BEVBert: Multimodal Map Pre-training for Language-guided Navigation

Dong An[1,2] Yuankai Qi[3] Yangguang Li[4] Yan Huang[1†] Liang Wang[1] Tieniu Tan[1,5] Jing Shao[6†]
[1]Institute of Automation, Chinese Academy of Sciences [2]School of Future Technology, UCAS
[3]Australian Institute for Machine Learning, University of Adelaide [4]SenseTime Research
[5]Nanjing University [6]Shanghai AI Laboratory

https://github.com/MarSaKi/VLN-BEVBert

Abstract

Large-scale pre-training has shown promising results on the vision-and-language navigation (VLN) task. However, most existing pre-training methods employ discrete panoramas to learn visual-textual associations. This requires the model to implicitly correlate incomplete, duplicate observations within the panoramas, which may impair an agent's spatial understanding. Thus, we propose a new map-based pre-training paradigm that is spatial-aware for use in VLN. Concretely, we build a local metric map to explicitly aggregate incomplete observations and remove duplicates, while modeling navigation dependency in a global topological map. This hybrid design can balance the demand of VLN for both short-term reasoning and long-term planning. Then, based on the hybrid map, we devise a pre-training framework to learn a multimodal map representation, which enhances spatial-aware cross-modal reasoning thereby facilitating the language-guided navigation goal. Extensive experiments demonstrate the effectiveness of the map-based pre-training route for VLN, and the proposed method achieves state-of-the-art on four VLN benchmarks.

1. Introduction

Interaction with an assistant 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 [1, 2, 3]. Given a natural language instruction, a VLN agent is required to interpret and follow the instruction to reach the desired location. Enhancing the learning of visual-textual association is essential for the agent to succeed. Inspired by the great success of vision-language pre-training [4, 5, 6, 7, 8, 9], a variety of VLN pre-training methods have been studied and achieved promising results [10, 11, 12, 13, 14].

However, most existing VLN pre-training models resort to discrete panoramas (Fig. 1 (a)) as visual inputs, which

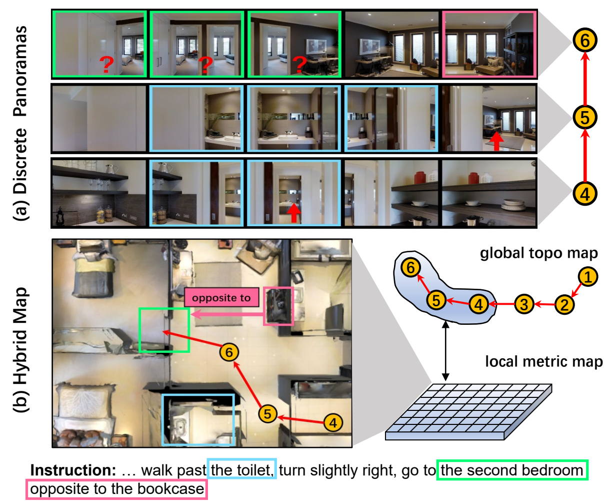

Figure 1: (a) Incomplete observations within a single view and duplicates across views may confuse the agent. (b) Projecting discrete panoramas into a unified map can solve the problem, thus facilitating spatial reasoning.

require the model to implicitly correlate incomplete, duplicate observations across views of the panoramas. This may hamper the agent's cross-modal spatial reasoning ability. As shown in Fig. 1 (a), it is difficult to infer "the second bedroom opposite to the bookcase" because there are duplicate images of "bedroom" and "bookcase" across different views, and therefore it is hard to tell they are images for the same object or multiple instances. A potential solution is to project these observations into a unified map, which explicitly aggregates incomplete observations and remove duplicate. Though this scheme has been successful in many navigation scenarios [15, 16, 17], its combination with pre-training remains unstudied, and this paper makes the first exploration.

In embodied navigation, maps generally fall into metric [18, 16] or topological [17, 19]. The metric map uses dense grid features to precisely describe the environment but has inefficiencies of scale [20]. As a result, using a large map to capture the long-horizon navigation dependency can cause

†Corresponding authors.

prohibitive computation [21], especially for the computation-intensive pre-training. Yet, such dependency has been shown crucial for VLN [22, 14]. On the other hand, the topo map can efficiently capture dependency by keeping track of visited locations as a graph structure [17]. It also allows the agent to make efficient long-term goal plans, such as backtracking to a previous location [23, 24]. However, each node in the graph is typically represented by condensed feature vectors, which lack fine-grained information for local spatial reasoning.

In this paper, instead of using a large global metric map, we propose a hybrid approach to balance the above two maps (shown in Fig. 1 (b)). It contains a local metric map for short-term spatial reasoning while conducting overall long-term action plans on a global topo map. This scheme shares similar spirits to classical topo-metric SLAM in robotics [25, 20], but it differs in a learnable multimodal representation. To learn such representation, we propose BEVBert, a novel map-based pre-training paradigm that learns better visual-textual associations in bird's-eye view to aid complex spatial reasoning of VLN agents. BEVBert first constructs offline hybrid maps based on large-scale VLN visual paths. Then, we employ a cross-modal transformer to conduct map-instruction interaction to obtain the multimodal map representation. To learn such representation, in addition to language modeling [26] and action prediction [10], we design a map prediction proxy task. This task learns to encode linguistic and spatial priors to predict the information of unobserved regions, thereby reducing the uncertainty for decision-making. Finally, we fine-tune the model with sequential action prediction and online constructed hybrid maps. Thanks to the learned map representations, our agent learns a more robust navigation policy and achieves state-of-the-art on four VLN benchmarks (R2R, R2R-CE, RxR, REVERIE).

In summary, the contributions of this work are three-fold:

• We explore topo-metric maps in VLN for the first time. The proposed hybrid approach presents an elegant balance between short-term reasoning and long-term planning.

• We propose a novel map-based pre-training paradigm, and empirically demonstrate that the learned map representation can enhance spatial-aware cross-modal reasoning.

