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

- Model details (§A)
- Experimental setups (§B)
- Additional results and visualization (§C)
- Additional analysis of Matterport3D (§D)
- Discussion (§E)

A. Model Details

In our model, BEV Scene Graph (BSG) is proposed to enable discriminative decision space based on BEV feature. However, to align with the discrete environments present in the VLN simulator [1, 2], it is necessary to convert the action space into nodes (Fig. A1). Consequently, BSG can serve as a valuable complement to existing works [3–5] that focus on panoramic decision space (c.f. §A.2). Specifically, our approach incorporates a panoramic branch [5]. We will give more details on how to train this combined model in §A.4.

A.1. Different Decision Space

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

Panoramic Decision Space. To enable high-level action reasoning, panoramic decision space [6] involves discretizing panoramic view of the surrounding environment into 36 view angles (12 headings × 3 elevations with 30 degree intervals). At each location, the agent is limited to a few navigable directions that correspond to these panoramic views. Most existing works [1, 5–9] adopt this decision space. However, due to the adjacency rule, multiple candidate nodes may correspond to the same panoramic view, resulting in ambiguity during route planning.

BEV Grid Decision Space. To address the aforementioned constraints, we introduce a grid-level decision space from bird’s eye view. Each candidate node corresponds to specific BEV grids. The node embedding is represented by its neighboring grid features (Fig. A1).

A.2. Complementary to Existing Methods

As shown in Fig. A1, our method predicts the next step action by fusing both global and fine-scale local decision-making strategies. Specifically, for the topological level, our model predicts the global score on all the navigable nodes, including previously visited and observed nodes, which are similar to previous works [3–5, 10]. Meanwhile, for the local level, the local score are for all navigable nodes of the current node, but our model first predicts the BEV grid-level score in the local level then converts to the score of navigable nodes to making a more accurate prediction. Thus, our model can be easily combined with existing work based on panoramic features as shown in Fig. A1. In this paper, we explore the complementary nature of our model...
with a recent state-of-the-art method [5], which also predicts the global and local score at each step.

A.3. Detailed Network Architecture on REVERIE

Object Prediction. For REVERIE [2], an agent is required to identify an object at each step where additional candidate object annotations are provided. To enable fine-grained perception, we incorporate an object prediction module into local branch. Specifically, we adopt the ViT-B/16 pretrained on ImageNet to extract the features of \( M \) objects at \( t \)-th step \( O_t = \{ o_m | o_m \in \mathbb{R}^{768} \}_{m=1}^M \), and add orientation feature [5, 7] with \( \sin \) and \( \cos \) values for heading and elevation angles. Then these object features are concatenated with BEV features as visual features, and we adopt a cross-modal transformer on visual and textual features to obtain contextual representations. Finally, grid-level decision score and object score are predicted by FFN.

A.4. Pretraining Objectives

For R2R [1] and R4R [11], we adopt Masked Language Modeling (MLM) [12, 13], Masked Region Classification (MRC) [14–17], and Single-step Action Prediction with Progress Monitoring (SAP-PM) [7–9, 18] as auxiliary tasks in the pretraining stage. For REVERIE [2], an additional Object Grounding (OG) [5, 19] is used for object reasoning and grounding, and the sample ratio is MLM:MRC:SAP-PM:OG=1:1:1:1. All the auxiliary tasks are based on the input pair \( (X, G_t, T_t) \), where \( X \) is the textual embedding, \( G_t \) is BSG built at time step \( t \), and \( T_t \) is topological map of complementary method [5] with panoramic visual feature \( V_t \) (c.f. §A.2).

MLM. The task aims to learn grounded language representations in VLN task and cross-modal alignment. It masks some percentage of the input tokens at random, and then predicts those masked tokens based on contextual words and [13]. We randomly mask out one of the word tokens in \( X \) with the probability of 15\% [5, 9], and the final hidden representations corresponding to the [mask] token are fed into an output softmax over the instruction vocabulary:

\[
\mathcal{L}_\text{MLM} = - \log p(x_i | X_{\setminus i}, G_t, T_t), \quad (A1)
\]

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

MRC. This task predicts the semantic labels of masked observation features given instructions and neighboring observations [9]. We only use this task for panoramic branch, and keep other settings consistent with [5, 9].

