Language-Conditioned Navigation Affordance Prediction under Occlusion

Xinyu Gao

Delft University of Technology

TU Delft

AUTONOMOUS MULTI-ROBOTS LAB

TU Delft | Cognitive Robotics

Master of Science in Robotics

Master Thesis

Language-Conditioned Navigation Affordance Prediction under Occlusion

---

Author:

Xinyu Gao
Student Number 5919320

Supervisors:

Prof. Javier Alonso-Mora – Main supervisor
Dr. Gang Chen – Daily supervisor

Date: Mar 30, 2026

Contents

BEACON: Language-Conditioned Navigation Affordance Prediction under Occlusion

Xinyu Gao, Gang Chen†, Javier Alonso-Mora

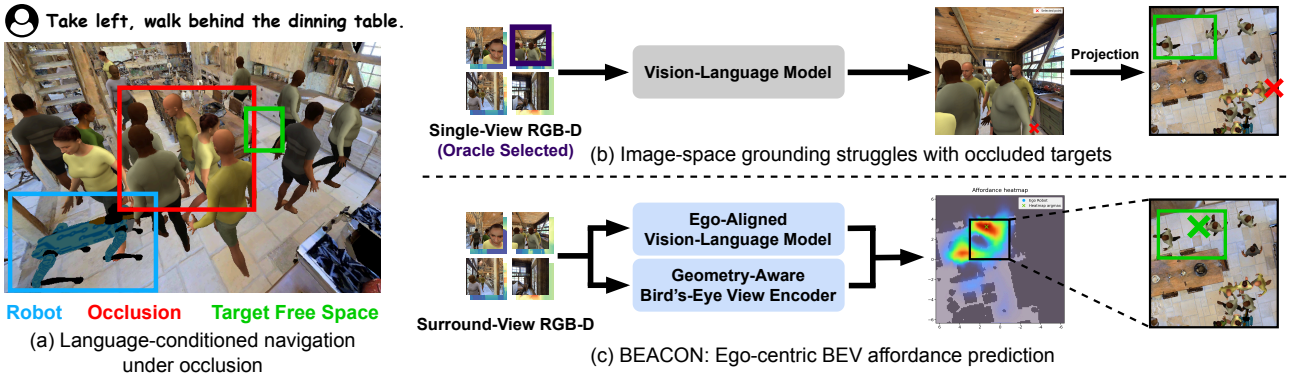


Fig. 1: BEACON predicts an ego-centric Bird's-Eye View (BEV) affordance heatmap for language-conditioned local navigation, which is better suited to occluded targets than the state-of-the-art image-space grounding method.

***Abstract*— Language-conditioned local navigation requires a robot to infer a nearby traversable target location from its current observation and an open-vocabulary, relational instruction. Existing vision-language spatial grounding methods usually rely on vision–language models (VLMs) to reason in image space, producing 2D predictions tied to visible pixels. As a result, they struggle to infer target locations in occluded regions, typically caused by furniture or moving humans. To address this issue, we propose BEACON, which predicts an ego-centric Bird's-Eye View (BEV) affordance heatmap over a bounded local region including occluded areas. Given an instruction and surround-view RGB-D observations from four directions around the robot, BEACON predicts the BEV heatmap by injecting spatial cues into a VLM and fusing the VLM's output with depth-derived BEV features. Using an occlusion-aware dataset built in the Habitat simulator, we conduct detailed experimental analysis to validate both our BEV space formulation and the design choices of each module. Our method improves the accuracy averaged across geodesic thresholds by 22.74 percentage points over the state-of-the-art image-space baseline on the validation subset with occluded target locations. Our project page is: https://xin-yu-gao.github.io/beacon.**

I. Introduction

Language-conditioned local navigation requires a robot to decide where to go from natural instructions that describe a nearby traversable target location through ego-centric directions, landmarks, or scene layout (e.g., "go behind the table," "turn left and move forward," or "go down the hallway"). Unlike tasks that can be addressed by detecting a single object instance, it requires spatial understanding and grounding to a precise traversable target location. In cluttered indoor environments, the target location can be hard to ground from the current observations due to occlusions caused by furniture or people, yet in many cases the robot still needs to choose a feasible local target. This setting requires the robot to infer, from language and current observations, a local target location in its ego-centric frame that is traversable, even when the target is occluded.

Recent vision-language spatial grounding methods [1]–[3] use vision-language models (VLMs) to map observations and instructions to spatial targets and represent the closest existing setup to our problem. These models typically produce image-space point predictions and demonstrate strong open-vocabulary spatial understanding across diverse scenes. However, because image-space outputs are tied to what is directly visible in a particular view, these models struggle to predict target locations under occlusion in the current observations. Occlusion-aware spatial perception for robots has been studied [4], [5], but generally not as a language-conditioned local target prediction problem. Meanwhile, ego-centric Bird's-Eye-View (BEV) representations have proven effective for producing ground-plane outputs under occlusion [6], and recent works show that injecting BEV feature or 3D cues to VLM can improve its performance on ego-centric tasks [7]–[9]. Together, these developments motivate combining VLM-based spatial grounding with ego-centric 3D cues and a robot-centric BEV output for local navigation target prediction under occlusion.

In this work, we propose **BEACON**, a BEV-enhanced affordance prediction model for language-conditioned local navigation under occlusion, shown in Figure 1. Given single-timestep surround-view RGB-D observations and a natural language instruction, BEACON predicts an ego-centric BEV affordance heatmap over nearby ground locations. Here, *affordance* denotes the score indicating how suitable each location is as a local navigation target. BEACON combines an Ego-Aligned Vision-Language Model for

† Corresponding author
The authors are with the Department of Cognitive Robotics (CoR), Delft University of Technology

instruction-conditioned ego-centric scene understanding with a Geometry-Aware Bird's-Eye View Encoder that provides metric spatial structure from RGB-D observations, allowing the model to infer traversable local targets even when they are occluded in the current views.

In detail, our main contributions are as follows: We propose a single-timestep ego-centric BEV navigation affordance prediction method that grounds open-vocabulary instructions into a local BEV affordance heatmap, making it better suited to occluded targets than image-space spatial grounding. We propose an Ego-Aligned VLM that incorporates 3D positional cues to improve language-conditioned target prediction, together with a BEV-space affordance formulation trained with explicit negatives over non-traversable regions to encourage structural validity. Systematic experiments on an occlusion-aware dataset in the Habitat [10] simulator show consistent gains over zero-shot image-space baselines and trained architecture variants under occlusion, validating the role of each design component.

II. Related Work

A. Vision-Language Spatial Grounding in Robotics

The most relevant line of work to our problem is vision-language spatial grounding in robotics, where models map observations and instructions to spatial intermediate outputs that can be consumed by downstream planners. A non-VLM method [11] struggles with non-object descriptions due to its object-centric design. VLM-based methods typically predict one or a few 2D coordinates as target points, often projecting them to 3D using depth for execution. RoboPoint [1] shows that instruction-tuning with synthetic object-reference and free-space reference data enables a general VLM to output image-space points satisfying spatial relations, and demonstrates downstream use in navigation and manipulation. Follow-up directions improve spatial capability via richer supervision [12], explicit reasoning [2], [3], or additional geometric cues [2], often explicitly considering navigation as a downstream use [1]–[3], [12]–[15].

These approaches are effective as general-purpose spatial interfaces because they leverage web-scale pretrained visual semantics and language reasoning. However, robot navigation in cluttered indoor scenes often involves occlusions and indirect cues, where the instruction may imply a landmark or target location behind people or structures. Many existing formulations express outputs in image coordinates or prioritize directly observable evidence during inference and evaluation, and they typically do not explicitly target robot-centric local goal inference under occlusions or enforce structural feasibility (e.g., avoiding walls) for local navigation targets. Our work focuses on this navigation-centric regime and leverages 3D cues with a robot-centric spatial representation to predict targets under occlusion while promoting traversability.

B. VLMs with Local 3D or Ego-Centric Multi-View Inputs

Robots often operate with richer geometric observations than a single RGB image, such as depth or multiple ego-centric views, motivating efforts to extend vision-language models with local 3D or ego-centric multi-view inputs for improved spatial understanding. Some approaches inject 3D cues directly into the 2D vision tokens [7], [8], [16], whereas others introduce a separate depth or 3D branch [2], [17]–[21]. While they demonstrate strong 3D understanding across captioning, question answering, and grounding tasks, these models are not typically designed to output robot-centric local navigation targets, nor are they commonly evaluated in tasks where the referred target is occluded in the current view. Recent work also explores ego-centric multi-view spatial reasoning in vision-language models, but remains focused on reasoning-oriented tasks such as question answering rather than local navigation target prediction [22].

C. BEV Representations and VLM Alignment

BEV representations provide a geometry-centric interface that inherently preserves metric spatial structure and are widely used in occlusion-heavy perception settings [23]–[26], mainly in the self-driving domain. BEV-based methods usually convert visual features to the ground plane using depth or point clouds to produce dense top-down features for various tasks (e.g., detection and segmentation), providing a natural spatial basis for modeling targets behind occlusions while respecting local traversability constraints. In natural language-conditioned tasks, recent advances in self-driving typically provide BEV features as input to the language model, sometimes compressing BEV information into a small set of tokens through adapter modules like [27] and not passing raw images to the downstream language model [9], [28], [29]. While effective for driving objectives, this design may obscure fine-grained spatial structure that is important for precise local goal selection in cluttered indoor environments and may reduce the benefit of web-scale knowledge priors from pretrained vision–language models.

Motivated by these gaps, BEACON aims to retain raw image inputs for VLM-based scene understanding while using depth-derived robot-centric BEV features to preserve local geometry. It then combines language understanding with dense spatial representation to predict instruction-consistent navigation affordance under occlusion while respecting local traversability constraints.

