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

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

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^2 (§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 $\times$ 3 elevations with 30 degree intervals). At each location, the agent is limited to a few navigable directions that correspond to these panoramic views. Most existing works [1, 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).

*Corresponding author: Wenguan Wang.

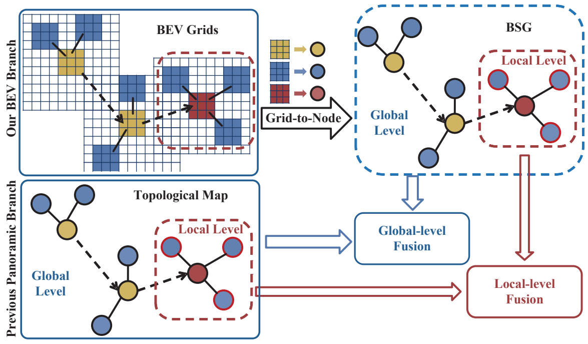


Figure A1: Integrating our framework with previous approaches.

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] are used for object reasoning and grounding, and the sample ratio is MLM:MRC:SAP-PM:OG=1:1:1:1. All the auxiliary tasks are based on the input pair $(\mathcal{X}, \mathcal{G}_t, \mathcal{T}_t)$, where $\mathcal{X}$ is the textual embedding, $\mathcal{G}_t$ is BSG built at time step t, and $\mathcal{T}_t$ is topological map of complementary method [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 $\mathcal{X}$ 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:

$$\mathcal{L}_{\mathrm{MLM}} = -\log p(x_i | \mathcal{X}_{\backslash i}, \mathcal{G}_t, \mathcal{T}_t), \tag{A1}$$

where x_i is the textual embedding of the masked token, $\mathcal{X}_{\backslash i}$ 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 employs imitation learning to predict the next action [5, 9, 17]. Specifically, we sample a map-action pair $(\mathcal{G}_t, \mathcal{T}_t, \mathcal{A}_t)$ from the groundtruth trajectory at the t-th step, and then the loss of panoramic branch is as follows:

$$\mathcal{L}_{\mathrm{SAP}} = \sum\nolimits_{t=1}^{T} -\log p(a_t | \mathcal{X}, \mathcal{T}_t). \tag{A2}$$

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

$$\mathcal{L}_{\mathrm{SAP-PM}} = \sum\nolimits_{t=1}^{T} -\log p(a_t | \mathcal{X}, \mathcal{G}_t) + (y_t^{pm} - p_t^{pm})^2, \tag{A3}$$

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 $\mathcal{L}_{\mathrm{SAP}}$ and $\mathcal{L}_{\mathrm{SAP-PM}}$.

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

$$\mathcal{L}_{\mathrm{OG}} = -\log p(o_i | \mathcal{X}, \mathcal{G}_t, \mathcal{T}_t), \tag{A4}$$

where o_i 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):

$$\begin{aligned} \mathcal{L}_{\mathrm{TF}} &= \sum\nolimits_{t=1}^{T} -\log p(a_t | \mathcal{X}, \mathcal{G}_t, \mathcal{T}_t), \\ \mathcal{L}_{\mathrm{SF}} &= \sum\nolimits_{t=1}^{T} -\log p(a_t^* | \mathcal{X}, \mathcal{G}_t^*, \mathcal{T}_t^*), \end{aligned} \tag{A5}$$

where $\mathcal{G}_t$ and $\mathcal{T}_t$ are maps built online following the expert trajectory, $\mathcal{G}_t^*$ and $\mathcal{T}_t^*$ are following the sampling trajectory, and a_t^* is supervised by the pseudo interactive demonstrator in [5, 20]. On REVERIE, the OG loss is also employed for finetuning, and we adopt a predefined weight $\alpha = 0.20$ to balance them:

$$\mathcal{L} = \alpha \mathcal{L}_{\mathrm{TF}} + \mathcal{L}_{\mathrm{SF}} + \mathcal{L}_{\mathrm{OG}}. \tag{A6}$$

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) Oracle Success Rate (OSR) is the success rate if the agent can stop at the closest point to the goal along its trajectory. (4) Success rate weighted by Path Length (SPL) is a trade-off between SR and TL. (5) Navigation Error (NE) refers to the shortest distance between agent's final position and the goal location. For REVERIE [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 $1280 \times 1024 \times 768$, 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 epoches, 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 Matterport3D^2. For evaluation, we utilize Average Precision (AP) and Average Recall (AR) with Intersection over Union (IoU) thresholds of 0.25 and 0.50, following established protocols [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 Matterport3D^2 can be further improved in the future.