• BEVBert achieves state-of-the-art on four VLN benchmarks (*e.g.*, in test-unseen splits, 73 `SR` on R2R dataset, 59 `SR` on R2R-CE dataset, and 54.2 `SDTW` on RxR dataset).

2. Related Work

Vision-and-Language Navigation. VLN has drawn increasing attention in recent years [1, 2, 27, 3, 28, 29, 30, 31]. Early VLN methods use sequence-to-sequence LSTMs to predict low-level actions [1] or high-level actions from discrete panoramas [32]. Different attention mechanisms [33, 34, 35, 36] are proposed to improve cross-modal alignment. Reinforcement learning is also explored to enhance policy learning [37, 38, 39]. To improve an agent's generalization ability to unseen environments, data augmentation strategies have been studied to mimic new environments [39, 40, 41, 42, 43, 44, 45, 46, 47]. Recently, transformer-based models achieve good performance thanks to their powerful ability to learn generic multi-modal representations [10, 11, 12]. This scheme is further extended by recurrent agent state [48, 13], episodic memory [22, 14], or topology memory [49, 24, 50] that significantly improves sequential action predictions. However, the widely used discrete panoramas [32] require implicit spatial modeling and may hamper the learning of generic language-environment correspondence. To address the limitation, we not only propose a multimodal topo-metric map but also devise a map-based pre-training framework.

Visual Representation in Vision-Language Pre-training. Existing approaches for VLP fall into image-based, object-based, and grid-based. Image-based methods [51] extract an overall feature for an image, yet neglect details, thus drawback on fine-grained language grounding. Object-based methods [52, 5] represent an image with dozens of objects identified by external detectors [53, 54]. The challenge is that objects can be redundant and limited in predefined categories. Grid-based methods [55, 56] directly use image grid features for pre-training, thus enabling multi-grained vision-language alignments. Most VLN pre-training are image-based [11, 12, 14], which rely on discrete panoramas. We introduce grid-based methods into VLN through metric maps, where the model can learn via multi-grained room layouts.

Maps for Navigation. Works on navigation have a long tradition of using SLAM [57] to construct metric maps [16, 58]. A metric map uses grid-based visual features to represent the scene layouts precisely but has inefficiencies of scale [20]. To avoid heavy computation, standard practices restrict the map size[21, 59], which can be inadequate for long-term modeling or planning. Therefore, graph-based topo maps are proposed to address the limitation [60, 61, 17, 19]. But the drawback is short-term reasoning within the condensed nodes [24]. In robotics, topo-metric maps are proposed to trade off their strengths [25, 20]. However, most of them are based on non-learning representations and focus on classical robotic tasks. We propose learnable topo-metric maps and explore the application to high-level VLN tasks.

3. Method

The proposed method focuses on improving VLN agents' planning capability with map-based pre-training. For conciseness, we put our technical description in the context of VLN in discrete environments [1], where maps can be derived from a predefined navigation graph. However, this method can also generalize to the task of VLN in continuous

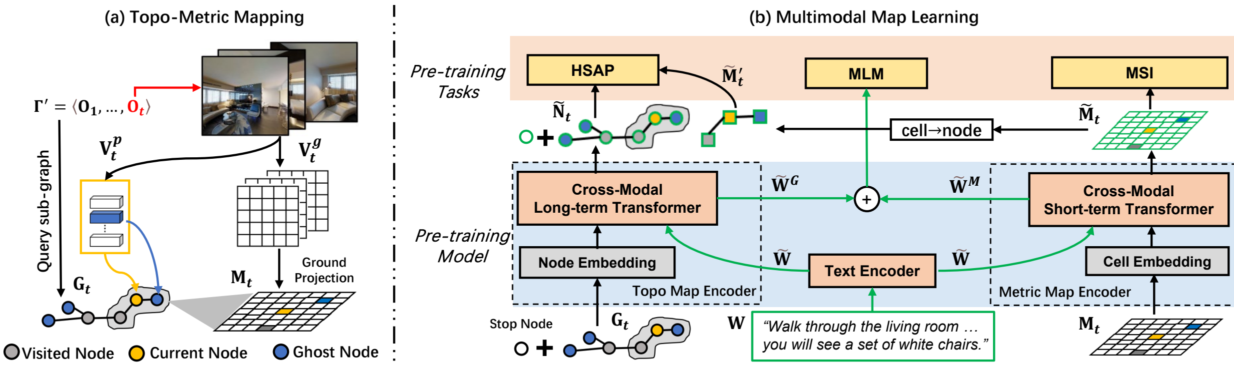

Figure 2: The main architecture of the proposed hybrid-map-based pre-training framework.

environments [27] and more details are presented in § 4.2.

Problem Definition. An agent is required to follow an instruction $\mathbf{W}$ to traverse on a predefined graph $\mathbf{G}^*$ to reach the target location. At each step t, the agent perceives a discrete panorama comprised of RGB images $\mathbf{V}_t$ and depth images $\mathbf{D}_t$. Following [62, 23, 19, 24], we provide the agent with pose information $\mathbf{P}_t$ to simplify the mapping process. With observations $\mathbf{O}_t = \{\mathbf{V}_t, \mathbf{D}_t, \mathbf{P}_t\}$, VLN aims to learn a policy $\pi(\mathbf{a}_t|\mathbf{W}, \mathbf{O}_t)$ to predict action $\mathbf{a}_t$. The action is predicted by selecting a navigable node from a candidate set provided by the simulator. VLN datasets provide annotated instruction-path pairs to learn the policy, *i.e.*, a pair contains an instruction $\mathbf{W}$ with L words, and an expert path $\mathbf{\Gamma} = \langle \mathbf{O}_1, ..., \mathbf{O}_T \rangle$ of length T.

Method Overview. As depicted in Fig. 2, our map-based pre-training framework consists of two modules, namely topo-metric mapping and multimodal map learning. The mapping module constructs an offline hybrid map via a sampled expert path (§ 3.1). The learning module conducts map-instruction interaction (§ 3.2), and then learns multimodal map representations with three pre-training tasks (§ 3.3). After pre-training, the same model is fine-tuned on a sequential action prediction task with online constructed maps (§ 3.4).