III. Problem Formulation

Given four surround RGB-D views o and a human language instruction x, we aim to infer an ego-centric local navigation target that matches the instruction and lies in traversable free space. This setting requires grounding open-vocabulary instructions, expressed through landmarks, scene structure, or ego-centric directions, to a local destination rather than detecting a single visible object. The intended target may be partially or fully occluded by static structures and/or transient obstacles while still lying within a bounded local area that does not require exploration. Figure 2 illustrates this setting: the left panel shows the task setup under occlusion (isometric view), the middle panel shows the ego-centric top-down local grid with a target region visualization,

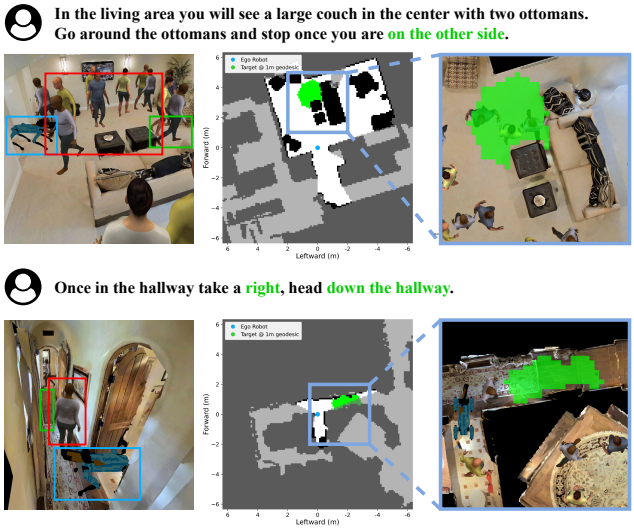


Fig. 2: Examples of language-conditioned local navigation under occlusion. The blue boxes mark the robot, the red boxes highlight humans and objects that cause occlusions, and the green boxes indicate target regions.

and the right panel overlays the same target region on a simulator top-down view.

We formulate local target inference as dense prediction in the ego-centric BEV space. The model outputs an ego-centric BEV navigation affordance map $\hat{A}$, where higher values indicate more likely instruction-relevant target locations, and a single target point for evaluation is obtained by taking the argmax of $\hat{A}$ and mapping the selected BEV cell to its metric location. Formally, we learn a model f_θ that maps observation and instruction to the affordance map:

$$f_\theta(o, x) \rightarrow \hat{A}. \tag{1}$$

During training, $\hat{A}$ is supervised with a target region mask around the annotated target point, as defined in Section IV-D. Dynamic obstacle avoidance and social navigation behaviors (e.g., yielding to pedestrians) are outside the scope.

IV. Methodology

Figure 3 summarizes BEACON, which consists of two stages. Stage 1 adapts a pretrained **Ego-Aligned VLM** (Section IV-A) for ego-centric scene understanding from surround-view observations and natural language instructions. To support this adaptation, we incorporate ego-centric 3D position encoding and perform auto-derived ego-centric instruction tuning, so that the model better interprets spatial language in the agent frame under the surround-view setting. Stage 2 then initializes from the Stage 1 Ego-Aligned VLM weights and builds the full navigation affordance predictor by combining the instruction-conditioned VLM output with a **Geometry-Aware BEV Encoder** (Section IV-B) and a **Post-Fusion Affordance Decoder** (Section IV-C). The Geometry-Aware BEV Encoder provides metric spatial features in the BEV frame for grounding local targets under occlusion, while the Post-Fusion Affordance Decoder combines these features with the VLM output to predict a dense ego-centric BEV navigation affordance heatmap. To encourage structurally valid target prediction in traversable space and reduce sensitivity to imprecise target annotations, Stage 2 is trained with **Geodesic Target Region Supervision** (Section IV-D). At inference time, the final navigation target is obtained by taking the argmax of the predicted heatmap.

A. Ego-Aligned Vision-Language Model

The Ego-Aligned VLM provides instruction-conditioned ego-centric scene understanding from surround-view RGB inputs. It interprets spatial language in the agent frame and outputs a compact signal for BEV target prediction.

1) Ego-Centric 3D Position Encoding: To improve ego-centric scene understanding, we incorporate ego-centric 3D position information into visual tokens following LLaVA-3D [7] and SpatialVLA [8], adapted to a surround-view setting. Given a 2D image patch token v_i from the frozen vision transformer image encoder, we compute its depth-derived 3D position $p_i = (x_i, y_i, z_i)$ in the agent frame. A learnable embedding function $E_{3D}(\cdot)$, implemented as a lightweight two-layer multi-layer perceptron (MLP), maps p_i to the visual feature dimension and is added to the corresponding visual token before the vision-to-language MLP projector:

$$\tilde{v}_i = v_i + E_{3D}(p_i) \tag{2}$$

Then, the MLP projector maps $\tilde{v}_i$ into the language model embedding space.

2) Navigation Task Token Interface: To obtain a single instruction-dependent signal for downstream Bird's-Eye-View prediction, we append a special token `[NAV]` to the prompt and use its final hidden state as a summary embedding following the common practice in vision-language robotic systems like TrackVLA [30], and the embedding is used as the vision-language input to the post-fusion affordance decoder.

3) Stage-1 Auto-Derived Ego-Centric Instruction Tuning: In Stage 1, we perform auto-derived ego-centric instruction tuning with a standard language modeling objective, optimizing the vision-to-language MLP projector, the ego-centric 3D position embedding E_{3D}, the language model's low-rank adaptation (LoRA) [31] parameters. Supervision is constructed automatically from the annotated target in the agent frame as coarse direction-and-range answers. Concretely, direction is discretized into eight 45° bins (e.g. Front, FrontLeft, etc.), and range is split into small or big by a fixed threshold d_{range}. These labels are expressed as short templated textual answers (e.g., "Move towards the FrontLeft region with a small step."), enabling the model to learn the ego-centric convention and integrate surround-view evidence.

In Stage 2, the trained model provides the `[NAV]` summary embedding while the full BEV affordance prediction is trained with geodesic target region supervision.

B. Geometry-Aware Bird's-Eye View Encoder

The Geometry-Aware BEV Encoder constructs an ego-centric BEV feature map F_{BEV} from two complementary sources: (i) dense image features projected to the ground

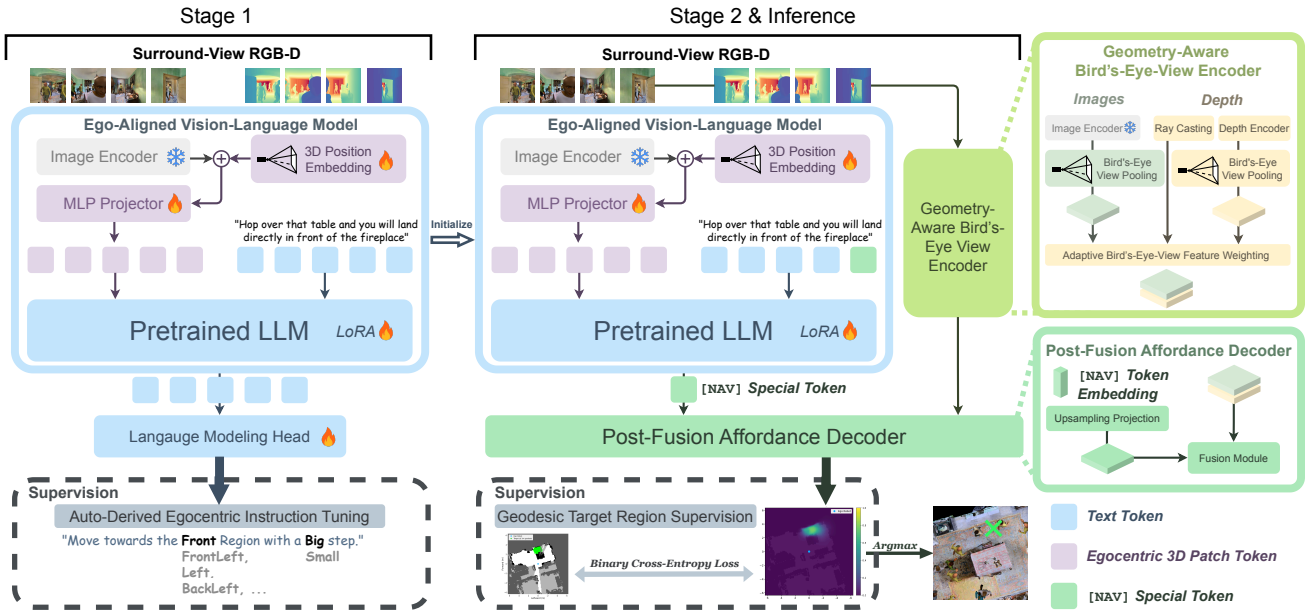


Fig. 3: BEACON overview. Stage 1 performs auto-derived ego-centric instruction tuning with ego-centric 3D position encoding to train the Ego-Aligned VLM. Stage 2 initializes the Ego-Aligned VLM weights from Stage 1, combines the resulting instruction-conditioned output with Geometry-Aware BEV features, and predicts an ego-centric BEV navigation affordance heatmap via a Post-Fusion Affordance Decoder. The two stages use different supervision signals, and inference selects the navigation target by taking the argmax.

plane using depth, camera calibration, and Bird's-Eye-View pooling; and (ii) depth geometry features produced by voxelizing depth points and encoding them with a 3D convolutional depth encoder based on SECOND [32]. Dense image features are extracted using a separate frozen vision backbone DINOv2 [33], distinct from the vision encoder inside the VLM, to preserve high-resolution detail for BEV projection.

