chen Gao, Jinyu Chen, Si Liu, Luting Wang, Qiong Zhang, and Qi Wu. Room-and-object aware knowledge reasoning for remote embodied referring expression. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3064–3073, 2021. 2

[24] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15460–15470, 2022. 1, 2

[25] 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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9673–9679. IEEE, 2020. 1, 2

[26] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7606–7623, 2022. 1, 2

[27] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1634–1643, 2021. 2, 5, 7, 8, 16, 17

[28] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13137–13146, 2020. 2, 5, 6, 15, 16

[29] Keji He, Yan Huang, Qi Wu, Jianhua Yang, Dong An, Shuanglin Sima, and Liang Wang. Landmark-rxr: Solving vision-and-language navigation with fine-grained alignment supervision. *Advances in Neural Information Processing Systems*, 34:652–663, 2021. 2

[30] Joao F Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8476–8484, 2018. 3

[31] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. *Advances in Neural Information Processing Systems*, 33:7685–7696, 2020. 2

[32] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1643–1653, 2021. 7, 8, 13, 14, 16, 17

[33] Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. Are you looking? grounding to multiple modalities in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6551–6557, 2019. 2

[34] Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. Seeing out of the box: End-to-end pre-training for vision-language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12976–12985, 2021. 2

[35] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020. 2

[36] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019. 5

[37] Muhammad Zubair Irshad, Niluthpol Chowdhury Mithun, Zachary Seymour, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Sasra: Semantically-aware spatio-temporal reasoning agent for vision-and-language navigation in continuous environments. *arXiv preprint arXiv:2108.11945*, 2021. 1, 2

[38] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276, 2020. 2

[39] Jindong Jiang, Lunan Zheng, Fei Luo, and Zhijun Zhang. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. *arXiv preprint arXiv:1806.01054*, 2018. 6, 15

[40] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. 4, 5

[41] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021. 2

[42] Jing Yu Koh, Harsh Agrawal, Dhruv Batra, Richard Tucker, Austin Waters, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Simple and effective synthesis of indoor 3d scenes. 6

[43] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14738–14748, 2021. 6

[44] Kurt Konolige, Eitan Marder-Eppstein, and Bhaskara Marthi. Navigation in hybrid metric-topological maps. In *2011 IEEE International Conference on Robotics and Automation*, pages 3041–3047. IEEE, 2011. 1, 2

[45] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020. 2

[46] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 2

[47] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4392–4412, 2020. 2, 5, 7, 13, 14, 17

[48] Obin Kwon, Nuri Kim, Yunho Choi, Hwiyeon Yoo, Jeongho Park, and Songhwai Oh. Visual graph memory with unsupervised representation for visual navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15890–15899, 2021. 2

[49] Jialu Li, Hao Tan, and Mohit Bansal. Clear: Improving vision-language navigation with cross-lingual, environment-agnostic representations. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 633–649, 2022. 7, 17

[50] Jialu Li, Hao Tan, and Mohit Bansal. Envedit: Environment editing for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15407–15417, 2022. 2, 7, 8, 17

[51] 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 *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 2

[52] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *International Conference on Learning Representations*, 2021. 2

[53] Xiangru Lin, Guanbin Li, and Yizhou Yu. Scene-intuitive agent for remote embodied visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7036–7045, 2021. 13, 14

[54] Chong Liu, Fengda Zhu, Xiaojun Chang, Xiaodan Liang, Zongyuan Ge, and Yi-Dong Shen. Vision-language navigation with random environmental mixup. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1644–1654, 2021. 2, 7, 16

[55] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 6

[56] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 15

[57] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 2, 14

[58] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. *arXiv preprint arXiv:1901.03035*, 2019. 2, 7, 16, 17

[59] 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 *European Conference on Computer Vision*, pages 259–274. Springer, 2020. 2, 5

[60] Abhinav Moudgil, Arjun Majumdar, Harsh Agrawal, Stefan Lee, and Dhruv Batra. Soat: A scene-and object-aware transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34:7357–7367, 2021. 2

[61] Shun Niijima, Ryusuke Umeyama, Yoko Sasaki, and Hiroshi Mizoguchi. City-scale grid-topological hybrid maps for autonomous mobile robot navigation in urban area. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2065–2071. IEEE, 2020. 1, 2

[62] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Javen Qinfeng Shi, and Anton van den Hengel. Counterfactual vision-and-language navigation: Unravelling the unseen. *Advances in Neural Information Processing Systems*, 33:5296–5307, 2020. 2

[63] Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15942–15952, 2021. 2

[64] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pages 194–210. Springer, 2020. 3

[65] Yuankai Qi, Zizheng Pan, Yicong Hong, Ming-Hsuan Yang, Anton van den Hengel, and Qi Wu. The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1655–1664, 2021. 2

[66] Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. Object-and-action aware model for visual language navigation. In *European Conference on Computer Vision*, pages 303–317. Springer, 2020. 2, 16