SAP-PM. We employ imitation learning to predict the next action [5, 9, 17]. Specifically, we sample a map-action pair \( (G_t, T_t, A_t) \) from the groundtruth trajectory at the \( t \)-th step, and then the loss of the panoramic branch is as follows:

\[
\mathcal{L}_\text{SAP} = \sum_{t=1}^{T} - \log p(a_t | X_t, T_t). \quad (A2)
\]

For our BEV branch, we employ an additional progress monitoring task [7, 18] to reflect the navigation progress:

\[
\mathcal{L}_{\text{SAP-PM}} = \sum_{t=1}^{T} - \log p(a_t | X_t, G_t) + \left( y_t^{pm} - p_t^{pm} \right)^2, \quad (A3)
\]

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

OG. The goal of this task is to predict the best matching object among a set of candidate objects at the current viewpoint [5, 19]. The loss is as follows:

\[
\mathcal{L}_\text{OG} = - \log p(o_t | X_t, G_t, T_t), \quad (A4)
\]

where \( o_t \) is the groundtruth object, and we average the matching score of panoramic branch and BEV branch.

A.5. Finetuning Objectives

Since reinforcement learning reward makes the agent pay more attention on shortest paths rather than path fidelity with instruction [9], we alternatively use Teacher-Forcing (TF) and Student-Forcing (SF) for action prediction as behavior cloning (BC):

\[
\mathcal{L}_\text{TF} = \sum_{t=1}^{T} - \log p(a_t | X_t, G_t, T_t), \quad (A5)
\]

\[
\mathcal{L}_\text{SF} = \sum_{t=1}^{T} - \log p(a_t^* | X_t, G_t^*, T_t^*), \quad (A6)
\]

\[
\mathcal{L} = \alpha \mathcal{L}_\text{TF} + \mathcal{L}_\text{SF} + \mathcal{L}_\text{OG}.
\]

B. Experimental Setups

B.1. Evaluation Metrics

VLN. Following the standard setting [1, 6, 9] of R2R, there are several metrics for evaluation: (1) Success Rate (SR) considers the percentage of final positions less than 3 m away from the goal location. (2) Trajectory Length (TL) measures the total length of agent trajectories. (3) Ora-cle Success Rate (OSR) is the success rate if the agent can stop at the closest point to the goal along its trajectory. (4) Success rate weighted by Path Length (SPL) is a trade-off between SR and TL. (5) Navigation Error (NE) refers to the shortest distance between agent’s final position and the goal location. For REVERIE [2, 5, 8], there are two additional metrics. (6) Remote Grounding Success rate (RGS) is the success rate of finding the target object. (7) Remote Grounding Success weighted by Path Length (RGSPL) uses the ratio between the length of the ground-truth path and the agent’s path to normalize RGS. For R4R [4, 9, 11], three metrics are used for instruction fidelity. (8) Coverage
weighted by Length Score (CLS) is the product of the path coverage and length score of the agent’s path with respect to reference path. (9) Normalized Dynamic Time Warping (nDTW) and (10) Success rate weighted normalized Dynamic Time Warping (SDTW) measure the order consistency of agent trajectories.

B.2. Training Details

VLN. During the pretraining stage, we train the combined model with a batch size of 32 for 100k iterations. We then finetune the model with the batch size of 8 for 25k iterations. On REVERIE [2], we select the best epoch by SPL on val unseen. On R2R and R4R [1, 11], the best model is selected according to the sum of SR and SPL on val unseen. For fair comparison, the same synthesize instructions in [5] by a speaker model [6] are also used for REVERIE.

3D Detection. For BEVFormer [21], a static model without using history BEV features is used for 3D detection. We adopt ViT-B/16 [22] pretrained on ImageNet as the backbone. The size of the image features are 1280 × 1024 × 768, and we don’t utilize the multi-scale features in previous work [21, 23, 24]. We train this BEV encoder with detection head [21, 25] using AdamW with a weight decay of 0.01 for 500 epochs, a learning rate of 1 × 10⁻⁴.

For LSS [26] and BEVDepth [24], we use ResNet-50 as the image backbone and the image size is processed to 256 × 704. We don’t adopt image or BEV data augmentations. AdamW is used as an optimizer with a learning rate set to 2 × 10⁻⁴ and batch size set to 48. All experiments are trained for 24 epochs.

C. Additional Results and Visualization

VLN. To compare the differences between the two datasets, we also show an example with the same groundtruth path but different instructions in Fig. C1. It shows that detailed instructions in R2R provide additional information that enables a more accurate navigation strategy.