We also compute an auxiliary BEV free-space cue M from the current depth observation via ray casting, summarizing which cells are directly observed as free space. This cue is used to predict a per-cell gate $G \in [0,1]$ that controls the relative contribution of image features and geometry features. Concretely, the two BEV sources are mixed and projected as:

$$F_{\mathrm{BEV}} = \phi\Big(\big[(1-G)\odot F_{\mathrm{BEV}}^{\mathrm{Img}},\ M,\ G\odot F_{\mathrm{BEV}}^{\mathrm{Geom}}\big]\Big), \quad (3)$$

where $F_{\mathrm{BEV}}^{\mathrm{Img}}$ denotes the depth-projected BEV image features, $F_{\mathrm{BEV}}^{\mathrm{Geom}}$ denotes the BEV geometry features from the depth encoder, $\odot$ is element-wise multiplication, $[\cdot]$ is channel-wise concatenation, and $\phi(\cdot)$ is a 1×1 projection.

C. Post-Fusion Affordance Decoder

The Post-Fusion Affordance Decoder predicts a dense ego-centric BEV navigation affordance heatmap $\hat{A}$ by fusing the BEV feature map F_{BEV} with the compact embedding $F_{[\mathrm{NAV}]}$ produced by the Ego-Aligned VLM. We map $F_{[\mathrm{NAV}]}$ to a Bird's-Eye-View-aligned feature map via a convolutional upsampling projection to match the BEV grid, concatenate it with F_{BEV}, and predict the BEV affordance heatmap with a standard BEV feature fusion module from BEVFusion [26] followed by convolutional layers.

D. Geodesic Target Region Supervision

Point-only supervision provides weak guidance for dense BEV affordance prediction because it marks a single target location but does not explicitly indicate where not to predict. We adopt BEV target region supervision by aggregating depth observations from a small temporal window around the annotated target to obtain a local traversability estimate. This assumes reasonable depth quality and local pose consistency, and does not rely on simulator ground-truth maps.

Given an annotated target point p^* on the BEV grid, the target region is defined as cells within a geodesic radius r:

$$R(p^*) = \{u \mid d_{\mathrm{geo}}(u, p^*) \le r\}, \quad (4)$$

where d_{geo} is the geodesic distance. Cells in $R(p^*)$ are treated as positives and all other cells as negatives, and we train with a binary cross-entropy loss between $\hat{A}$ and the target region mask.

V. Experimental Setup

We evaluate BEACON on language-conditioned local navigation target prediction, and additionally analyze performance on an occluded-target subset. This section describes the experimental setup, including data construction in the Habitat simulator [10] (Section V-A), the compared baselines (Section V-B), the evaluation metrics (Section V-C), and implementation details (Section V-D). Results are presented in Section VI.

A. Data Construction

We derive local navigation samples from Landmark-RxR [34], [35] by converting each instruction segment into (start viewpoint, instruction, target), where the target is the segment endpoint viewpoint. At each start viewpoint, we render 4 surround-view RGB-D cameras (448×448, 90° FOV each). We restrict targets to a bounded local region (± 6.4 m) and filter out samples outside this bound, requiring exploration beyond the local area (approximated by horizontal raycasts on the scene mesh), or with large height changes (> 0.5 m). The resulting split contains 70 scenes with 75K training samples and 12K unseen validation samples.

Occluded-target subset. We define an occluded-target subset using a depth-consistency test: the target is projected into each rendered view and marked occluded if its projected depth exceeds the rendered depth by more than 0.1 m in all views. Under this definition, 35.84% / 34.37% of train/validation samples are in this subset. To better reflect realistic occlusions from both scene structure and people, we also introduce non-interactive moving pedestrians, implemented with the simulator extension from Social-MP3D [36]. Pedestrian motion is randomized and collision-avoiding. The resulting subset has a slightly larger median target distance than the full validation set (3.12 m vs. 2.32 m).

B. Baselines

We compare BEACON with three groups of methods: general-purpose VLM baselines, spatial-grounding VLM baselines evaluated with oracle-view selection, and a trained task-specific model. The general-purpose VLM baseline is ChatGPT-4o, which we prompt to output image-space points following the RoboRefer [2] prompting setup, and evaluate either on all four views jointly or with oracle-view selection. The spatial-grounding VLM baselines are RoboPoint [1] and RoboRefer, which include navigation-related target grounding as part of their capabilities. In our experiments, we use the RoboPoint-13B checkpoint and the largest publicly released RoboRefer checkpoint, denoted as RoboRefer-8B-SFT. RoboRefer-8B-SFT is our strongest open-source image-space baseline in this setting. Because they use a single-view image-space interface, we evaluate them with oracle-view selection. We additionally report RoboPoint-13B (best point) as a diagnostic upper bound on candidate selection, as RoboPoint outputs multiple point candidates per query. These methods output image-space predictions, so we evaluate them in zero-shot transfer rather than retraining them under our ground-plane target supervision, because occluded navigation targets do not have a well-defined image-space label in the current observation. Finally, as the most straightforward supervised alternative, we train the same VLM with an MLP head to regress a single target point in BEV space from the images and instruction, testing whether BEACON's gains can be explained by straightforward supervised adaptation alone.

C. Evaluation Metrics

Following RoboPoint, we report thresholded target accuracy as the percentage of predicted points that fall within a target region. Because each sample provides a single annotated target point, we evaluate against a radius-t region rather than exact point equality, reducing sensitivity to annotation imprecision and local endpoint ambiguity. We instantiate this in two ways: $\mathrm{GeoAcc@}t$ and $\mathrm{EucAcc@}t$ at $t \in \{0.5, 1.0, 1.5\}$ m, where GeoAcc uses a geodesic target region of radius t in traversable free space and EucAcc uses a Euclidean target region of radius t on the ground plane. GeoAcc is the main metric because it reflects both localization and traversability, while EucAcc isolates spatial proximity even when a prediction falls inside static structure. We also report SIR (structural invalid rate), the fraction of predictions inside non-traversable static structure, to measure geometric validity directly. In the main tables, we report the average over thresholds, denoted by $\overline{\mathrm{GeoAcc}}$ and $\overline{\mathrm{EucAcc}}$.

D. Implementation

We train BEACON in two stages for one epoch each. Stage 1 uses learning rate 3×10^{-5}, with $d_{\mathrm{range}} = 2.4$ m as defined in Section IV-A. Stage 2 uses base learning rate 2×10^{-5}, with a $5\times$ multiplier for the BEV encoder and the post-fusion decoder. We use InternVL2-2B [37] as the VLM throughout the experiments, optimizing the vision-to-language MLP projector, token embeddings, and the language model with LoRA (rank 16, alpha 256, dropout 0.05) while keeping the vision encoder frozen. All experiments run on a single NVIDIA A40 GPU with batch size 4 and gradient accumulation 2. The target geodesic radius r is set to 1 m.

VI. Results

We analyze experiment results to answer three primary questions:

- How does BEACON compare with image-space baselines and the most straightforward trained alternative on local navigation target prediction under occlusion?
- To what extent does each proposed design choice contribute to accuracy and structural validity?
- What qualitative behaviors and failure modes does BEACON exhibit in challenging navigation cases?

We answer these via quantitative results (Section VI-A) and qualitative analysis (Section VI-B) respectively.

A. Quantitative Results

Table I reports the main comparison against the baselines defined in Section V-B on the full validation set and the occluded-target subset, while Table II analyzes the contribution of the Ego-Aligned VLM and BEV-space design choices with an ablation study. In Table II, removing both BEV Encoder and BEV Output gives an Ego-Aligned VLM with an MLP point head; removing only BEV Encoder gives an Ego-Aligned VLM with an MLP heatmap head; and removing only BEV Output gives an Ego-Aligned VLM whose output is updated by attending to BEV features through cross-attention before an MLP point head. Based on these results, we draw the following findings:

TABLE I: Overall quantitative results on local navigation target prediction, comparing image-space baselines, straightforward trained alternative, and BEACON on the full validation set and occluded-target subset. Best results are shown in **bold**.

Method	Input	Output	Full Validation Set (%)				Occluded-Target Subset (%)			
			$\overline{\text{GeoAcc}}\uparrow$	$\overline{\text{EucAcc}}\uparrow$	SIR$\downarrow$	$\overline{\text{GeoAcc}}^{\dagger}_{\text{snap}}\uparrow$	$\overline{\text{GeoAcc}}\uparrow$	$\overline{\text{EucAcc}}\uparrow$	SIR$\downarrow$	$\overline{\text{GeoAcc}}^{\dagger}_{\text{snap}}\uparrow$
General-purpose VLM baselines										
ChatGPT-4o [38]	RGB	image point	9.69	20.65	57.25	18.94	5.69	11.55	54.03	10.39
ChatGPT-4o [38] (oracle-view)	RGB	image point	15.97	30.79	48.20	28.28	9.52	17.11	41.68	15.31
Spatial-grounding VLM baselines with oracle-view selection										
RoboPoint-13B [1]	RGB	image point	20.86	32.59	39.34	30.96	15.43	23.46	35.18	21.88
RoboPoint-13B [1] (best point)	RGB	image point	35.86	46.96	27.63	45.50	29.14	37.72	26.96	36.42
RoboRefer-8B-SFT [2]	RGB-D	image point	38.00	44.65	15.97	42.47	20.09	25.45	21.49	23.65
Trained task-specific models										
VLM + point head	RGB	BEV point	41.25	50.15	19.81	47.50	32.15	39.17	20.00	36.99
BEACON (Ours)	RGB-D	BEV heatmap	**57.72**	**60.17**	**2.13**	**58.50**	**42.83**	**45.36**	**2.60**	**43.56**

† $\text{GeoAcc}_{\text{snap}}$ is computed after snapping the prediction to the nearest oracle traversable cell as a diagnostic upper bound.

TABLE II: Ablation study of key Ego-Aligned VLM and BEV-space design choices. Best results are shown in **bold**.