[67] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991, 2020. 1, 2, 5, 8, 13, 14, 17

[68] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15418–15427, 2022. 2

[69] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 6, 7

[70] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[71] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform

for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 6

[72] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? In *International Conference on Learning Representations*, 2021. 7, 17

[73] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, 2019. 2, 6

[74] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of NAACL-HLT*, pages 2610–2621, 2019. 2, 7, 16

[75] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15471–15481, 2022. 2

[76] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8455–8464, 2021. 2, 7, 16

[77] Hanqing Wang, Wenguan Wang, Tianmin Shu, Wei Liang, and Jianbing Shen. Active visual information gathering for vision-language navigation. In *European Conference on Computer Vision*, pages 307–322. Springer, 2020. 16

[78] Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin Waters, Jason Baldridge, and Peter Anderson. Less is more: Generating grounded navigation instructions from landmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15428–15438, 2022. 2, 6, 7, 8, 15, 17

[79] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019. 2, 7, 16, 17

[80] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 37–53, 2018. 2

[81] Xin Eric Wang, Vihan Jain, Eugene Ie, William Yang Wang, Zornitsa Kozareva, and Sujith Ravi. Environment-agnostic multitask learning for natural language grounded navigation. In *European Conference on Computer Vision*, pages 413–430. Springer, 2020. 2

[82] Siying Wu, Xueyang Fu, Feng Wu, and Zheng-Jun Zha. Cross-modal semantic alignment pre-training for vision-and-language navigation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4233–4241, 2022. 2

[83] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *arXiv preprint arXiv:2111.08276*, 2021. 2

[84] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021. 2

[85] Yue Zhang and Parisa Kordjamshidi. Lovis: Learning orientation and visual signals for vision and language navigation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5745–5754, 2022. 2

[86] 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 *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4194–4203, 2022. 7, 8, 16, 17

[87] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10012–10022, 2020. 7, 16

[88] Wanrong Zhu, Yuankai Qi, P. Narayana, Kazoo Sone, Sugato Basu, Xin Eric Wang, Qi Wu, Miguel P. Eckstein, and William Yang Wang. Diagnosing vision-and-language navigation: What really matters. In *NAACL*, 2022. 2

Appendices

Section A presents more details about evaluation datasets and metrics. Model variants and training objectives are described in Section B. Details about experimental setups are provided in Section C, and more comparisons against state-of-the-art methods are shown in Section D. Finally, we present several visualizations of failure cases in Section E.

A. Evaluation Datasets and Metrics

Our approach is evaluated on R2R [5], RxR [47] and REVERIE [67] datasets, which are built upon the Matterport3D [8] indoor scene dataset. We summarize the dataset statistics in Tab. 9. The three datasets differ in task type (instruction-following v.s. goal-oriented) and annotation granularity of instructions (low-level v.s. high-level).

Room-to-Room (R2R) [5] is an instruction-following dataset that provides low-level (step-by-step) instructions. The agent is required to follow an instruction to reach the target location. The agent receives panoramic observations at each viewpoint, and the navigation is simplified as transporting among viewpoints predefined on the connectivity graph of an environment. The dataset contains 61 houses for training, 56 houses for validation in seen environments, 11 and 18 houses for validation and testing in unseen environments, respectively. Each instruction in R2R describes a full path, such as "*Head straight until you pass the wall with holes in it the turn left and wait by the glass table with the white chairs.*".

Room-across-Room (RxR) [47] is a more challenging instruction-following dataset, which emphasizes the role of language guidance and provides large-scale multilingual instructions (en-IN, en-US, hi-IN, te-IN). Instructions in RxR involve more descriptions about visual entities and relations, such as "*You are facing towards a wall, turn around and move forward. You are now standing in front of a glass cabin and on your right side you have a table with a computer. You can see a table with a black chair right in front of you. Walk between these two tables and move forward. Now you can see a table with orange chair on your right side. Move forward towards the centre table which is right in front of a couch and that is your end point.*". Besides, annotated paths in RxR are much longer than R2R ($\sim$ 3 times) and the ground truth paths are not the shortest path between the starting and ending points.

REVERIE [67] is a goal-oriented dataset which provides high-level instructions. Instructions in REVERIE are concise and mainly describe target rooms and target objects, such as "*Go to the office and clean the black and white picture of a child*". The task focuses more on knowledge and commonsense exploitation rather than instruction following. Furthermore, the agent must identify the target object from a set of candidates after reaching the desired location. The dataset has 4,140 target objects in 489 categories, and each target viewpoint has 7 objects on average.