3D Detection. Table E2 present the detection results on test unseen in Matterport3D². For evaluation, we utilize Average Precision (AP) and Average Recall (AR) with Intersection over Union (IoU) thresholds of 0.25 and 0.50, following established protocols [27–30]. We find that it has good detection performance on larger objects, such as “bed” and ‘sofa’ with 0.535 and 0.394 for AP in Table E2. However, detecting small objects like ‘picture’ and ‘plant’ presents more difficulty since they are almost flat. The detection performance on Matterport3D² can be further improved in the future.

D. Additional Analysis of Matterport3D²

D.1. Detailed Annotation Process

Images of Skybox from Simulator. For each panorama in original Matterport3D [31], the acquisition equipment rotates around the direction of gravity to six distinct orientations, stopping at each to acquire three 1280 × 1024 photos from three RGB cameras pointing up, horizontal, and down, respectively. Consequently, each panorama view contains 6 × 3 raw images. In the VLN task, most previous works [2, 5–8] use the split “skybox” images [31] for panoramic viewing. These “skybox” images are generated by stitching the raw 6 × 3 images. Then, Matterport3D Simulator [1, 6] in the VLN task splits the skybox-based panoramic view into 12 × 3 images with the pre-defined size of 640 × 480. However, this approach does not produce an explicit view transformation matrix.

Raw Camera Images. In order to use accurate camera internal and external parameters for projection in 3D detection¹, we acquire the six raw color images at each viewpoint from the horizontal view for Matterport3D² dataset. Multi-view perspective images captured by camera can access to the original transformation matrix. Given the camera parameters, the resolution of raw camera image is also fixed. Thus we have to use 1280 × 1024 resolution. Specifically, we use the undistorted color images and undistorted camera parameters.

Oriented Bounding Boxes. Although original dataset [31] provides the axis-aligned bounding boxes, they do not provide accurate annotations for 3D detection. Thus, to conform with standard protocols [28, 32], we annotate the oriented bounding boxes (OBB) under LiDAR coordinate system [21, 23]², which surrounding the outline of the objects more tightly than the axis-aligned bounding boxes. We apply Principal Component Analysis (PCA) to the x and y coordinates of segments in each object, as each object consists of many annotated segments.

D.2. Detailed Dataset Statistics

In Table D1, we present the detailed statistics of our Matterport3D² dataset. At each viewpoint, there are six multi-view images (c.f. §D.1). However, since we need to filter the objects at each viewpoint, we only collect the multi-view images of viewpoints that have objects. We use the same train seen, val unseen, and test unseen splits as existing VLN datasets [1, 2].

¹https://github.com/niessner/Matterport/blob/master/data_organization.md
E. Discussion

Asset License and Consent. In this study, we explore vision-language navigation using famous datasets, i.e. Matterport3D [31], R2R [1], and REVERIE [2], that are all publicly available for academic purposes. All the code is released under the MIT license. We implement all models on the MMDetection3D codebase. MMDetection3D codebase (https://github.com/open-mmlab/mmdetection3d) is released under Apache 2.0 license.

Broader Impact. Our work introduces BEV feature for VLN with BSG. Our approach not only achieves a promising improvement of model performance, but also enhances the decision-making by providing grid-level decision score. Furthermore, Matterport3D$^2$ dataset, which includes oriented bounding boxes for indoor 3D detection, will contribute to future research in the community. It should be noted that our navigation agents are developed and evaluated in virtual simulated environments. Since we primarily trained the model in a static environment where all objects are relatively stationary, deploying the algorithm on a real-world robot may result in collisions with moving objects and cause harm to individuals. Therefore, further research and development should be conducted to ensure safe deployment in real-world scenarios, such as adding more speed sensors to avoid collisions and including additional environments to study potential damage risks.

Table D1: Statistics of Matterport3D$^2$ dataset.