3.1. Topo-Metric Mapping

To balance the demand of VLN for long-term planning and short-term reasoning, we propose to construct a hybrid map. As shown in Fig. 2 (a), assuming the agent currently is at step t and the walked path is $\mathbf{\Gamma}'$, we construct a global topo map $\mathbf{G}_t$ and a local metric map $\mathbf{M}_t$. We next introduce how to construct these two maps.

Image Processing. For panoramic RGB images $\mathbf{V}_t$ of each step t, we use a pre-trained vision transformer (ViT) [63] to extract feature vectors $\mathbf{V}_t^p$ and downsized grid features $\mathbf{V}_t^g$. The associated depth images $\mathbf{D}_t$ are downsized to the same scale as $\mathbf{D'}_t$.

Topo Mapping. The graph-based topo map $\mathbf{G}_t = \{\mathbf{N}_t, \mathbf{E}_t\}$ keeps track of all observed nodes along the path $\mathbf{\Gamma}'$. Given $\mathbf{\Gamma}'$, we initialize $\mathbf{G}_t$ by deriving its corresponding sub-graph from the predefined graph $\mathbf{G}^*$. The nodes $\mathbf{N}_t$ are divided into three categories: visited node ⚪, current node 🟡 and ghost node 🔵, where 'ghost' denotes navigable nodes observed along the path $\mathbf{\Gamma}'$ but have not been explored. The edges $\mathbf{E}_t$ record the Euclidean distances among all adjacent nodes. We map feature vectors $\mathbf{V}_*^p$ onto the nodes as their visual representations. Taking time step t as an example, $\mathbf{V}_t^p$ are first fed into a pano encoder [22] (a two-layer transformer) to obtain contextual view embeddings $\widehat{\mathbf{V}}_t^p$. Since 🟡 and ⚪ have been visited and can access panoramas, they are represented by an average of panoramic view embeddings, *e.g.*, Average$(\widehat{\mathbf{V}}_t^p) \in \mathcal{R}^D$ for 🟡 (D is the embedding dimension). 🔵 is partially observed and therefore is represented by accumulated embeddings of views from which 🔵 can be observed. We equip $\mathbf{G}_t$ with a global action space $\mathcal{A}^G$ for long-term planning, which consists of all observed nodes.

Metric Mapping. The grid-based metric map $\mathbf{M}_t \in \mathcal{R}^{U\times V\times D}$ is constructed locally centered on the current node 🟡. We define $\mathbf{M}_t$ as an egocentric map in which each cell contains a D-sized latent feature representing a small region of the surrounding layouts. Similar to MapNet [18], we ground project grid visual features $\mathbf{V}_*^g$ onto the cells to represent the map. Since $\mathbf{M}_t$ is a local representation and can be observed from nearby visited nodes of the current node, we integrate grid features from nearby visited nodes to construct the map. Concretely, assuming the current node is $\mathbf{n}_i$, we first query the topo map $\mathbf{G}_t$ to get its nearby visited nodes within κ order: $\mathcal{N}_\kappa = \{\mathbf{n}_j|\text{order}(\mathbf{n}_i, \mathbf{n}_j) \le \kappa\}$. Then, we combine the grid features $\mathbf{V}_*^g$ of nodes in $\mathcal{N}_\kappa$, and project them onto the ground plane (centered on the current node), using the corresponding depths $\mathbf{D'}_*$ and poses $\mathbf{P}_*$. The projected features are discretized into the 2D spatial grid $\mathbf{M}_t$, using elementwise average pooling to handle feature collisions in a cell. We equip $\mathbf{M}_t$ with a local action space $\mathcal{A}^M$ for short-term reasoning, which consists of the current node and its adjacent nodes. We compute these nodes' coordinates on $\mathbf{M}_t$ by ground projecting their poses onto the map, namely 'node→cell'.

3.2. Pre-training Model

As presented in Fig. 2 (b), we then fed the hybrid map $(\mathbf{G}_t, \mathbf{M}_t)$ obtained in § 3.1 into a pre-training model to obtain multimodal map representations. The pre-training model contains a topo map encoder and a metric map encoder, which take the instruction $\mathbf{W}$ to fuse with $\mathbf{G}_t$ and $\mathbf{M}_t$ respectively. The outputs are later fed into three pre-training tasks to learn navigation-oriented multimodal map representations (§ 3.3).

3.2.1 Text Encoder

Each word embedding in the instruction $\mathbf{W}$ is added with a position embedding [26] and a text type embedding [5]. Then, all embeddings are fed into a multi-layer transformer to obtain contextual word embeddings $\widetilde{\mathbf{W}}$.

3.2.2 Topo Map Encoder

This module takes the topo map $\mathbf{G}_t$ and the encoded instruction $\widetilde{\mathbf{W}}$ to conduct node-level cross-modal fusion.

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 euclidean distance of each node to the current node, and the step embedding is the latest visited time step for visited nodes (,) and 0 for ghost nodes . We add a zero-vector 'stop' node $\mathbf{n}_0$ in the graph to denote a stop action and connect it with all other nodes.

Cross-modal Long-term Transformer. The encoded node and word embeddings are fed into a multi-layer transformer to conduct node-level cross-modal fusion. The architecture of each layer is similar to LXMERT [5], which contains one bi-directional cross-attention sub-layer, two self-attention sub-layers, and two feed-forward sub-layers. Following [24], we replace the vision self-attention sub-layers with graph-aware self-attention (GASA), which introduces graph topology for node encoding. The outputs are node-instruction-associated representations ($\widetilde{\mathbf{N}}_t$, $\widetilde{\mathbf{W}}^G$).

3.2.3 Metric Map Encoder

This module takes the metric map $\mathbf{M}_t$ and the encoded instruction $\widetilde{\mathbf{W}}$ to conduct cell-level cross-modal fusion.