Evaluation Metrics. VLN tasks mainly focus on the agent's generalization ability in unseen environments (val unseen and test unseen splits). The main evaluation metrics of the above datasets are slightly different. On the **R2R** dataset, SR and SPL are the main metrics to evaluate the navigation accuracy and efficiency, where a predicted path is regarded as *success* if the agent stops within 3 meters of the target viewpoint. **RxR** dataset does not have shortest-path prior, and it additionally takes NDTW and SDTW to measure the fidelity between predicted and annotated paths. The two metrics reflect how well the agent interprets and follows instructions. On the **REVERIE** dataset, annotated paths have shortest-path prior and the main navigation metrics are SR and SPL. A predicted path is considered as *success* if the target object is visible within 3 meters at the endpoint. It additionally uses RGS and RGSPL to evaluate the object grounding capacity of the agent, an grounding is *success* if the predicted and annotated objects are the same.

B. Model Details

B.1. Adaptation to the REVERIE Task

REVERIE task provides candidate object annotations at each step and requires the agent to point out the target object when it stops. As shown in Fig. 8, we feed these candidates into the short-term branch (metric map encoder) to enable object grounding. Specifically, at each step, we first use the same ViT as in Section 3.1 to extract object features $\mathbf{O}_t = \{\mathbf{o}_z | \mathbf{o}_z \in \mathcal{R}^D\}_{z=1}^Z$. After adding with position embeddings (sine and cosine of orientations [15, 32, 53]), these object features are concatenated with view features $\mathbf{V}_t$ and fed into the pano encoder to obtain contextual representations $\hat{\mathbf{O}}_t$:

$$[\hat{\mathbf{V}}_t^p; \hat{\mathbf{O}}_t] = \text{PanoEncoder}([\mathbf{V}_t^p; \mathbf{O}_t]) \tag{11}$$

where the pano encoder is the same one used in Section 3.1. Then, cell and object embeddings are concatenated as the visual modality while encoded instructions as the linguistic modality. We feed them into the short-term transformer to perform cross-modal reasoning as explained in Section 3.2.3. The output language-informed object representations $\tilde{\mathbf{O}}_t = \{\tilde{o}_z\}_{z=1}^Z$ are learned via MRC and OG tasks, which will be detailed in next section.

B.2. Pre-training Objectives

R2R and RxR. We sample pre-training tasks for each mini-batch to train the BEVBert model. The sampling ratio for R2R and RxR datasets is HMLM : HSAP : MSI $= 5 : 5 : 1$. We randomly chunk a sampled expert trajectory $\mathbf{\Gamma}$ from head to learn intermediate map representations of the trajectory.

REVERIE. Instructions in REVERIE mainly describe the rooms and target objects at endpoints. We do not employ MSI task due to the lack of intermediate path descriptions;

Task Type	Granularity	Dataset	Train #house	Train #instr	Val Seen #house	Val Seen #instr	Val Unseen #house	Val Unseen #instr	Test Unseen #house	Test Unseen #instr
Instruction-following	Low-level	R2R [5]	61	14,039	56	1,021	11	2,349	18	4,173
		RxR [47]	60	79,467	58	8,813	11	13,652	17	12,249
Goal-oriented	High-level	REVERIE [67]	60	10,466	46	1,423	10	3,521	16	6,292

Table 9. Dataset statistics. #house, #instr denote the number of houses and instructions respectively.