Stage 1 Tuning	3D Pos. Enc.	BEV Encoder	BEV Output	Val. (%) $\overline{\text{GeoAcc}}\uparrow$	Occluded-Target Subset (%) $\overline{\text{GeoAcc}}\uparrow$	$\overline{\text{EucAcc}}\uparrow$	SIR$\downarrow$
Ego-Aligned VLM design ablations							
		✓	✓	54.76	40.06	42.49	**2.37**
✓		✓	✓	54.36	40.22	42.64	2.62
	✓	✓	✓	53.59	37.93	40.31	2.50
BEV-space design ablations							
✓	✓			48.40	37.26	43.82	16.53
✓	✓	✓		48.57	37.45	43.84	15.73
✓	✓		✓	52.80	36.97	42.01	11.08
✓	✓	✓	✓	**57.72**	**42.83**	**45.36**	2.60

Finding 1: BEACON substantially outperforms prior image-space baselines, especially under occlusion. Table I shows that BEACON achieves the best results among all compared methods on both the full validation set and the occluded-target subset. This holds across both general-purpose VLM baselines and spatial-grounding VLM baselines. Compared with RoboRefer-8B-SFT, the state-of-the-art image-space baseline in our setting, BEACON improves occluded-subset $\overline{\text{GeoAcc}}$ by 22.74 percentage points and reduces SIR from 21.49% to 2.60%. Together, these results show a consistent gap between BEACON and prior image-space baselines in both target accuracy and structural validity, especially under occlusion.

Finding 2: Straightforward supervised adaptation alone is insufficient. Table I shows that training the same VLM with an MLP point head improves over prior image-space baselines, confirming that task-specific supervision is beneficial. However, the gain remains limited: on the full validation set, its $\overline{\text{GeoAcc}}$ is only 3.25 points higher than RoboRefer-8B, and it still remains clearly below BEACON on both accuracy and structural validity. Table II further shows that removing any major proposed component leads to a noticeable drop in performance. Together, these results indicate that BEACON's gains do not come from supervised adaptation alone, but from the combined effect of its proposed design choices.

Finding 3: BEACON's gains are not just from post-hoc snapping. Table I shows that BEACON improves $\overline{\text{EucAcc}}$ and $\overline{\text{GeoAcc}}_{\text{snap}}$ in addition to $\overline{\text{GeoAcc}}$, so its gains are not explained only by producing fewer invalid predictions and relying on snapping as a post-hoc correction. Table II further shows that the Ego-Aligned VLM design improves $\overline{\text{EucAcc}}$ on the occluded-target subset, indicating better language-conditioned target prediction even under a metric that does not enforce structural validity. Notably, ego-centric 3D position encoding alone does not consistently help, and only becomes beneficial when combined with Stage-1 ego-centric instruction tuning, which further shows that the gain comes from the coordinated design of our Ego-Aligned VLM, rather than from adding 3D positional information in isolation.

Finding 4: BEACON yields drastically lower non-traversable predictions. Table I shows that BEACON achieves a drastically lower SIR on both the full validation set and the occluded-target subset (2.13 and 2.60, respectively), indicating that its predicted targets rarely fall inside non-traversable static structure. Table II supports that this improvement comes from BEV-space modeling: removing BEV components sharply increases SIR on the occluded-target subset (11.08–16.53). Notably, the lowest SIR is achieved only when both BEV Encoder and BEV Output are enabled, consistent with our design choice of combining BEV geometric features with a BEV-space affordance output.

TABLE III: Ablation study on F_{BEV} components.

$F^{\text{Img}}_{\text{BEV}}$	$F^{\text{Geom}}_{\text{BEV}}$	G	Full Val. (%) $\overline{\text{GeoAcc}}\uparrow$ / SIR$\downarrow$	Occ. Subset (%) $\overline{\text{GeoAcc}}\uparrow$ / SIR$\downarrow$
✓			55.99 / 3.22	41.52 / 3.77
	✓		51.67 / 9.54	36.93 / 12.51
✓	✓		56.96 / **2.12**	42.34 / 2.62
✓	✓	✓	**57.72** / 2.13	**42.83** / **2.60**

BEV feature component ablation. Table III studies the BEV feature construction in Section IV-B by ablating the image feature branch $F_{\mathrm{BEV}}^{\mathrm{Img}}$, the geometry feature branch $F_{\mathrm{BEV}}^{\mathrm{Geom}}$, and the learned gate G. Using only $F_{\mathrm{BEV}}^{\mathrm{Img}}$ already gives strong performance, while using only $F_{\mathrm{BEV}}^{\mathrm{Geom}}$ substantially lowers $\overline{\mathrm{GeoAcc}}$ and increases SIR. Combining the two branches improves both accuracy and validity, showing their complementarity. Adding the gate gives a further gain in $\overline{\mathrm{GeoAcc}}$ while preserving very low SIR, supporting the design of the Geometry-Aware BEV Encoder.

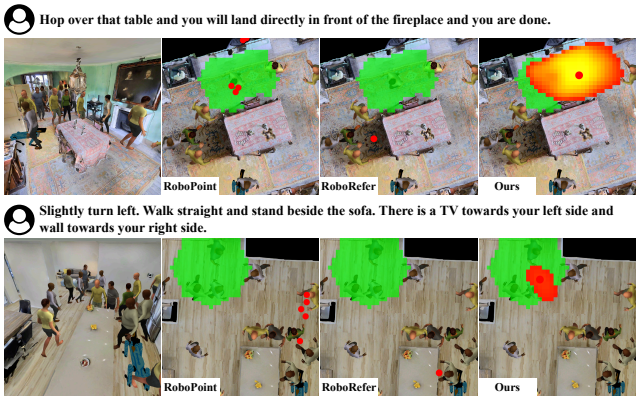
B. Qualitative Analysis

Figure 4 compares BEACON's BEV affordance predictions with the image-space baselines RoboPoint and RoboRefer. For readability, the heatmap overlay is thresholded at 0.40 so that only high-confidence regions are shown. The top two rows show successful examples under heavy occlusion, while the bottom two rows illustrate representative failure cases. Here, successful means that the selected target lies inside the 1 m geodesic target region.

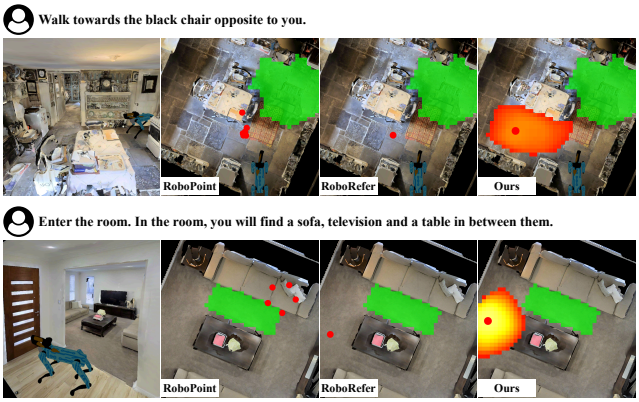
Affordance prediction under heavy occlusion. In the first successful example shown in Figure 4a, part of the referred structure is visible, but the target-side free space is heavily occluded; BEACON concentrates affordance in the feasible gap and selects a target inside the target region. In the second example, the relevant landmarks are not directly visible and the observation provides mainly layout cues; BEACON still assigns probability mass toward the correct direction and rough location, whereas the image-space baselines fail without a directly visible grounding cue.

Uncertainty representation and structural validity. Our affordance map explicitly represents uncertainty as a spatial distribution over candidate free-space goals, while remaining anchored to traversable geometry. In the first successful example and the first failure case, probability mass follows feasible corridors around furniture rather than spreading into walls or obstacles, so the selected target is less likely to fall inside static structure. This behavior is encouraged by the geodesic region supervision, which provides explicit negatives on infeasible regions and suppresses non-traversable areas even when the semantic prediction is imperfect, consistent with the low SIR observed quantitatively. By contrast, image-space baselines do not explicitly model free-space feasibility: RoboRefer tends to select conservative visible-floor points that miss occluded targets, while RoboPoint may predict semantically relevant pixels whose depth projection is not traversable.

Failure cases. The two rows in Figure 4b summarize two common failure modes. In the first row, the model confuses the referred landmark or relation (e.g., which chair is black and which one is "opposite"), yielding a coherent but misplaced affordance peak. In the second row, the instruction is underspecified about how far to proceed after entering the room; the prediction corresponds to a plausible stopping region but exhibits an ambiguity-induced mismatch with the single annotated endpoint.



(a) Successful examples under heavy occlusion.



(b) Failures due to landmark confusion or instruction ambiguity.

Fig. 4: Qualitative examples of language-conditioned navigation affordance prediction, comparing BEACON's BEV affordance predictions with the image-space baselines RoboPoint [1] and RoboRefer [2]. Target regions are shown in green.

VII. Conclusion

In this work, we propose BEACON, a VLM-based BEV affordance predictor for local navigation target prediction conditioned on an open-vocabulary instruction. In unseen environments in the Habitat simulator, BEACON shows consistent gains over prior image-space baselines, with the largest improvements on the occluded-target subset. While image-space baselines struggle with occluded cues or targets, BEACON outputs an ego-centric BEV affordance heatmap that yields more accurate targets and substantially fewer non-traversable predictions. These improvements are not simply the result of adding task-specific supervision, nor are they explained solely by post-hoc snapping to free space; instead, BEACON improves both Euclidean target accuracy and traversable-target validity. Extensive ablations further validate the importance of ego-aligned 3D cues and BEV-space design choices.

While BEACON demonstrates strong results in simulation, evaluating it on real-world surround-view RGB-D data with matched instruction segments is an important next step. Looking ahead, incorporating more explicit compositional grounding of intermediate entities and relations, together

with process-level supervision, may further improve multi-step spatial reasoning.

REFERENCES

[1] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox, "Robopoint: A vision-language model for spatial affordance prediction in robotics," in *Conference on Robot Learning*. PMLR, 2025, pp. 4005–4020.