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:

$$\mathbf{p}_{u,v} = [\cos(\theta_{u,v}), \sin(\theta_{u,v}), \mathrm{dis}_{u,v}] \tag{1}$$

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 [26] or 2D position embedding [63]. 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.

Cross-modal Short-term Transformer. The encoded cell and word embeddings are fed into a multi-layer transformer to conduct cross-modal fusion. Each layer architecture is similar to that in § 3.2.2, but uses self-attention for cell encoding rather than GASA. The short-term transformer conduct cross-modal reasoning on the fine-grained (cell-level) map representation, which can benefit reasoning about complicated spatial relations, such as *"go into the hallway second to the right from the stairs"*. The outputs are cell-instruction-associated representations ($\widetilde{\mathbf{M}}_t$, $\widetilde{\mathbf{W}}^M$).

3.3. Pre-training Tasks

We devise three tasks to learn the multimodal map representations $(\widetilde{\mathbf{N}}_t, \widetilde{\mathbf{M}}_t)$ obtained in § 3.2.

Masked Language Modeling (MLM). MLM is the most commonly used proxy task in BERT pre-training [26]. For VLN, MLM aims to recover masked words $\mathbf{W}_m$ via reasoning over the surrounding words $\mathbf{W}_{\backslash m}$ and the hybrid map. Precisely, we first randomly mask out input tokens of the instruction with a 15% probability and then conduct map-instruction interaction as explained in § 3.2. To learn both long-term and short-term reasoning, we sum the obtained $\widetilde{\mathbf{W}}^G_{\backslash m}$ and $\widetilde{\mathbf{W}}^M_{\backslash m}$, then feed it into the MLM head. This task is optimized by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{MLM}} = -\mathbb{E}_{(\mathbf{W}, \mathbf{\Gamma}') \sim \mathcal{D}} \log \mathcal{P}_\theta(\mathbf{W}_m | \mathbf{W}_{\backslash m}, \mathbf{G}_t, \mathbf{M}_t) \tag{2}$$

where $\mathcal{D}$ denotes the training dataset and θ represents trainable parameters.

Hybrid Single Action Prediction (HSAP). HSAP is designed to benefit the downstream goal: predicting navigation actions. Our model predicts an overall action in the global action space $\mathcal{A}^G$. For a more robust action plan, we integrate the short-term reasoning results from the metric map into the topo map. In practice, we first convert cells lying in the local action space $\mathcal{A}^M$ into the global action space $\mathcal{A}^G$, using a 'cell$\rightarrow$node' operation (the inverse of 'node$\rightarrow$cell' in § 3.1). We denote the converted cells as $\widetilde{\mathbf{M}'}_t = \{\tilde{\mathbf{m}}_i | i \in \mathcal{A}^{G'}\}$, where $\mathcal{A}^{G'}$ is a subset of the global action space $\mathcal{A}^G$. Then, we use two feedforward networks (FFN) to predict navigation scores for nodes $\tilde{\mathbf{n}}_i \in \widetilde{\mathbf{N}}_t$ and cells $\tilde{\mathbf{m}}_i \in \widetilde{\mathbf{M}'}_t$, and dynamic fuse them conditioned on the agent state:

$$\mathbf{s}_i^G = \text{FFN}(\tilde{\mathbf{n}}_i), \quad \mathbf{s}_i^M = \text{FFN}(\tilde{\mathbf{m}}_i) \tag{3}$$

$$\mathbf{s}_i = \begin{cases} \delta_t \mathbf{s}_i^G + (1-\delta_t)\mathbf{s}_i^M, \text{ if } i \in \mathcal{A}^G \cap \mathcal{A}^{G'} \\ \mathbf{s}_i^G, \text{ otherwise} \end{cases} \tag{4}$$

where the padded 'stop' node $\tilde{\mathbf{n}}_0$ and central cell $\tilde{\mathbf{m}}_{c,c}$ denote the agent state, therefore $\delta_t = \text{Sigmoid}(\text{FFN}([\tilde{\mathbf{n}}_0; \tilde{\mathbf{m}}_{c,c}]))$. In most VLN tasks, it is not necessary for an agent to revisit a node, therefore we mask the scores of visited nodes. The task is optimized via a cross-entropy loss over fused scores

$\{\mathbf{s}_i\}$ and teacher action $\mathbf{a}_t^*$:

$$\mathcal{L}_{\text{HSAP}} = -\mathbb{E}_{(\mathbf{W},\mathbf{r},\mathbf{a}_t^*)\sim\mathcal{D}} \log \mathcal{P}_\theta(\mathbf{a}_t^*|\mathbf{W}, \mathbf{G}_t, \mathbf{M}_t) \quad (5)$$

Masked Semantic Imagination (MSI). We note there are some unobserved areas on the metric map $\mathbf{M}_t$, which brings uncertainty for decision-making. To mitigate this issue, we propose MSI to enable the agent to imagine the information of unobserved areas, by reasoning over instructions and partially observed maps. Concretely, we first randomly mask out cells of the metric map $\mathbf{M}_t$ with an empirical 15% probability to simulate unobserved areas. Then, the masked map $\mathbf{M}_{t,\backslash m}$ interacts with the instruction $\mathbf{W}$ as explained in § 3.2. Finally, the MSI head forces the model to predict semantics $\mathbf{S}$ of masked regions conditioned on the multimodal map representation $\widetilde{\mathbf{M}}_{t,\backslash m}$. Each cell of the metric map may contain multiple semantics; therefore, the task is formulated as a multi-label classification problem and optimized via a binary cross-entropy loss:

$$\begin{aligned}\mathcal{L}_{\text{MSI}} = -\mathbb{E}_{(\mathbf{W},\mathbf{r})\sim\mathcal{D}} \sum_i^C [\mathbf{S}_i \log \mathcal{P}_\theta(\mathbf{S}_i|\mathbf{W}, \mathbf{M}_{t,\backslash m}) \\ +(1-\mathbf{S}_i)\log(1-\mathcal{P}_\theta(\mathbf{S}_i|\mathbf{W}, \mathbf{M}_{t,\backslash m}))]\end{aligned} \quad (6)$$

where $\mathbf{S}_i$ corresponds to the i-th semantic class ($C = 40$), and we obtain these labels from Matterport3D dataset [64].