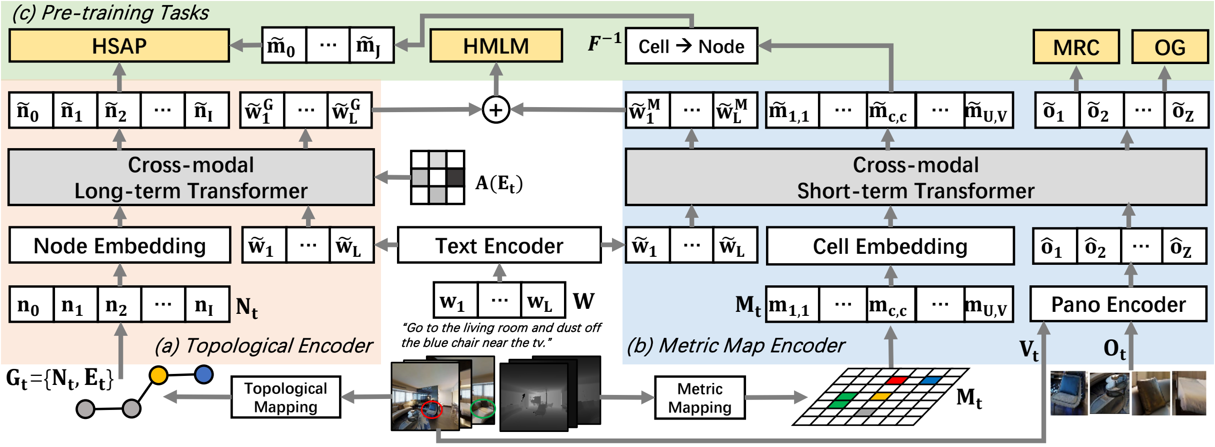

(c) Pre-training Tasks
HSAP
$\tilde{m}_0$ ··· $\tilde{m}_J$
HMLM
F^{-1}
Cell → Node
MRC
OG
$\tilde{n}_0$ $\tilde{n}_1$ $\tilde{n}_2$ ··· $\tilde{n}_I$
$\tilde{w}_1^G$ ··· $\tilde{w}_L^G$
$\tilde{w}_1^M$ ··· $\tilde{w}_L^M$
$\tilde{m}_{1,1}$ ··· $\tilde{m}_{c,c}$ ··· $\tilde{m}_{U,V}$
$\tilde{o}_1$ $\tilde{o}_2$ ··· $\tilde{o}_Z$
Cross-modal Long-term Transformer
$A(E_t)$
Cross-modal Short-term Transformer
Node Embedding
$\tilde{w}_1$ ··· $\tilde{w}_L$
Text Encoder
Cell Embedding
$\hat{o}_1$ $\hat{o}_2$ ··· $\hat{o}_Z$
n_0 n_1 n_2 ··· n_I N_t
w_1 ··· w_L W
M_t $m_{1,1}$ ··· $m_{c,c}$ ··· $m_{U,V}$
Pano Encoder
"Go to the living room and dust off the blue chair near the tv."
(a) Topological Encoder
(b) Metric Map Encoder
V_t
O_t
$G_t=\{N_t, E_t\}$
Topological Mapping
Metric Mapping
M_t

Figure 8. Adapt the proposed pre-training model to the REVERIE task. Additional candidate objects are fed into the metric map encoder to obtain language-informed object representations $\tilde{\mathbf{o}}_*$. We use MRC and OG tasks to learn cross-modal object reasoning and grounding.

instead, Masked Region Classification (MRC) [57] and Object Grounding (OG) [53] are used for final object reasoning and grounding. **MRC** aims to predict semantic labels of masked objects by reasoning over the surrounding objects and the instructions. We randomly mask objects with a 15% probability and feed them into the metric map encoder. The semantic labels of objects are class probability predicted by a ViT pre-trained on ImageNet [16]. The task is optimized by minimizing the KL divergence between the predicted and target probability distribution. **OG** is a downstream-specific task. After obtaining the language-informed object representation $\tilde{\mathbf{O}}_t$, a two-layer feed-forward network is employed to predict the target object logits. Given the target object label $\mathbf{o}^*$ at step T (the endpoint), the task is optimized by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{OG}} = -\log \mathcal{P}_\theta(\mathbf{o}^* | \mathbf{W}, \mathbf{\Gamma}_T) \tag{12}$$

We sample pre-training tasks for each mini-batch to train the model on REVERIE dataset, and the sampling ratio is HMLM : HSAP : MRC : OG = 1 : 1 : 1 : 1.

B.3. Layer Variants in Fine-tuning

During fine-tuning, the computational graph gradually accumulates as the trajectory rolls out and may lead to GPU being out of memory in extreme cases. To alleviate the problem (see Fig. 9), we turn off the language-to-vision cross-attention and language-to-language self-attention in each layer. All transformer layers share the same text representations. We empirically found it does not severely hamper navigation performance, and the same phenomenon is also observed in RecBert [32] and HAMT [14].

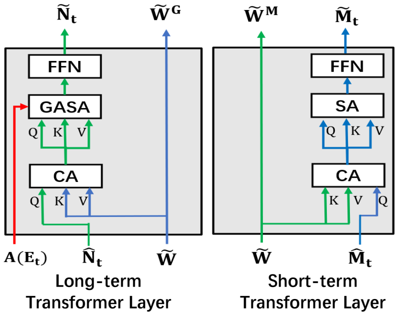


Figure 9. Layer variants in fine-tuning. Self-attention and cross-attention of the text branch are cut off for computation efficiency.

B.4. Fine-tuning Objectives

During fine-tuning, we alternatively run 'teacher-forcing' $\mathcal{L}_{\text{TF}}$ and 'student-forcing' $\mathcal{L}_{\text{SF}}$. The model is optimized by the mixed loss $\mathcal{L}$, formally:

$$\begin{aligned} \mathcal{L}_{\text{TF}} &= -\sum\nolimits_{t=1}^{T} \log \mathcal{P}_\theta(\mathbf{a}_t^* | \mathbf{W}, \mathbf{\Gamma}_{<t}) \\ \mathcal{L}_{\text{SF}} &= -\sum\nolimits_{t=1}^{T} \log \mathcal{P}_\theta(\mathbf{a}_t^{G*} | \mathbf{W}, \tilde{\mathbf{\Gamma}}_{<t}) \\ \mathcal{L} &= \lambda \cdot \mathcal{L}_{\text{TF}} + \mathcal{L}_{\text{SF}} \end{aligned} \tag{13}$$