[2] E. Zhou, J. An, C. Chi, Y. Han, S. Rong, C. Zhang, P. Wang, Z. Wang, T. Huang, L. Sheng *et al.*, "Roborefer: Towards spatial referring with reasoning in vision-language models for robotics," *arXiv preprint arXiv:2506.04308*, 2025.

[3] Y. Liu, D. Chi, S. Wu, Z. Zhang, Y. Hu, L. Zhang, Y. Zhang, S. Wu, T. Cao, G. Huang *et al.*, "Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning," *arXiv preprint arXiv:2501.10074*, 2025.

[4] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proc. of the IEEE/CVF Comput. Vis. and Pattern Recognition Conf. (CVPR)*, 2017, pp. 1746–1754.

[5] A. Reed, B. Crowe, D. Albin, L. Achey, B. Hayes, and C. Heckman, "Scenesense: Diffusion models for 3d occupancy synthesis from partial observation," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (IROS)*. IEEE, 2024, pp. 7383–7390.

[6] Y. Zhang, J. Zhang, Z. Wang, J. Xu, and D. Huang, "Vision-based 3d occupancy prediction in autonomous driving: a review and outlook," *Frontiers of Computer Science*, vol. 20, no. 1, p. 2001301, 2026.

[7] C. Zhu, T. Wang, W. Zhang, J. Pang, and X. Liu, "Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities," in *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2025, pp. 4295–4305.

[8] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang *et al.*, "Spatialvla: Exploring spatial representations for visual-language-action model," *arXiv preprint arXiv:2501.15830*, 2025.

[9] H. Shao, Y. Hu, L. Wang, G. Song, S. L. Waslander, Y. Liu, and H. Li, "Lmdrive: Closed-loop end-to-end driving with large language models," in *Proc. of the IEEE/CVF Comput. Vis. and Pattern Recognition Conf. (CVPR)*, 2024, pp. 15 120–15 130.

[10] X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. W. Clegg, M. Hlavac, S. Y. Min *et al.*, "Habitat 3.0: A co-habitat for humans, avatars and robots," *arXiv preprint arXiv:2310.13724*, 2023.

[11] D. Kim, N. Oh, D. Hwang, and D. Park, "Lingo-space: Language-conditioned incremental grounding for space," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 9, 2024, pp. 10 314–10 322.

[12] X. Shao, Y. Tang, P. Xie, K. Zhou, Y. Zhuang, X. Quan, J. Hao, L. Zeng, and X. Li, "More than a point: Capturing uncertainty with adaptive affordance heatmaps for spatial grounding in robotic tasks," *arXiv preprint arXiv:2510.10912*, 2025.

[13] C. H. Song, V. Blukis, J. Tremblay, S. Tyree, Y. Su, and S. Birchfield, "Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics," in *Proc. of the IEEE/CVF Comput. Vis. and Pattern Recognition Conf. (CVPR)*, 2025, pp. 15 768–15 780.

[14] Y. Tang, L. Zhang, S. Zhang, Y. Zhao, and X. Hao, "Roboafford: A dataset and benchmark for enhancing object and spatial affordance learning in robot manipulation," in *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025, pp. 12 706–12 713.

[15] X. Hao, Y. Tang, L. Zhang, Y. Ma, Y. Diao, Z. Jia, W. Ding, H. Ye, and L. Chen, "Roboafford++: A generative ai-enhanced dataset for multimodal affordance learning in robotic manipulation and navigation," *arXiv preprint arXiv:2511.12436*, 2025.

[16] A.-C. Cheng, Y. Fu, Y. Chen, Z. Liu, X. Li, S. Radhakrishnan, S. Han, Y. Lu, J. Kautz, P. Molchanov *et al.*, "3d aware region prompted vision language model," *arXiv preprint arXiv:2509.13317*, 2025.

[17] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan, "3d-llm: Injecting the 3d world into large language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 20 482–20 494, 2023.

[18] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang, "An embodied generalist agent in 3d world," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 20 413–20 451.

[19] J. Huang, X. Ma, X. Linghu, Y. Fan, J. He, W. Tan, Q. Li, S.-C. Zhu, Y. Chen, B. Jia *et al.*, "Leo-vl: Towards 3d vision-language generalists via data scaling with efficient representation," *arXiv e-prints*, pp. arXiv–2506, 2025.

[20] W. Cai, I. Ponomarenko, J. Yuan, X. Li, W. Yang, H. Dong, and B. Zhao, "Spatialbot: Precise spatial understanding with vision language models," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*. IEEE, 2025, pp. 9490–9498.

[21] A.-C. Cheng, H. Yin, Y. Fu, Q. Guo, R. Yang, J. Kautz, X. Wang, and S. Liu, "Spatialrgpt: Grounded spatial reasoning in vision-language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 135 062–135 093, 2024.

[22] M. Gholami, A. Rezaei, Z. Weimin, S. Mao, S. Zhou, Y. Zhang, and M. Akbari, "Spatial reasoning with vision-language models in ego-centric multi-view scenes," *arXiv preprint arXiv:2509.06266*, 2025.

[23] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *European conference on computer vision*. Springer, 2020, pp. 194–210.

[24] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 3, pp. 2020–2036, 2024.

[25] Z. Li, Z. Yu, D. Austin, M. Fang, S. Lan, J. Kautz, and J. M. Alvarez, "Fb-occ: 3d occupancy prediction based on forward-backward view transformation," *arXiv preprint arXiv:2307.01492*, 2023.

[26] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*. IEEE, 2023, pp. 2774–2781.

[27] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[28] K. Winter, M. Azer, and F. B. Flohr, "Bevdriver: Leveraging bev maps in llms for robust closed-loop driving," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (IROS)*. IEEE, 2025, pp. 20 379–20 385.

[29] Z. Liu, R. Huang, R. Yang, S. Yan, Z. Wang, L. Hou, D. Lin, X. Bai, and H. Zhao, "Drivepi: Spatial-aware 4d mllm for unified autonomous driving understanding, perception, prediction and planning," *arXiv preprint arXiv:2512.12799*, 2025.

[30] S. Wang, J. Zhang, M. Li, J. Liu, A. Li, K. Wu, F. Zhong, J. Yu, Z. Zhang, and H. Wang, "Trackvla: Embodied visual tracking in the wild," in *Conference on Robot Learning*. PMLR, 2025, pp. 4139–4164.

[31] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[32] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[33] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[34] K. He, Y. Huang, Q. Wu, J. Yang, D. An, S. Sima, and L. Wang, "Landmark-rxr: Solving vision-and-language navigation with fine-grained alignment supervision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 652–663, 2021.

[35] A. Ku, P. Anderson, R. Patel, E. Ie, and J. 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)*, 2020, pp. 4392–4412.

[36] Z. Gong, T. Hu, R. Qiu, and J. Liang, "From cognition to precognition: A future-aware framework for social navigation," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*. IEEE, 2025, pp. 9122–9129.

[37] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu *et al.*, "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," in *Proc. of the IEEE/CVF Comput. Vis. and Pattern Recognition Conf. (CVPR)*, 2024, pp. 24 185–24 198.

[38] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

Appendix
Parallel Work: Occupancy Prediction Benchmarking for Mobile Robots

A.1. Motivation

While the main body of this thesis focuses on language-conditioned navigation affordance prediction, it does not explicitly represent the surrounding environment beyond the predicted target region. In many practical mobile robot navigation systems, however, safe motion still often relies on an explicit model of the local environment, for example to represent free space, obstacles, and nearby dynamic agents. To complement the main thesis from this perception-oriented perspective, this appendix presents parallel work carried out during the thesis on occupancy prediction for mobile robots in human-populated environments.

Occupancy prediction provides a structured local representation of free space, obstacles, and semantic scene layout that is directly useful for collision avoidance and local planning. While semantic and panoptic occupancy prediction have been widely studied in autonomous driving [1], corresponding benchmarks for mobile robots remain much less developed, especially in near-field human-populated environments. This appendix therefore focuses on vision-based occupancy prediction for mobile robots, including benchmark setup together with baseline adaptation and evaluation.

A.2. Related Work

A.2.1. Occupancy Prediction for Mobile Robots

Vision-based semantic occupancy prediction estimates a dense 3D field of free/occupied/unknown space together with semantic labels, providing a map-like representation that can support downstream navigation. Most widely used occupancy benchmarks are built for autonomous driving [2, 3, 4, 5], where large-scale human-annotated datasets enable holistic evaluation and have motivated extensions toward instance-aware (panoptic) occupancy [6] and motion-aware occupancy prediction [7].

In contrast, mobile-robot settings are less well covered by standardized occupancy datasets. Robot platforms often operate at shorter ranges and in human-populated environments, where non-rigid pedestrians and near-field occlusions are frequent; however, existing robot-centric occupancy resources [8, 9] remain comparatively scarce and often lack human-cluttered dynamics, panoptic instance annotations, and dynamic-oriented evaluation. This appendix focuses on this gap through a standardized benchmark setup for mobile-robot occupancy prediction.

A.2.2. Representative Vision-based Occupancy Methods

Modern camera-based occupancy pipelines largely follow two families. First, depth-lifting pipelines adopt Lift–Splat–Shoot style designs [10, 11, 12, 13] and have been extended to occupancy prediction through explicit spatial feature construction [14]. Second, query-based methods form explicit spatial representations via attention over upstream features, including scene-query designs such as Bird's-Eye View (BEV), Tri-Perspective View (TPV), and volume queries [15, 16, 17, 18], as well as sparse query variants that decode masks or Gaussian primitives for efficiency [19, 20, 21, 22, 23].

For panoptic occupancy, representative designs either use explicit instance queries to predict instance-level properties and assign voxels accordingly [6], or reuse shared spatial feature maps and attach lightweight detection-style heads to recover instances alongside dense occupancy [24]. The baselines evaluated in this appendix are drawn from these representative paradigms.