D. Additional Analysis of Matterport3D^2

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 detection[1], we acquire the six raw color images at each viewpoint from the horizontal view for Matterport3D^2 dataset. Multi-view perspective images captured by camera can access to the original transformation matrix. Given the camera parameters, the resolution of raw camera image is also fixed. Thus we have to use 1280×1024 resolution. 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.

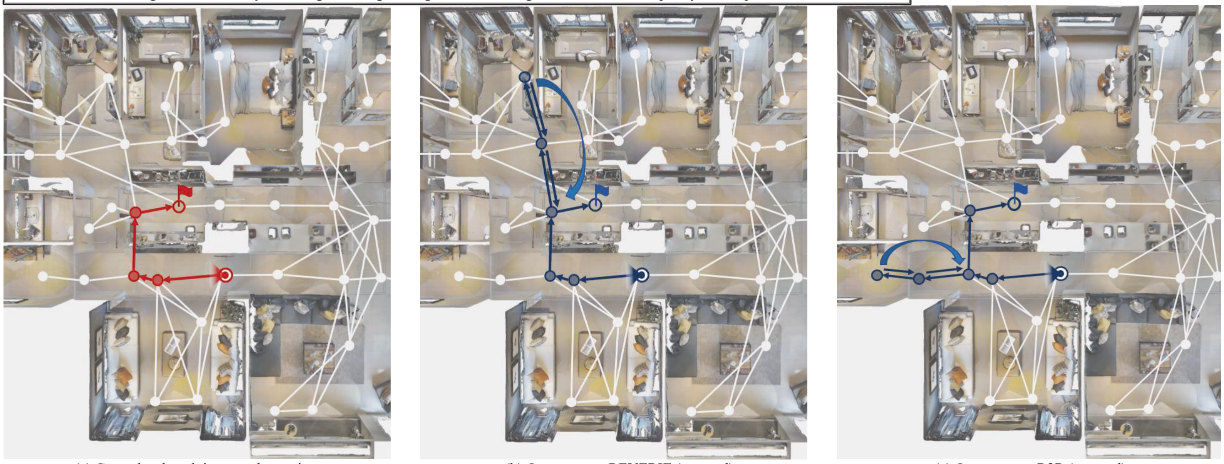
D.2. Detailed Dataset Statistics

In Table D1, we present the detailed statistics of our Matterport3D^2 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].

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

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

REVERIE: Go to the kitchen and turn sink next to the scales on and off.
R2R: Walk across living room, at hallway on the right turn right and go down. Turn right at first door, enter pantry and stop in the middle of counter.



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

Figure C1: Visual results with the same groundtruth path on REVERIE and R2R dataset.

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

Table D1: Statistics of Matterport3D^2 dataset.

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.

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

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

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018. 1, 2, 3, 4

[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 *CVPR*, 2020. 1, 2, 3, 4

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

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

[5] 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 *ICCV*, 2022. 1, 2, 3

[6] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *NIPS*, 2018. 1, 2, 3

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

[8] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez Opazo, and Stephen Gould. VLN BERT: A recurrent vision-and-language BERT for navigation. In *CVPR*, 2021. 2, 3

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

[10] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Topo-metric map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022. 1

[11] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. In *ACL*, 2019. 2, 3

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

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

[14] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NIPS*, 2019. 2

[15] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *ECCV*, 2020.

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

[17] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *CVPR*, 2020. 2

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

[19] Xiangru Lin, Guanbin Li, and Yizhou Yu. Scene-intuitive agent for remote embodied visual grounding. In *CVPR*, 2021. 2

[20] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *ICAIS*, 2011. 2

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

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 3

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

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

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

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

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 3

[28] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 3

[29] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015.

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

[31] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *3DV*, 2017. 3, 4

[32] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 3