In $\mathcal{L}_{\text{TF}}$, the agent executes ground-truth actions $\mathbf{a}_t^*$ to follow the expert trajectory $\mathbf{\Gamma}$, and the loss is the accumulation of negative log-likelihoods of these expert actions. In $\mathcal{L}_{\text{SF}}$, the agent generates trajectory $\tilde{\mathbf{\Gamma}}$ by on-policy action sampling

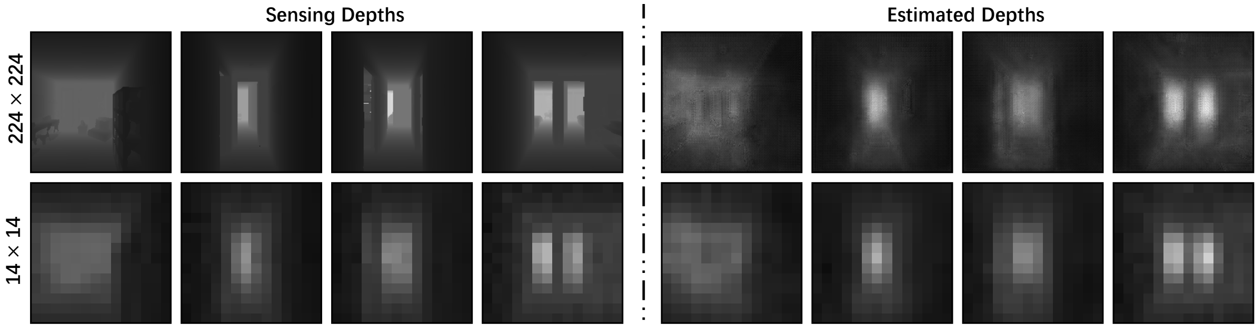


Figure 10. Qualitative visualization of the sensing and estimated depths. The top row represents depths of the original scale, and the bottom row presents downsized depths that have the same scale with grid features.

and supervised by pseudo labels $\mathbf{a}_t^{G*}$. Specifically, the action is sampled via the predicted action probability distribution at each step, and the pseudo labels are determined via goal-oriented or fidelity-oriented heuristics. A goal-oriented label is an observed node that has the shortest path length to the final target location, while a fidelity-oriented label is an observed node through which the sampled path has the highest fidelity (NDTW) with the expert path. We use goal-oriented labels for R2R and REVERIE, while fidelity-oriented labels for RxR because it does not have shortest-path prior. Besides, OG loss is added for fine-tuning on REVERIE:

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{\text{TF}} + \mathcal{L}_{\text{SF}} + \mathcal{L}_{\text{OG}} \tag{14}$$

The 'teacher-forcing' encourages the agent to follow the expert path, while the 'student-forcing' encourages the agent to explore the environment. We set balanced term $\lambda = 0.2$ on R2R and REVERIE datasets, while $\lambda = 0.8$ on the RxR dataset. Because annotated paths in RxR are much longer, a small λ can lead to unnecessary exploration and hamper navigation fidelity.

C. Experimental Setups

C.1. Training details

During offline pre-training, we train BEVBert with batch size 64 for 100k iterations using 4 NVIDIA Tesla A100 GPUs (∼10 hours). The optimizer is AdamW [56], and a linear warm-up schedule is adopted to raise the learning rate to 5×10^{-5} at the first 10k iterations, and then gradually decays for the left iterations. During fine-tuning, we use the pre-trained model with the best zero-shot performance as initialization and set the learning rate as 1×10^{-5}. The model is trained by online interaction in the Matterport3D Simulator [5] with batch size 16 for 40k iterations using 4 NVIDIA Tesla A100 GPUs (∼20 hours). The best iterations are selected by best performance on validation unseen splits. We adopt Prevalent [28] and RxR-Markey [78] synthetic instructions as data augmentation in both pre-training and fine-tuning stages of R2R and RxR, respectively. The REVERIE-Spk [15] augmentation is only used in pre-training on REVERIE. We found fine-tuning with REVERIE-Spk leads to overfitting in seen environments and affect the agent's generalization ability.

C.2. Depth Estimation

This section details the depth ablation study in Section 4.2 (2). We utilize RedNet [39] for depth estimation, which takes RGB images as inputs and uses a U-Net-like architecture for depth regression. The estimated depths are outputs of a sigmoid layer, and downsized depth images also supervise intermediate layers to speed up convergence. We train the model in train-split houses of Matterport3D dataset [8]. Then, a trained model is used to estimate depths for all viewpoints of all houses, and we retrain BEVBert with these pseudo depths. We visualize some sensing and estimated depths in Fig. 10. Intuitively, estimated depths are of low quality and have noise. But after downsampling, the noise reduces, and they have similar quality with sensing depths. It also indicates that our metric mapping does not rely on very accurate depth images.