A.3. Task Definition

This appendix focuses on the benchmark setup and baseline evaluation in our work *MobileOcc* [25], where raw sensor data are taken from the CODa dataset [26], occupancy labels are generated through a separate annotation pipeline, and the present contribution is to turn the labeled data into a benchmark setting for mobile-robot occupancy prediction through benchmark setup, baseline adaptation, and evaluation.

A.3.1. Task Setup

This appendix considers local occupancy prediction around the robot from visual observations. The model receives temporal stereo camera inputs and uses estimated ego-motion to align information across time. During dataset construction and training, LiDAR is available and is used to provide supervision signals such as pseudo-depth and occupancy labels; however, at inference time the system relies on stereo images only. The output is a semantic occupancy voxel grid covering a bounded local region in front of the robot with semantic categories relevant to human-populated scenes. In addition, the benchmark supports panoptic occupancy for pedestrians, enabling evaluation of both semantic occupancy and instance-level pedestrian prediction. The task is defined primarily as current-time occupancy prediction, with pedestrian ground-plane velocity prediction considered as an auxiliary motion output. Table A.1 summarizes the training and inference setup, while Figure A.1 illustrates the task input and output at inference time.

Training input	Training supervision	Inference input	Inference output
• stereo images • LiDAR for label generation	• semantic occupancy • pedestrian instances • pedestrian planar velocity	• stereo images	• semantic occupancy • pedestrian instances • pedestrian planar velocity

Table A.1: Task setup for occupancy prediction.

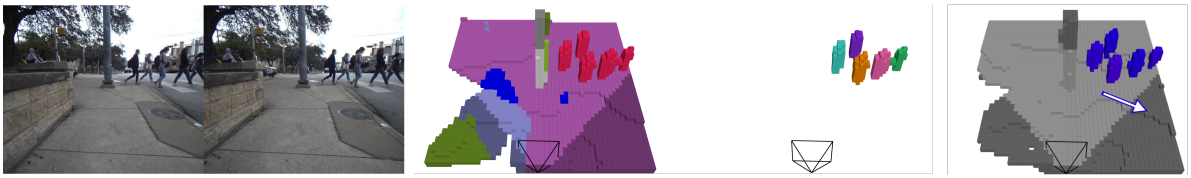


Figure A.1: Task input and output at inference time.

All predictions are defined in the ego frame within a bounded local region around the robot; the exact spatial range, voxel resolution, and temporal context are specified in the benchmark and experimental setup sections. Table A.2 summarizes the notation used in this appendix.

Symbol	Meaning
$\mathcal{F}_e$	ego/robot coordinate frame (with x forward, y left, z up)
$\mathcal{R}$	bounded local region around the robot used for occupancy prediction
I	visual observation(s)
V, $\hat{V}$	ground-truth and predicted semantic occupancy volumes over $\mathcal{R}$
$\mathbf{v} = (v_x, v_y)$	planar velocity for pedestrian instances

Table A.2: Notation used in the appendix.

A.3.2. Evaluation Metrics

We evaluate occupancy prediction using the following metrics.

Geometric IoU. We report the geometric intersection-over-union (IoU) between predicted and ground-truth non-free voxels:

$$\mathrm{IoU} \;=\; \frac{|\hat{\Omega} \cap \Omega|}{|\hat{\Omega} \cup \Omega|}, \tag{A.1}$$

where Ω and $\hat{\Omega}$ denote the sets of non-free voxels in V and $\hat{V}$, respectively.

Mean IoU (mIoU). For each semantic class c, let TP_c, FP_c, and FN_c denote the number of true-positive, false-positive, and false-negative voxels. The class IoU and mean IoU are

$$\mathrm{IoU}_c \;=\; \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c}, \qquad \mathrm{mIoU} \;=\; \frac{1}{C}\sum_{c=1}^{C} \mathrm{IoU}_c. \tag{A.2}$$

Panoptic quality. For panoptic occupancy, we report Panoptic Quality (PQ) and its relaxed variant $\mathrm{PQ}^{\dagger}$. Standard PQ matches predicted and ground-truth segments using an IoU threshold (commonly 0.5), which is well-suited to LiDAR-based panoptic segmentation but can be overly strict for vision-only occupancy where stuff regions are typically less accurate. Following [27], $\mathrm{PQ}^{\dagger}$ relaxes the strict IoU requirement for stuff classes and is therefore more indicative for camera-only occupancy prediction.

$$\mathrm{PQ} \;=\; \frac{\sum_{(p,g)\in \mathrm{TP}} \mathrm{IoU}(p,g)}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|}. \tag{A.3}$$

We additionally report pedestrian detection performance using Average Precision $\mathrm{AP}^{\mathrm{Ped}}$ evaluated on pedestrian instance centers under multiple distance thresholds following [28].

Pedestrian velocity error. For pedestrian velocity prediction, we report the mean absolute velocity error (AVE):

$$\mathrm{AVE} \;=\; \frac{1}{N}\sum_{i=1}^{N} \|\hat{\mathbf{v}}_i - \mathbf{v}_i\|_2\,, \tag{A.4}$$

where $\mathbf{v}_i$ and $\hat{\mathbf{v}}_i$ denote the ground-truth and predicted planar velocities, and N is the number of evaluated entities (e.g., pedestrian voxels or matched detections). In Section A.5.2 we report AVE under different evaluation subsets consistent with the benchmark protocol.

A.4. Benchmarking Methodology

This section summarizes the occupancy prediction benchmark setup used in this appendix, corresponding to our submitted paper [25]. It focuses on (i) the benchmark setup and standardization choices for mobile-robot perception, and (ii) the adaptation of representative published baselines to our sensor setup and label space. Further implementation details are provided in Section A.5.

A.4.1. Benchmark Setup and Standardization

Benchmark Scope and Evaluation Target

We evaluate semantic occupancy prediction in a local 3D occupancy volume $\hat{V}$ defined over the bounded region $\mathcal{R}$ in front of the robot. The evaluation volume covers $x \in [0.4, 10.0]\,\mathrm{m}$, $y \in [-4.8, 4.8]\,\mathrm{m}$, and $z \in [-1.0, 3.8]\,\mathrm{m}$ at a voxel resolution of $0.2\,\mathrm{m}$. This spatial extent captures the near-field region most relevant for mobile-robot navigation while remaining computationally tractable. We downsample the original sequences to 5 Hz (every second frame) to reduce temporal redundancy compared to 10 Hz, while providing denser temporal supervision than typical autonomous-driving benchmarks (e.g., nuScenes at 2 Hz) and better matching the temporal scale of near-field human motion in mobile-robot scenes.

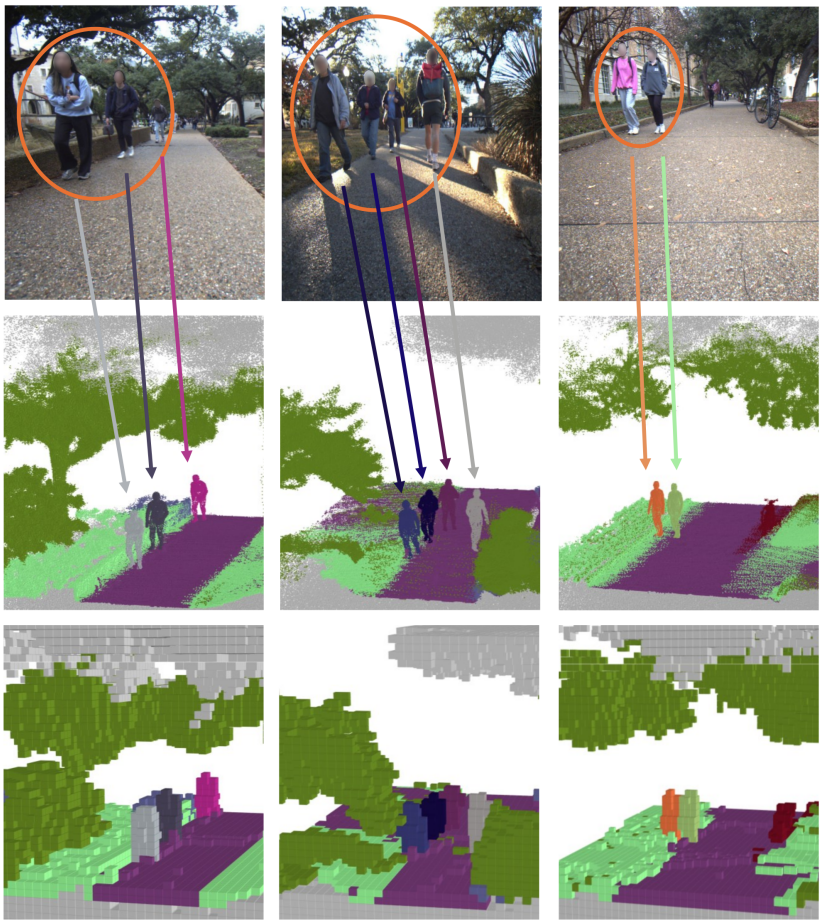


Figure A.2: Qualitative example from the *MobileOcc* benchmark illustrating occupancy and instance labels. Top: input images. Middle: semantic occupancy label at a fine resolution (e.g., 0.02 m). Bottom: semantic occupancy at the evaluation resolution used for benchmark evaluation (0.2 m). Gray voxels indicate unknown regions; free space is omitted for clarity.