3.4. Training and Inference

Training. As standard practices in transformer-based VLN methods [10, 11, 12], we first mix the three tasks in § 3.3 to pre-train the model with offline expert data. To avoid overfitting to expert experience, we then fine-tune the model with sequential action prediction. The topo map $\mathbf{G}_t$ in this stage is online updated. As shown in Fig. 3, at step t, we obtain $\mathbf{G}_t$ by adding newly observed nodes to $\mathbf{G}_{t-1}$ and updating the node status. For trajectory rollout in fine-tuning, we alternately run 'teacher-forcing' and 'student-forcing' [1]. The 'teacher-forcing' is equivalent to Eq. 5, where the agent always executes the teacher action. In 'student-forcing', at each step, the next action is sampled from the predicted score distribution (Eq. 4) and supervised by pseudo labels [24]. More details are in the appendix.

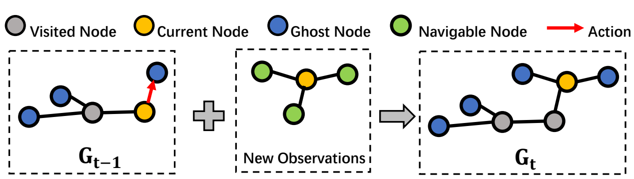

Figure 3: Online topo map update at step t. The agent executes an action to reach a ghost node and receives new observations. It then adds newly observed nodes to $\mathbf{G}_{t-1}$, updating node representations and types. The simulator provides navigable nodes at each step.

Inference. At each step during testing, the agent online constructs a hybrid map similar to the fine-tuning stage, and then performs cross-modal reasoning over the map as explained in § 3.2. Following the single-run setting of VLN, the agent greedily selects the node (ghost node or 'stop' node) with the maximum predicted score (Eq. 4) as the next action. If the selected node is a long-term action (not adjacent to the current node), the agent plans the shortest path to reach the selected node using Dijkstra's algorithm on the current topo map. The agent stops if it selects the 'stop' node or reaches the maximum action steps.

4. Experiments

We evaluate the proposed method on R2R [1], R2R-CE [27], RxR [3] and REVERIE [2] datasets. R2R, R2R-CE, and RxR focus on fine-grained instruction following, whereas R2R-CE is a variant of R2R in continuous environments and RxR provides more detailed path descriptions (*e.g.*, objects and their relations). REVERIE is a goal-oriented task using coarse-grained instructions, such as "Go to the entryway and clean the coffee table".

Navigation Metrics. As in [1, 65, 66], 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 [2], we use Remote Grounding Success (`RGS`) and `RGSPL` (`RGS` penalized by Path Length) to evaluate the capacity of object grounding. 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. Following [67, 42], we use ViT-B/16-CLIP [51] to extract visual features. The scale of grid visual features $\mathbf{V}_t^g$ is 14×14 (outputs before the MLP head of ViT). 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 Configuration. Following [48, 24], we set the layers' number of the text encoder, and the two map encoders as 9, 4, 4. Other hyperparameters are the same as LXMERT [5] (*e.g.* the hidden layer size is 768). In the pre-training stage, we use pre-trained LXMERT for initialization on R2R, R2R-CE, and REVERIE datasets, and pre-trained RoBerta [68] 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. The trainable modules in our model include the pano encoder in § 3.1, the text encoder, and the two map encoders. For all datasets, we first offline pre-train BEVBert with batch size 64 for 100k iterations using 4 NVIDIA Tesla A100 GPUs (∼10 hours). We use the Prevalent [10], RxR-Markey [69] and REVERIE-Spk [24] synthetic instructions as data augmentation on R2R/R2R-CE, RxR and REVERIE respectively. We choose a pre-trained model with the best zero-shot performance (*e.g.*, `SR` + `SPL` on R2R/R2R-CE, `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 (∼20 hours). The best iterations are selected by best performance on validation unseen splits.

4.2. Comparison with State-of-the-Art

R2R. Tab. 1 compares BEVBert against state-of-the-art (SoTA) methods on the R2R dataset. BEVBert beats other methods on all evaluation metrics except for the ensemble-based EnvEdit [42]. On the test unseen split, for instance, BEVBert outperforms the previous best method DUET [24] by 4 `SR` and 3 `SPL`. It is worth noticing that compared with Chasing [62] which also uses metric maps, our improvement is substantial (↑ 40 `SR` and ↑ 32 `SPL` on the test unseen split). We attribute this to our hybrid map design, which balances short-term reasoning and long-term planning, whereas Chasing resorts to metric maps, leading to non-ideal long-term planning capacity. Moreover, Chasing is trained from scratch, while BEVBert gains superior generalization ability with the proposed pre-training framework.

R2R-CE. Tab. 2 presents the results on the R2R-CE dataset. We adjust the topo mapping process in § 3.1 to adapt BEVBert to continuous environments. Specifically, at each step, the agent predicts a set of waypoints [70] and organizes them as a topo map similar to [50]. BEVBert sets new SoTA on the R2R-CE dataset, with 4 `SR` and 2 `SPL` improvement over the topo-map-only ETPNav [50]. This further highlights the efficacy of the proposed hybrid map.

RxR. Tab. 3 reports the results on the RxR dataset. RxR is more challenging than R2R because its paths are much longer and involve more detailed path descriptions. With the fine-grained metric map, BEVBert is skilled at these complex instructions and achieves considerable improvement. For instance, on the test unseen split, BEVBert surpasses the ensemble-based EnvEdit [42] by 4 `SR`, 0.8 `NDTW` and 2.4 `SDTW`. We also report BEVBert's performance without Marky synthetic instructions [69]. Compared to EnvEdit, BEVBert still leads on `SR`, and the improvements over the SoTA single-model HAMT [22] are notable (*e.g.* ↑ 7.6 `SR`, ↑ 0.8 `NDTW` and ↑ 4.3 `SDTW` on the val unseen split).