C.3. Spatial and Numerical Tokens

In Tab. 11, We summarize the token templates we used to extract spatial and numerical related instructions used for evaluation in Section 4.4. Extracted instructions are grouped via the number of special tokens in each instruction. To ensure reliable performance estimation, we omit those groups which contain less than 40 instructions in Fig. 6.

D. More Comparisons with State-of-the-Art

In Tab. 10, Tab. 12 and Tab. 13, we present more comparisons with state-of-the-art methods on R2R, RxR and REVERIE, respectively. The main metrics of each dataset are highlighted in grey. BEVBert also achieves state-of-the-art performance in seen splits on all metrics, but the performance is still far behind humans'. For example, on the test unseen split of RxR dataset, humans can achieve 93.9% SR and 76.9% SDTW, while BEVBert has 64.4% SR and 54.2% SDTW.

	Val Seen					Val Unseen					Test Unseen				
Methods	TL	NE↓	OSR↑	SR↑	SPL↑	TL	NE↓	OSR↑	SR↑	SPL↑	TL	NE↓	OSR↑	SR↑	SPL↑
Human	-	-	-	-	-	-	-	-	-	-	11.85	1.61	90	86	76
Seq2Seq [5]	11.33	6.01	53	39	-	8.39	7.81	28	21	-	8.13	7.85	27	20	-
SF [19]	-	3.36	74	66	-	-	6.62	45	36	-	14.82	6.62	-	35	28
Chasing [4]	10.15	7.59	42	34	30	9.64	7.20	44	35	31	10.03	7.83	42	33	30
RCM [79]	10.65	3.53	75	67	-	11.46	6.09	50	43	-	11.97	6.12	50	43	38
SM [58]	-	3.22	78	67	58	-	5.52	56	45	32	18.04	5.67	59	48	35
EnvDrop [74]	11.00	3.99	-	62	59	10.70	5.22	-	52	48	11.66	5.23	59	51	47
OAAM [66]	-	-	73	65	62	-	-	61	54	50	-	-	61	53	50
AuxRN [87]	-	3.33	78	70	67	-	5.28	62	55	50	-	5.15	62	55	51
Active [77]	-	3.20	80	70	52	-	4.36	70	58	40	-	4.33	71	60	41
PREVALENT [28]	10.32	3.67	-	69	65	10.19	4.71	-	58	53	10.51	5.30	61	54	51
NvEM [1]	11.09	3.44	-	69	65	11.83	4.27	-	60	55	12.98	4.37	66	58	54
SSM [76]	14.70	3.10	80	71	62	20.70	4.32	73	62	45	20.40	4.57	70	61	46
AirBert [27]	11.09	2.68	-	75	70	11.78	4.10	-	62	56	12.41	4.13	-	62	57
SEvol [12]	11.97	3.56	-	67	63	12.26	3.99	-	62	57	13.40	4.13	-	62	57
RecBert [32]	11.13	2.90	-	72	68	12.01	3.93	-	63	57	12.35	4.09	70	63	57
EnvMix [54]	10.88	2.48	-	75	72	12.44	3.89	-	64	58	13.11	3.87	72	65	59
HAMT [14]	11.15	2.51	-	76	72	11.46	3.65	-	66	61	12.27	3.93	72	65	60
TD-STP [86]	-	2.34	83	77	73	-	3.22	76	70	63	-	3.73	72	67	61
DUET [15]	12.32	2.28	86	79	73	13.94	3.31	81	72	60	14.73	3.65	76	69	59
BEVBert (Ours)	13.56	2.17	88	81	74	14.55	2.81	84	75	64	15.87	3.13	81	73	62

Table 10. Comparison with state-of-the-art methods on R2R dataset.

Token Type	**Token Templates**
Spatial	on the left, on your left, to the left, to your left left of, left side of, leftmost, on the right, on your right, to the right, to your right, right of, right side of, rightmost, near, nearest, behind, between, next to, end of, edge of, front of, middle of, top of, bottom of
Numerical	first, second, third, fourth, fifth, sixth, seventh, eighth, one, two, three, four, five, six, seven, eight, 1, 2, 3, 4, 5, 6, 7, 8

Table 11. Templates of spatial and numerical tokens.

E. More Qualitative Examples

We visualize some failure cases in Fig. 11 and Fig. 12. We attribute the failure reasons to ‘early lost’ and ‘ambiguity’.

Fig. 11 presents four ‘early lost’ cases. The agent loses the state tracking of navigation due to early mistakes, leading to too much backtracking in cases (a,b,c). However, it does not go back to the right path till the end. In case (d), the agent does not “turn left” after “into the hallway”. This does not trigger backtracking, but the agent directly “wait by the kitchen counter” at the wrong location after seeing a counter.