Processing Pipeline

Pipeline overview. To maximize compatibility with published camera-based occupancy codebases, we convert the raw sequences into a nuScenes-style dataset structure and precompute the auxiliary signals required by different baselines. Starting from synchronized sensor logs (stereo images, LiDAR packets, calibrations, timestamps, poses) and released semantic occupancy grids, we (i) build nuScenes tables and token chains to enable consistent time alignment and coordinate transforms, (ii) preprocess occupancy labels (axis convention fix and voxel downsampling/cropping to the evaluation grid), (iii) derive pedestrian-centric supervision signals (instance tracks and planar velocities), and (iv) export per-sample metadata (`infos` files) used by common training pipelines. The overall workflow is summarized in Fig. A.3.

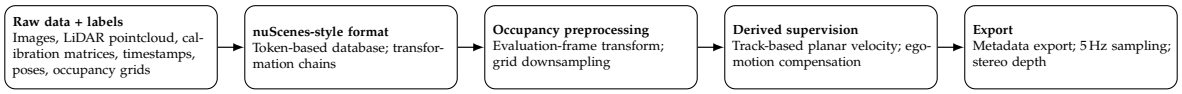


Figure A.3: Benchmark preparation pipeline. Convert sensor recordings and occupancy labels to a nuScenes-style format, preprocess occupancy grids, derive pedestrian velocities from tracks, and export training metadata.

Pedestrian velocity derivation and filtering. In addition to semantic occupancy, we derive planar pedestrian velocity targets (v_x, v_y) as an auxiliary supervision signal. Pedestrian instances are identified from instance IDs embedded in the occupancy volume and represented by their voxel-set centroid in metric coordinates. We associate centroids over time using a simple distance-gated temporal tracking

rule and estimate a smoothed constant planar velocity per track using local temporal windows. Since the robot platform is moving, we compensate velocities using ego poses from the standardized nuScenes token chain, correcting both linear motion and yaw-rate-induced apparent motion.

Because these velocity targets are *derived supervision* and can be unreliable under occlusions, short tracks, or label noise, we apply conservative filtering before export. In particular, we discard (i) pedestrian tracks with insufficient temporal support (very short tracks or large temporal gaps) and (ii) implausible motion estimates (e.g., speeds above a conservative threshold such as 3 m/s). We report the resulting compensated pedestrian motion statistics in Fig. A.4 as a sanity check.

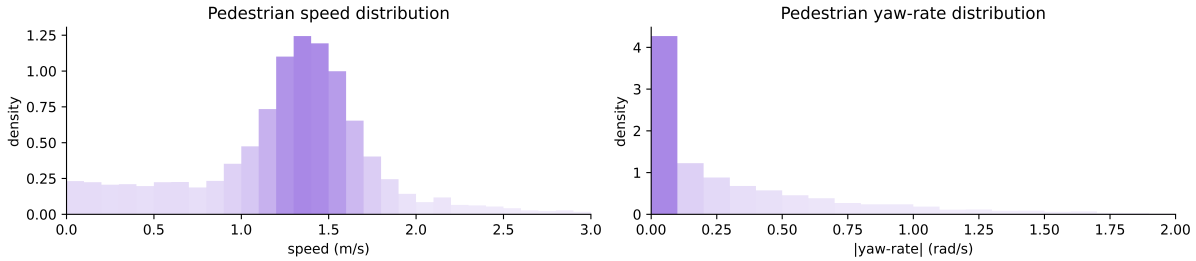


Figure A.4: Distributions of compensated and filtered pedestrian motion statistics. Left: linear speed. Right: angular velocity.

Data Sanitization

Frame-level quality control. Before constructing the final benchmark splits and training metadata, we apply conservative quality control to remove corrupted or unreliable intervals (e.g., sensor synchronization issues or severe LiDAR artifacts such as glass reflections). This filtering is performed once on the raw sequences and then kept fixed for all methods, ensuring that all baselines are trained and evaluated on the same set of valid frames. After filtering, the benchmark contains 116,511 frames at 5Hz.

Label-space sanitization. The provided semantic IDs follow a Cityscapes-like taxonomy and include categories that are either implausible in the target environment (e.g., *train*, *bus*) or too rare/ambiguous for stable voxel-level evaluation. To obtain a compact and consistent benchmark label space, we map irrelevant or inconsistent categories to *unknown* and merge visually confusable categories. In particular, we merge bicycles and motorcycles into a single *two-wheeler* class and group static structural categories into an *other-structure* class. The final benchmark taxonomy contains one free-space class, nine occupied semantic classes, and an unknown class.

Table A.3: Mapping from the original Cityscapes-style semantic IDs to the final benchmark taxonomy used for benchmark evaluation. Only representative and frequently occurring classes are listed; rare or implausible categories are mapped to *unknown*.

Original class	Benchmark class
Free	Free
Unknown	Unknown
Road	Road
Terrain	Terrain
Vegetation	Vegetation
Wall / Building / Fence	Other-structure
Pole / Traffic sign / Traffic light	Pole
Car	Car
Truck	Truck
Bicycle / Motorcycle	Two-wheeler
Bus / Train	Unknown
Person (instance-based)	Pedestrian

Sanitization effect. Fig. A.5 compares the class frequency distribution before and after label-space sanitization. Implausible categories are absorbed into the *unknown* label, while visually overlapping

classes (e.g., *bicycle/motorcycle*) and fine-grained static structures are consolidated into *two-wheeler* and *other-structure*, respectively. This redistribution reduces the long-tail sparsity of the original taxonomy and yields a more compact and consistent label space for evaluation.

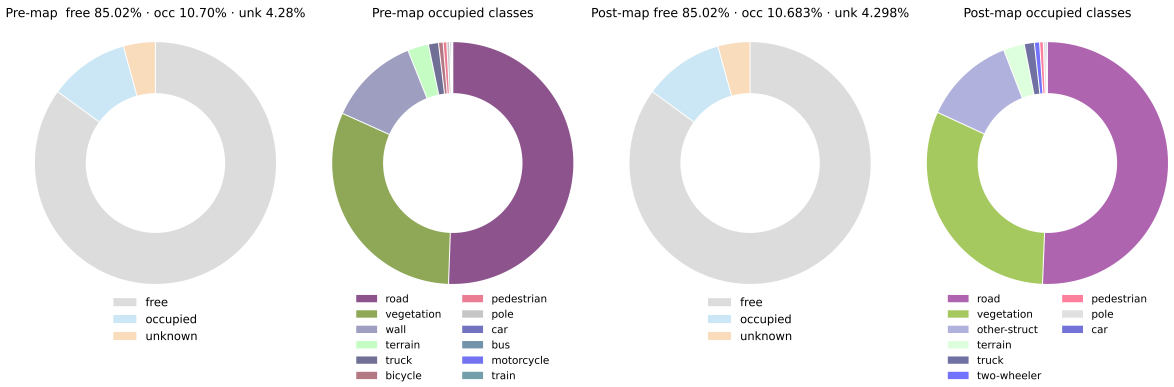


Figure A.5: Class distribution before and after label-space sanitization.

Split Design and Balancing

Motivation: avoid leakage while keeping evaluation stable. Occupancy sequences derived from [26] exhibit strong temporal and spatial correlation. A random frame-level split would place near-duplicate viewpoints in both training and validation, leading to over-optimistic results. We therefore split at the recording-sequence level and assign each sequence entirely to one split, which prevents recording-session leakage.

The dataset contains only four recorded places, so place overlap across splits is unavoidable. Rather than targeting unseen-site generalization, we use a sequence-disjoint validation split that approximately matches the overall place and illumination statistics for stable in-distribution comparison. Some coarse condition overlap remains due to the limited site diversity of the source locations, and we make this explicit in the sequence-level similarity analysis.

Balancing factors. We balance the split along two axes that substantially affect appearance and occupancy inference: (i) **illumination** and (ii) **place** (GDC, Guad, UNB, WCP). Figure A.6 shows representative appearance variation under different lighting and weather conditions.

Day (Sunny) **Day (Cloudy/Rainy)** **Dark**

Figure A.6: Example frames under different illumination conditions in the dataset. These examples illustrate representative appearance variation across day and dark scenes, including differences caused by weather.

Group-stratified split objective. We select the validation scenes by minimizing a simple weighted objective while keeping whole scenes intact:

$$\mathcal{L}_{\text{split}} = 5\big(\hat{r} - 0.20\big)^2 + 1.5 \sum_{b\in\{\text{Day,Dark}\}} \big(\hat{p}_b - p_b\big)^2 + \sum_{m\in\{\text{GDC,Guad,UNB,WCP}\}} \big(\hat{p}_m - p_m\big)^2, \tag{A.5}$$

where $\hat{r}$ is the validation frame ratio, p_b and p_m denote the overall (post-drop) proportions of illumination condition b and place m, and $\hat{p}_b$ and $\hat{p}_m$ denote the corresponding proportions within the validation

split. We enumerate candidate scene subsets, keep whole scenes intact, and select the lowest-loss candidate among those with validation size constrained to 15–25% of all frames.

Resulting split. After filtering and dropping invalid frames, the dataset contains 233,025 frames, of which 184,608 are used for training and 48,417 for validation (20.78%). Table A.4 reports the resulting statistics, showing that the validation split closely matches the overall place and illumination distributions. Training and validation remain disjoint at the recording-sequence level, preventing recording-session leakage. Figure A.7 summarizes the sequence-level assignments through a coarse metadata similarity analysis.

Table A.4: Scene-level split summary and distribution matching. "Total" refers to all frames after dropping invalid samples; "Train" and "Val" are the resulting subsets.

Metric	Total	Train	Val
Frames (post-filter)	233,025	184,608	48,417
Ratio in total	–	0.7922	0.2078
Dark ratio	0.1814	0.1789	0.1906
Place ratio			
GDC	0.2357	0.2349	0.2385
Guad	0.4750	0.4829	0.4447
UNB	0.0255	0.0123	0.0757
WCP	0.2639	0.2698	0.2412

Figure A.7: Sequence-level train–validation similarity heatmap based on coarse capture metadata. Similarity is a discrete score $s \in \{0, 1, 2, 3\}$: $s = 3$ if place, trajectory direction, and illumination–weather label all match; $s = 2$ if place matches and exactly one of direction or illumination–weather matches; $s = 1$ if only place matches; and $s = 0$ otherwise. Disjoint sequence identifiers prevent recording-session leakage, while limited site diversity leads to some residual coarse-metadata overlap between training and validation sequences.