Methods	Val Unseen				Test Unseen			
	NE↓	OSR↑	**SR↑**	**SPL↑**	NE↓	OSR↑	**SR↑**	**SPL↑**
Seq2Seq [1]	7.81	28	21	-	7.85	27	20	-
SF [32]	6.62	45	36	-	6.62	-	35	28
Chasing [62]	7.20	44	35	31	7.83	42	33	30
RCM [38]	6.09	50	43	-	6.12	50	43	38
SM [33]	5.52	56	45	32	5.67	59	48	35
EnvDrop [39]	5.22	-	52	48	5.23	59	51	47
AuxRN [71]	5.28	62	55	50	5.15	62	55	51
NvEM [36]	4.27	-	60	55	4.37	66	58	54
SSM [19]	4.32	73	62	45	4.57	70	61	46
PREVAL [10]†	4.71	-	58	53	5.30	61	54	51
AirBert [12]†	4.10	-	62	56	4.13	-	62	57
RecBert [48]†	3.93	-	63	57	4.09	70	63	57
REM [40]	3.89	-	64	58	3.87	72	65	59
HAMT [22]†	3.65	-	66	61	3.93	72	65	60
HOP+ [72]†	3.49	-	67	61	3.71	-	66	60
EnvEdit* [42]†	3.24	-	69	64	3.59	-	68	64
TD-STP [49]†	3.22	76	70	63	3.73	72	67	61
DUET [24]†	3.31	81	72	60	3.65	76	69	59
BEVBert (Ours)†	2.81	84	75	64	3.13	81	73	62

Table 1: Comparison with SoTA methods on R2R dataset. * Ensemble of three agents. †Pre-training-based methods.

Methods	Val Unseen				Test Unseen			
	NE↓	OSR↑	**SR↑**	**SPL↑**	NE↓	OSR↑	**SR↑**	**SPL↑**
Seq2Seq [27]	7.37	40	32	30	7.91	36	28	25
CM2 [21]	7.02	42	34	28	7.70	39	31	24
HPN [73]	6.31	40	36	34	6.65	37	32	30
MGMAP [15]	6.28	48	39	34	7.11	45	35	28
CWP [70]	5.74	53	44	39	5.89	51	42	36
Sim2Sim [74]	6.07	52	43	36	6.17	52	44	37
Reborn [75]	5.40	57	50	46	5.55	57	49	45
ETPNav [50]	4.71	65	57	49	5.12	63	55	48
BEVBert (Ours)	4.57	67	59	50	4.70	67	59	50

Table 2: Comparison with SoTA methods on R2R-CE dataset.

Methods	Val Unseen				Test Unseen			
	NE↓	**SR↑**	**NDTW↑**	**SDTW↑**	NE↓	**SR↑**	**NDTW↑**	**SDTW↑**
LSTM [3]	10.9	22.8	38.9	18.2	12.0	21.0	36.8	16.9
EnvDrop+ [67]	-	42.6	55.7	-	-	38.3	51.1	32.4
CLEAR-C [76]	-	-	-	-	-	40.3	53.7	34.9
HAMT [22]	-	56.5	63.1	48.3	6.2	53.1	59.9	45.2
EnvEdit* [42]	-	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 3: Comparison with SoTA methods on RxR dataset. * Ensemble of three agents. †Without Markey-T5 instructions [69].

Methods	Val Unseen			Test Unseen		
	SR↑	**RGS↑**	**RGSPL↑**	**SR↑**	**RGS↑**	**RGSPL↑**
AutoVLN* [47]	55.89	36.58	26.76	55.17	32.23	22.68
FAST [2]	14.40	7.84	4.67	19.88	11.28	6.08
SIA [77]	31.53	22.41	11.56	30.80	19.02	9.20
RecBert [48]	30.67	18.77	15.27	29.61	16.50	13.51
AirBert [12]	27.89	18.23	14.18	30.26	16.83	13.28
HAMT [22]	32.95	18.92	17.28	30.40	14.88	13.08
TD-STP [49]	34.88	21.16	16.56	35.89	19.88	15.40
DUET [24]	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 4: Comparison with SoTA methods on REVERIE dataset. * 900 extra scenes for training.

REVERIE. BEVBert also generalizes well on the goal-oriented REVERIE dataset as shown in Tab. 4. On the val unseen split, BEVBert surpasses the previous best model DUET [24] by 4.80 `SR`, 2.56 `RGS`, and 1.41 `RGSPL`. We also note improvements on the test unseen split are less prominent compared to DUET. We attribute it to the distribution shift between the val unseen and test unseen splits (*e.g.*, comparing the performance difference between val unseen and test unseen, HAMT ↓ 2.55 `SR` *v.s.* DUET ↑ 5.33 `SR`).

4.3. Quantitative and Qualitative Analysis

We present quantitative and qualitative analyses to illustrate BEVBert's efficacy for complex spatial reasoning.

Quantitative Analysis. We aim to evaluate BEVBert's performance on instructions that involve spatial reasoning, such as "go into the hallway second to the right from the stairs". Thus, from R2R and RxR val unseen splits, we first extract the relevant instructions which contain either spatial tokens (*e.g.* "left of", "rightmost") or numerical tokens (*e.g.* "second", "fourth"). An agent's reasoning capability can be inferred from how well it follows these instructions. We compare the performance of BEVBert and SoTA methods on these instructions in Fig. 4. As the number of special tokens in each instruction increases, the performance of all models shows downward trends. This indicates spatial reasoning is a bottleneck of existing methods. However, BEVBert consistently outperforms these counterparts, especially on the RxR dataset which contains more spatial descriptions. This highlights BEVBert's superiority in spatial reasoning.