<table>
<thead>
<tr>
<th>Split</th>
<th>viewpoints</th>
<th>chair</th>
<th>door</th>
<th>table</th>
<th>picture</th>
<th>cabinet</th>
<th>cushion</th>
<th>window</th>
<th>sofa</th>
<th>plant</th>
<th>shelving</th>
<th>sofa</th>
<th>lighting</th>
<th>plant</th>
<th>sink</th>
<th>table</th>
<th>bed</th>
<th>toilet</th>
<th>chest</th>
<th>appliances</th>
<th>monitor</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>train seen</td>
<td>3463</td>
<td>14665</td>
<td>18934</td>
<td>5511</td>
<td>8493</td>
<td>3632</td>
<td>5334</td>
<td>13918</td>
<td>1056</td>
<td>1000</td>
<td>2213</td>
<td>1875</td>
<td>1831</td>
<td>605</td>
<td>1745</td>
<td>8171</td>
<td>2629</td>
<td>847</td>
<td>9223</td>
<td>20432</td>
<td></td>
<td></td>
</tr>
<tr>
<td>val unseen</td>
<td>439</td>
<td>1634</td>
<td>2456</td>
<td>863</td>
<td>1388</td>
<td>726</td>
<td>1491</td>
<td>1501</td>
<td>176</td>
<td>97</td>
<td>211</td>
<td>233</td>
<td>179</td>
<td>48</td>
<td>72</td>
<td>762</td>
<td>1501</td>
<td>107</td>
<td>12314</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>test unseen</td>
<td>829</td>
<td>2388</td>
<td>4105</td>
<td>1009</td>
<td>2492</td>
<td>1223</td>
<td>1411</td>
<td>2365</td>
<td>289</td>
<td>285</td>
<td>277</td>
<td>1063</td>
<td>601</td>
<td>228</td>
<td>323</td>
<td>547</td>
<td>357</td>
<td>20432</td>
<td>12314</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table E2: Results on Matterport3D$^2$ test unseen.

<table>
<thead>
<tr>
<th>Classes</th>
<th>AP25</th>
<th>AR25</th>
<th>AP50</th>
<th>AR50</th>
</tr>
</thead>
<tbody>
<tr>
<td>cabinet</td>
<td>0.522</td>
<td>0.676</td>
<td>0.348</td>
<td>0.551</td>
</tr>
<tr>
<td>door</td>
<td>0.451</td>
<td>0.649</td>
<td>0.279</td>
<td>0.516</td>
</tr>
<tr>
<td>picture</td>
<td>0.152</td>
<td>0.334</td>
<td>0.053</td>
<td>0.186</td>
</tr>
<tr>
<td>cushion</td>
<td>0.489</td>
<td>0.659</td>
<td>0.281</td>
<td>0.505</td>
</tr>
<tr>
<td>window</td>
<td>0.413</td>
<td>0.570</td>
<td>0.251</td>
<td>0.434</td>
</tr>
<tr>
<td>shelving</td>
<td>0.501</td>
<td>0.629</td>
<td>0.320</td>
<td>0.501</td>
</tr>
<tr>
<td>sofa</td>
<td>0.663</td>
<td>0.765</td>
<td>0.394</td>
<td>0.581</td>
</tr>
<tr>
<td>lighting</td>
<td>0.257</td>
<td>0.486</td>
<td>0.103</td>
<td>0.308</td>
</tr>
<tr>
<td>plant</td>
<td>0.587</td>
<td>0.729</td>
<td>0.352</td>
<td>0.566</td>
</tr>
<tr>
<td>sink</td>
<td>0.486</td>
<td>0.654</td>
<td>0.265</td>
<td>0.486</td>
</tr>
<tr>
<td>table</td>
<td>0.487</td>
<td>0.668</td>
<td>0.306</td>
<td>0.525</td>
</tr>
<tr>
<td>bed</td>
<td>0.691</td>
<td>0.740</td>
<td>0.535</td>
<td>0.649</td>
</tr>
<tr>
<td>toilet</td>
<td>0.529</td>
<td>0.645</td>
<td>0.306</td>
<td>0.456</td>
</tr>
<tr>
<td>chair</td>
<td>0.542</td>
<td>0.695</td>
<td>0.374</td>
<td>0.579</td>
</tr>
<tr>
<td>appliances</td>
<td>0.504</td>
<td>0.613</td>
<td>0.346</td>
<td>0.507</td>
</tr>
<tr>
<td>chest</td>
<td>0.447</td>
<td>0.607</td>
<td>0.247</td>
<td>0.448</td>
</tr>
<tr>
<td>monitor</td>
<td>0.413</td>
<td>0.570</td>
<td>0.264</td>
<td>0.446</td>
</tr>
<tr>
<td>Overall</td>
<td>0.478</td>
<td>0.629</td>
<td>0.295</td>
<td>0.485</td>
</tr>
</tbody>
</table>

References