Fig. 12 shows four ‘ambiguity’ cases. Some ambiguous instructions may confuse the agent, such as “enter another bedroom straight ahead” in case (a) and “enter the second room on the left” in case (b). In case (c), the agent “walk to the end of the entrance way” in the opposite direction. After reaching the hallway end, it has lost state tracking and cannot backtrack. In case (d), the agent does not know whether it has finished “down the hallway” or not, then makes an early “turn right” and stops in advance.

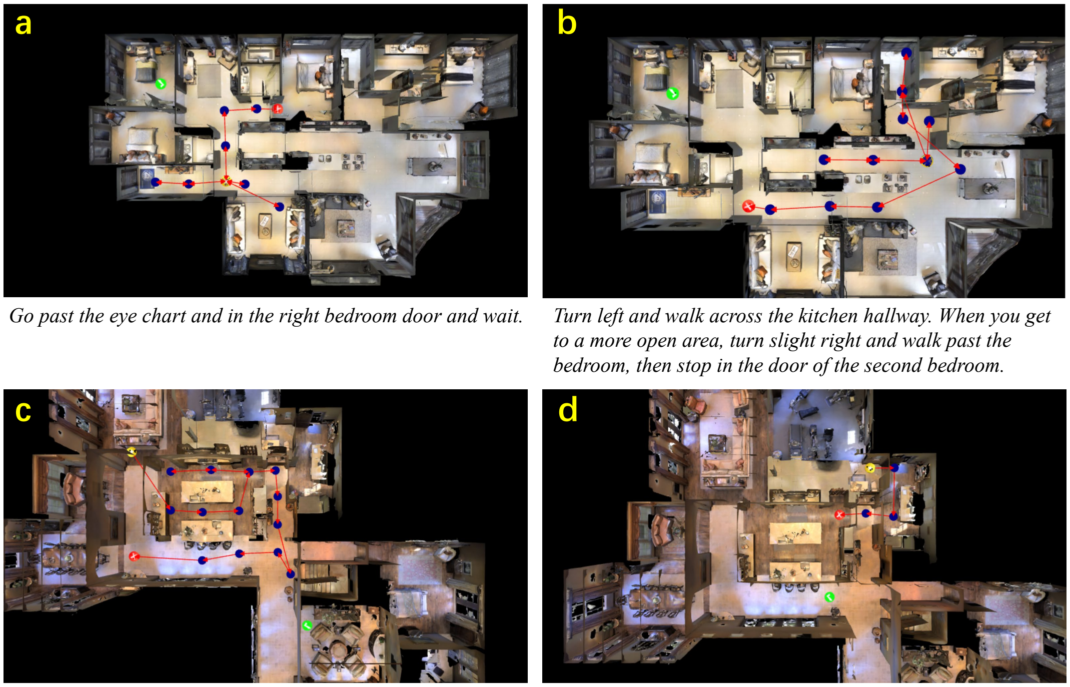
Methods	Val Seen				Val Unseen				Test Unseen			
	NE↓	SR↑	NDTW↑	SDTW↑	NE↓	SR↑	NDTW↑	SDTW↑	NE↓	SR↑	NDTW↑	SDTW↑
Human	-	-	-	-	-	-	-	-	0.9	93.9	79.5	76.9
LSTM [47]	10.7	25.2	42.2	20.7	10.9	22.8	38.9	18.2	12.0	21.0	36.8	16.9
EnvDrop+ [72]	-	-	-	-	-	42.6	55.7	-	-	38.3	51.1	32.4
CLEAR-C [49]	-	-	-	-	-	-	-	-	-	40.3	53.7	34.9
HAMT [14]	-	59.4	65.3	50.9	-	56.5	63.1	48.3	6.2	53.1	59.9	45.2
EnvEdit* [50]	-	67.2	71.1	58.5	-	62.8	68.5	54.6	5.1	60.4	64.6	51.8
BEVBert†(Ours)	3.8	68.9	70.0	58.4	4.6	64.1	63.9	52.6	-	-	-	-
BEVBert (Ours)	3.2	75.0	76.3	66.7	4.0	68.5	69.6	58.6	4.8	64.4	65.4	54.2

*Results from an ensemble of three agents. †Results without Marky synthetic instructions [78].

Table 12. Comparison with state-of-the-art methods on RxR dataset.