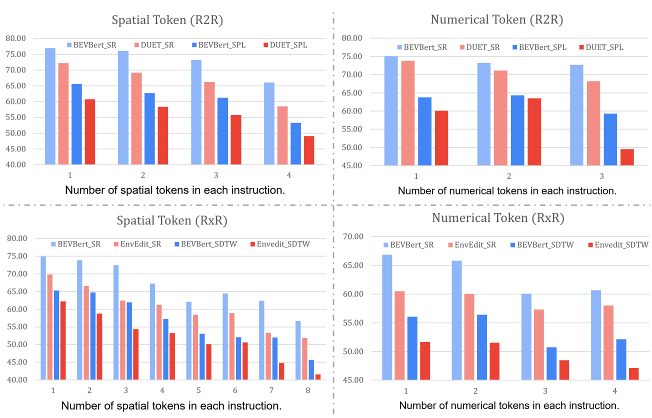

Figure 4: Comparison of navigation performance on spatial and numerical related instructions (BEVBert vs. DUET [24] `SR` (light color) and `SPL` (dark color) on R2R val unseen split, BEVBert vs. EnvEdit [42] `SR` and `SDTW` on RxR val unseen split).

Qualitative Analysis. We visualize the predicted paths of BEVBert and DUET [24] in Fig. 5. DUET uses discrete panoramas for local reasoning, leading to 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. By contrast, thanks to the explicit spatial representation, BEVBert could interpret these complicated descriptions and make correct decisions.

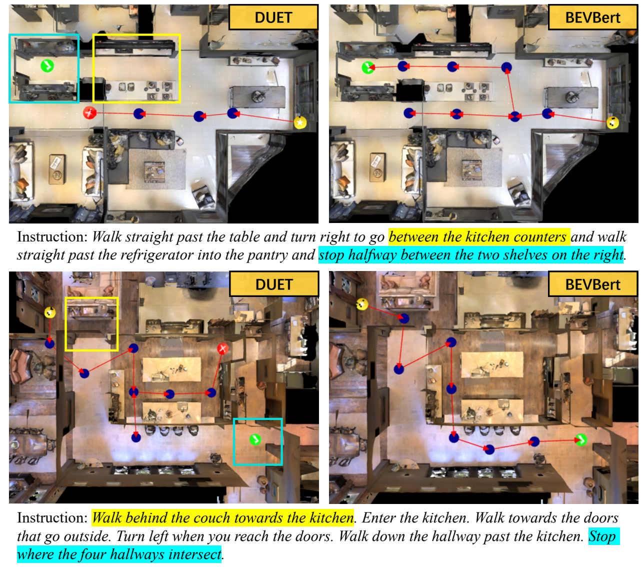

Figure 5: Predicted paths of DUET [24] and BEVBert on R2R-unseen. Yellow and green circles denote the start and target locations, respectively, and the red circles represent incorrect endpoints.

4.4. Ablation Study

We conduct extensive experiments to evaluate key design choices of BEVBert. Results are reported on the R2R val unseen split and the main metrics are highlighted.

1) Comparison of map variants. Tab. 5 presents the results of our model trained with different map variants. Row 1 only uses topo maps for action prediction. It achieves a decent 70.25 `SR`, but there is a clear gap ($\sim$ 4.5 `SR`) with hybrid maps (Row 5, Row 6), due to the lack of metric information for local spatial reasoning. Row 2 further fuses depth features [78] into topo maps' node representations, but with no gain. This suggests that simple depth fusion cannot improve spatial reasoning ability. Row 3 and Row 4 only use metric maps, leading to higher `TL` but poorer navigation performance (`OSR` and `SR`), because the agent lacks long-term planning ability and makes some ineffective exploration. In Row 5 and Row 6, the navigation performance increases substantially when applying the proposed topo-metric maps. It indicates that the proposed hybrid map is a good trade-off between the above two maps, which enables long-term and short-term balanced decision-making.

2) The dependency on depth sensors. We adopt in-domain pre-trained RedNet [79] for depth estimation and then investigate BEVBert's dependence on depth sensors. As shown in Tab. 5 (Row 3 *v.s.* Row 4, Row 5 *v.s.* Row 6), there is almost no performance drop when applying estimated depths for metric mapping. This suggests that our approach does not

#	Map	Depth	TL	NE↓	OSR↑	**SR↑**	**SPL↑**
1	Topo	-	12.59	3.39	78.01	70.25	61.29
2		sensing†	11.76	3.38	77.95	70.03	61.45
3	Metric	estimated	14.41	3.91	70.35	60.64	52.17
4		sensing	14.15	3.93	70.95	60.90	52.80
5	Hybrid	estimated	13.61	2.88	82.63	74.67	63.63
6		sensing	14.55	2.81	83.65	74.88	63.60

Table 5: Comparison of map variants. We denote ground-truth and estimated depths as 'sensing' and 'estimated' respectively. † represents fusing depth features in the topo map setting, other variants do not take depths as model inputs.

highly rely on accurate depth sensing. The main reason is that our metric maps are constructed in feature space, where we use rough grid depths (*e.g.*, 14×14) for feature projection. We believe BEVBert has the potential to be extended in large-scale training with synthetic environments [41, 80], where depth sensors are unavailable.

#	Proxy Tasks	TL	NE↓	OSR↑	**SR↑**	**SPL↑**
1	None	15.56	4.36	73.61	60.24	48.29
2	MLM	16.26	3.09	83.82	73.52	60.13
3	MLM + HSAP	14.50	3.03	82.67	74.03	63.03
4	MLM + HSAP + MSI	14.55	2.81	83.65	74.88	63.60

Table 6: Ablation study of pre-training tasks.

3) The effect of pre-training tasks. Tab. 6 illustrates the effect of different pre-training tasks. Row 1 trains the model from scratch. It has the worst performance because the learned map lacks generic multimodal representations. With the generic MLM task, Row 2 can achieve decent performance (*e.g.*, 73.52 `SR` and 60.13 `SPL`). However, the `TL` is high, thus leading to lower `SPL` compared to Row 3 and Row 4. In Row 3, the `TL` decreases, and `SPL` increases significantly after applying the HSAP task (*e.g.*, ↑ 2.90 `SPL` over Row 2). It indicates that action prediction tasks are beneficial to learn action-informed map representations for efficient navigation. Row 4 further improves the navigation performance with the proposed MSI task (*e.g.*, ↑ 0.85 `SR` and ↑ 0.57 `SPL` over Row 3). The potential reason is that the agent learns to imagine unobserved areas and reduce the uncertainty for decision-making, which helps generalize unseen environments.

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

Table 7: The effect of metric maps scale and size. Scales are set to odd to ensure the agent is at the central cell.

4) Scale and size of metric maps. Tab. 7 reports BEVBert's performance using different scales and sizes of metric maps and the short-term transformer flops. There is an upward trend in performance as the map size increases (Row 2 *v.s.* Row 1), because the agent could perceive environments in a boarder scope. Row 3 performs slightly better than Row 2 when the cell size decreases, which can be contributed to a better perception of minor objects. With a larger map scale, Row 4's performance does not increase obviously. The potential reason lies in the topo map used to capture long-range navigation dependency; thus, a large metric map only brings marginal benefit. On the other hand, a larger metric map causes heavy computation (*e.g.*, flops of the transformer are approximately quadratic w.r.t. the map scale). Therefore, Row 3 is our default setting.

5) The effect of multi-step integration for metric maps. We devise a local integration strategy for metric mapping in § 3.1, which incorporates historical observations from visited nodes within κ order. Tab. 8 presents the effect of κ. With $\kappa = 0$, the metric map is constructed from the current node's observations alone. It has the worst performance due to the lack of historical information, which may confuse the agent to understand mentioned short-term temporal dependency, such as "keep the exhibit board on your right, go ...". When incorporating 1st-order historical observations, Row 2 improves `SPL` by 1.23 over Row 1, but no more gain as κ goes up in Row 3. Because 1st-order integration is enough for a small local map.

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

Table 8: The effect of order κ in metric mapping.

6) Visual features. BEVBert achieves better performance with CLIP pre-trained features as shown in Tab. 9. Imagenet features may lack diverse visual concepts because they are learned by a one-hot classification task that focuses on salient regions of images. By contrast, CLIP features are learned by large-scale image-text matching, where visual grid features are informed by diverse linguistic concepts [67], which can be more suitable for metric mapping.

#	Features	TL	NE↓	OSR↑	**SR↑**	**SPL↑**
1	ViT-B/16-ImageNet [81]	15.90	2.91	83.44	74.03	61.86
2	ViT-B/16-CLIP [51]	14.55	2.81	83.65	74.88	63.60

Table 9: Comparison of different visual features.

5. Conclusion

In this paper, we first devise a hybrid map to balance the demand of VLN for both short-term reasoning and long-term planning. Based on the hybrid map, we propose a new pre-training paradigm, BEVBert, to learn visual-textual associations in an explicit spatial representation. We empirically validate that the learned multimodal map representations could enhance spatial-aware cross-modal reasoning and facilitate the final language-guided navigation goal. Extensive experiments demonstrate the effectiveness of the proposed method and BEVBert achieves state-of-the-art.

6. Acknowledgments

This work was partly supported by National Key Research and Development Program of China Grant No. 2018AAA0100400, National Natural Science Foundation of China (62236010 and 62276261), and Key Research Program of Frontier Sciences CAS Grant No. ZDBS-LYJSC032. We warmly thank ICCV reviewers, Enze Xie, Yicong Hong, and Zun Wang for their valuable suggestions that have helped improve the soundness and quality of this paper.

References

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

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

[3] 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.

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*, pages 104–120. Springer, 2020.

[5] 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.

[6] Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023.

[7] Ge-Peng Ji, Mingchen Zhuge, Dehong Gao, Deng-Ping Fan, Christos Sakaridis, and Luc Van Gool. Masked vision-language transformer in fashion. *Machine Intelligence Research*, 20(3):421–434, 2023.

[8] Yuxian Gu, Jiaxin Wen, Hao Sun, Yi Song, Pei Ke, Chujie Zheng, Zheng Zhang, Jianzhu Yao, Lei Liu, Xiaoyan Zhu, et al. Eva2. 0: Investigating open-domain chinese dialogue systems with large-scale pre-training. *Machine Intelligence Research*, 20(2):207–219, 2023.

[9] Xiao Wang, Guangyao Chen, Guangwu Qian, Pengcheng Gao, Xiao-Yong Wei, Yaowei Wang, Yonghong Tian, and Wen Gao. Large-scale multi-modal pre-trained models: A comprehensive survey. *Machine Intelligence Research*, pages 1–36, 2023.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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 *Advances in Neural Information Processing Systems*.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] Hanqing Wang, Wei Liang, Luc V Gool, and Wenguan Wang. Towards versatile embodied navigation. *Advances in Neural Information Processing Systems*, 35:36858–36874, 2022.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] Jialu Li and Mohit Bansal. Improving vision-and-language navigation by generating future-view image semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10803–10812, 2023.

[44] Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[45] Xiaohan Wang, Wenguan Wang, Jiayi Shao, and Yi Yang. Lana: A language-capable navigator for instruction following and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19048–19058, 2023.

[46] 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.

[47] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Learning from unlabeled 3d environments for vision-and-language navigation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 638–655. Springer, 2022.

[48] 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.

[49] 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.

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

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] Medhini Narasimhan, Erik Wijmans, Xinlei Chen, Trevor Darrell, Dhruv Batra, Devi Parikh, and Amanpreet Singh. Seeing the un-scene: Learning amodal semantic maps for room navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 513–529. Springer, 2020.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

[70] Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15439–15449, 2022.

[71] 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.

[72] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. Hop+: History-enhanced and order-aware pre-training for vision-and-language navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

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

[74] Jacob Krantz and Stefan Lee. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 588–603. Springer, 2022.

[75] Dong An, Zun Wang, Yangguang Li, Yi Wang, Yicong Hong, Yan Huang, Liang Wang, and Jing Shao. 1st place solutions for rxr-habitat vision-and-language navigation competition (cvpr 2022). *arXiv preprint arXiv:2206.11610*, 2022.

[76] 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.

[77] 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.

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

[79] 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.

[80] 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.

[81] 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.