Methods	Val seen					Val Unseen					Test Unseen				
	Navigation			Grounding		Navigation			Grounding		Navigation			Grounding	
	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑
Human	-	-	-	-	-	-	-	-	-	-	81.51	53.66	86.83	77.84	51.44
Seq2Seq [5]	35.70	29.59	24.01	18.97	14.96	8.07	4.20	2.84	2.16	1.63	6.88	3.99	3.09	2.00	1.58
RCM [79]	29.44	23.33	21.82	16.23	15.36	14.23	9.29	6.97	4.89	3.89	11.68	7.84	6.67	3.67	3.14
SMNA [58]	43.29	41.25	39.61	30.07	28.98	11.28	8.15	6.44	4.54	3.61	8.39	5.80	4.53	3.10	2.39
FAST-MATTN [67]	55.17	50.53	45.50	31.97	29.66	28.20	14.40	7.19	7.84	4.67	30.63	19.88	11.6	11.28	6.08
SIA [53]	65.85	61.91	57.08	45.96	42.65	44.67	31.53	16.28	22.41	11.56	44.56	30.80	14.85	19.02	9.20
RecBERT [32]	53.90	51.79	47.96	38.23	35.61	35.20	30.67	24.90	18.77	15.27	32.91	29.61	23.99	16.50	13.51
AirBert [27]	48.98	47.01	42.34	32.75	30.01	34.51	27.89	21.88	18.23	14.18	34.20	30.26	23.61	16.83	13.28
HAMT [14]	47.65	43.29	40.19	27.20	25.18	36.84	32.95	30.20	18.92	17.28	33.41	30.40	26.67	14.88	13.08
TD-STP [86]	-	-	-	-	-	39.48	34.88	27.32	21.16	16.56	40.26	35.89	27.51	19.88	15.40
DUET [15]	73.86	71.75	63.94	57.41	51.14	51.07	46.98	33.73	32.15	23.03	56.91	52.51	36.06	31.88	22.06
BEVBert (Ours)	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

Table 13. Comparison with state-of-the-art methods on REVERIE dataset.



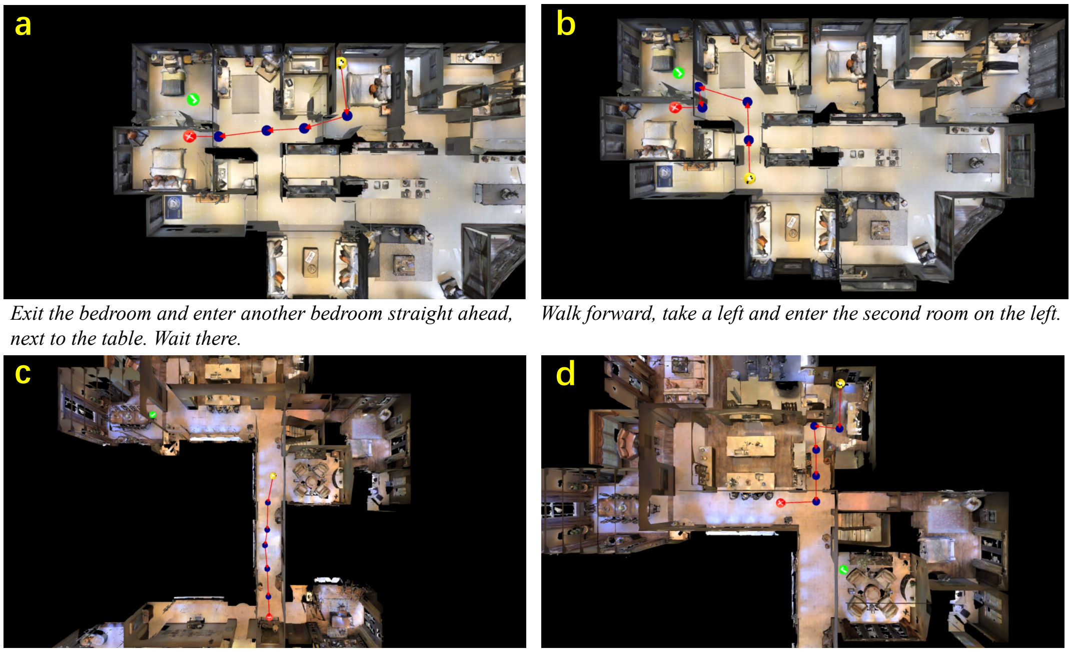
Go past the eye chart and in the right bedroom door and wait.

Turn left and walk across the kitchen hallway. When you get to a more open area, turn slight right and walk past the bedroom, then stop in the door of the second bedroom.

Veer left to walk through the kitchen, then turn left. Make the first right and then left into the living room then wait by the console table.

Exit the bathroom through the doorway by the toilet, then turn right, Walk past the counter into the hallway, turn left. At the end of the hallway turn right, then wait by the kitchen counter.

Figure 11. Failure cases of 'early lost' in val unseen splits of R2R. Yellow and green circles denote the start and target locations, respectively, and the red circles represent incorrect endpoints.



Exit the bedroom and enter another bedroom straight ahead, next to the table. Wait there.

Walk forward, take a left and enter the second room on the left.

Walk to the end of the entrance way and turn left. Travel across the kitchen area with the counter and chairs on your right. Continue straight until you reach the dining room.

turn right and walk past wood table, turn right at the doorway, turn left down the hallway, turn right and stop behind plush chair.

Figure 12. Failure cases of ‘ambiguity’ in val unseen splits of R2R. Yellow and green circles denote the start and target locations, respectively, and the red circles represent incorrect endpoints.