A.4.2. Baseline Adaptation for Mobile-Robot Front-View Stereo

This subsection summarizes the representative published baselines evaluated for this task and the minimal adaptations required to apply driving-centric occupancy frameworks to our mobile-robot benchmark. Most public occupancy codebases assume (i) surround-view multi-camera rigs and (ii) target occupancy volumes designed for longer-range autonomous driving, whereas our benchmark uses a front-view stereo setup and a near-field target occupancy volume. We therefore (i) reduce multi-camera inputs to a single front-view stream (or stereo pair when required) and (ii) align the target occupancy volume to our benchmark, while keeping the original model designs and losses unchanged whenever possible to ensure fair and reproducible comparison.

Adaptation Principles

We follow three principles when porting baselines to our benchmark: (i) **minimal deviation** from the official implementations (no architectural changes unless required by input/output interfaces), (ii) **consistent evaluation target** (all methods predict the same target occupancy volume and label space defined in Section A.4.1), and (iii) **comparable temporal coverage** across methods whenever feasible.

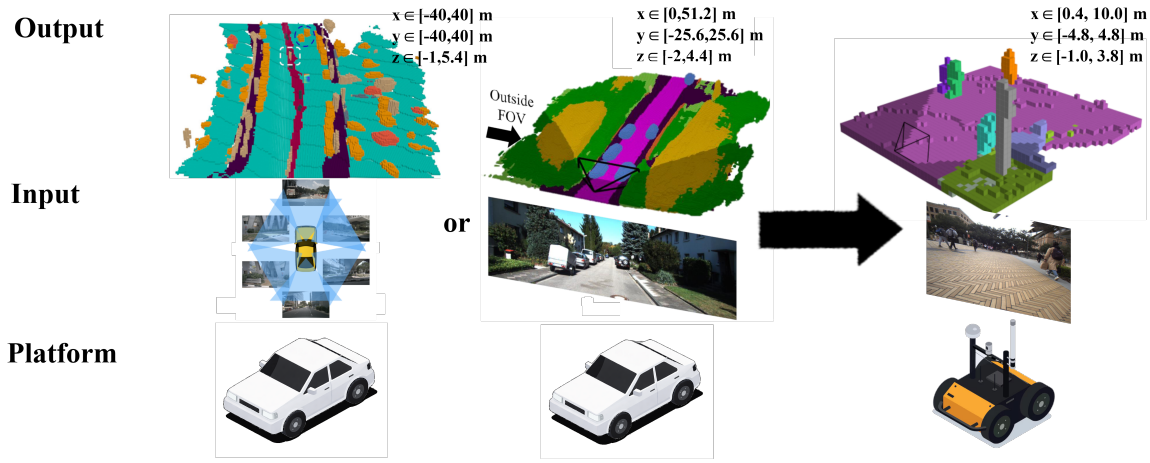


Figure A.8: Overview of baseline adaptations for the *MobileOcc* benchmark.

Monocular BEV-Based Baselines

BEVDet4D. We include BEVDet4D [29], originally proposed for BEV-based 3D detection, as a representative monocular BEV lifting pipeline and as an initialization source for downstream occupancy heads. The original implementation is designed for multi-view surround-camera inputs; in our setting, we evaluate a monocular variant by using only the left stereo camera stream for depth estimation and BEV lifting. Apart from this input formatting change and the alignment of the target occupancy volume to our benchmark setup, we retain the original model components and supervision signals.

FlashOcc and Panoptic-FlashOcc. We evaluate FlashOcc [14] and Panoptic-FlashOcc [24] as semantic and panoptic occupancy baselines. Similar to BEVDet4D, these methods assume surround-view multi-camera inputs and are adapted to our front-view setting by operating on a single camera stream (left stereo camera) while keeping their BEV lifting and occupancy decoding pipelines unchanged. We use temporal input queues (8 historical frames) following their public settings and align the occupancy prediction volume and semantic head to the benchmark target occupancy volume and label space.

Stereo Baseline

We evaluate VoxFormer-T [17], the temporal variant of VoxFormer using 4 historical frames, as a stereo occupancy baseline. VoxFormer employs a two-stage pipeline: Stage-1 consumes voxelized depth cues to produce a coarse 3D occupancy proposal, and Stage-2 refines this proposal using transformer-based volumetric decoding. The public implementation assumes an available learned stereo depth module to provide Stage-1 depth inputs [30] which was only pretrained on SemanticKITTI; to match our sensor setup and keep the remaining architecture unchanged, we replace this dependency with stereo matching on the front stereo pair, followed by back-projection and voxelization to produce the required depth voxel input. Stage-1 predicts a coarse occupancy volume (0.4 m resolution in the public setup), which is then aligned and upsampled to our 0.2 m evaluation volume for training and evaluation. To maintain comparable temporal coverage with the monocular baselines using 8 historical frames, we use 4 historical frames sampled alternately in time.

Panoptic Occupancy with Velocity

To jointly evaluate panoptic occupancy and pedestrian motion prediction, we introduce *Panoptic-FlashOcc-vel*, a lightweight extension of Panoptic-FlashOcc that adds planar pedestrian velocity prediction (v_x, v_y). Concretely, we augment the original detection/panoptic head with a velocity regression branch trained with an ℓ_1 loss using the tracking-derived velocity targets described in Section A.4.1. All other components of Panoptic-FlashOcc remain unchanged, enabling joint evaluation of voxel-level panoptic occupancy and pedestrian motion.

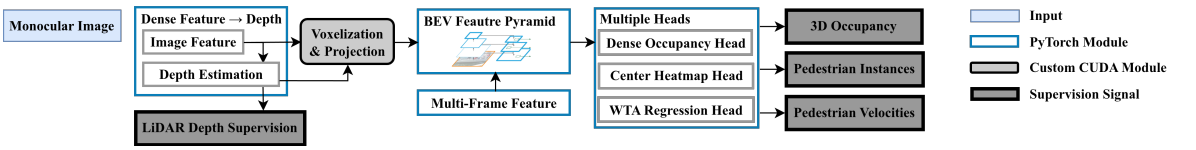


Figure A.9: Architecture for *Panoptic-FlashOcc-vel*. We extend the original Panoptic-FlashOcc head with a Winner-Takes-All (WTA) pedestrian velocity regression branch that predicts planar velocity components (v_x, v_y), supervised with an ℓ_1 loss. The occupancy and panoptic heads remain unchanged.

A.5. Experiments

A.5.1. Experimental Setup

We evaluate all occupancy prediction methods on the *MobileOcc* benchmark described in Section A.4. Unless otherwise stated, all predictions and metrics are defined in the ego frame $\mathcal{F}_e$ within the bounded local region $\mathcal{R}$ following Section A.3.

Evaluation grid and data splits. We use a voxel resolution of 0.2 m and evaluate occupancy within the grid spanning $x \in [0.4, 10.0]$ m, $y \in [-4.8, 4.8]$ m, and $z \in [-1.0, 3.8]$ m. The raw sequences are downsampled to 5 Hz by taking every second frame. After filtering and downsampling, the training split contains 92,303 samples and the validation split contains 24,208 samples, corresponding to 79.22% and 20.78% of the total data, respectively. Among these, 30,457 training frames and 7,165 validation frames include pedestrian annotations.

Benchmark protocol, methods, and metrics. We follow the *MobileOcc* benchmark specification for label space and baseline adaptations described in Section A.4. In brief, we evaluate semantic occupancy, panoptic occupancy for pedestrian instances, and pedestrian planar velocity prediction using representative monocular BEV lifting and occupancy decoding baselines (BEVDet4D [29], FlashOcc [14], Panoptic-FlashOcc [24]) and a stereo occupancy baseline (VoxFormer-T [17]), as well as our Panoptic-FlashOcc-vel extension.

We report geometric IoU and semantic mIoU for voxel occupancy (Section A.3.2), and panoptic metrics (PQ and $\text{PQ}^{\dagger}$) for voxel-level panoptic occupancy. Pedestrian detection performance is reported using Average Precision AP^{Ped} computed from predicted pedestrian instance centers under multiple center-distance thresholds (0.1 m, 0.2 m, 0.5 m, and 1.0 m), following the benchmark protocol used in the *MobileOcc* paper. For pedestrian velocity prediction, we report the mean absolute velocity error (AVE) under three subsets: **AVE-T** over all ground-truth pedestrian voxels, **AVE-D** over matched true-positive pedestrian detections (center distance ≤ 1 m), and **AVE-O** over correctly classified pedestrian-occupied voxels.

A.5.2. Main Results

Semantic Occupancy Prediction

Method	IoU	mIoU	pedestrian	car	other struct.	pole	road	terrain	truck	two-wheeler	vegetation
VoxFormer-T [17]	57.81	24.89	**32.79**	1.39	28.47	7.08	**70.90**	23.57	5.23	28.04	26.52
FlashOcc (8f) [14]	57.71	**27.18**	31.91	**5.36**	31.00	**10.16**	70.82	**27.29**	**7.35**	30.85	**29.88**
Panoptic-FlashOcc (8f) [24]	**57.83**	26.64	32.45	3.66	**31.79**	9.96	70.35	25.90	5.87	**30.99**	28.83

Table A.5: 3D occupancy prediction performance. The best results are shown in **bold**. IoU denotes the geometric IoU.

Table A.5 reports semantic occupancy performance on the validation split. Among the evaluated methods, FlashOcc achieves the highest mIoU and performs best on most semantic categories. Panoptic-

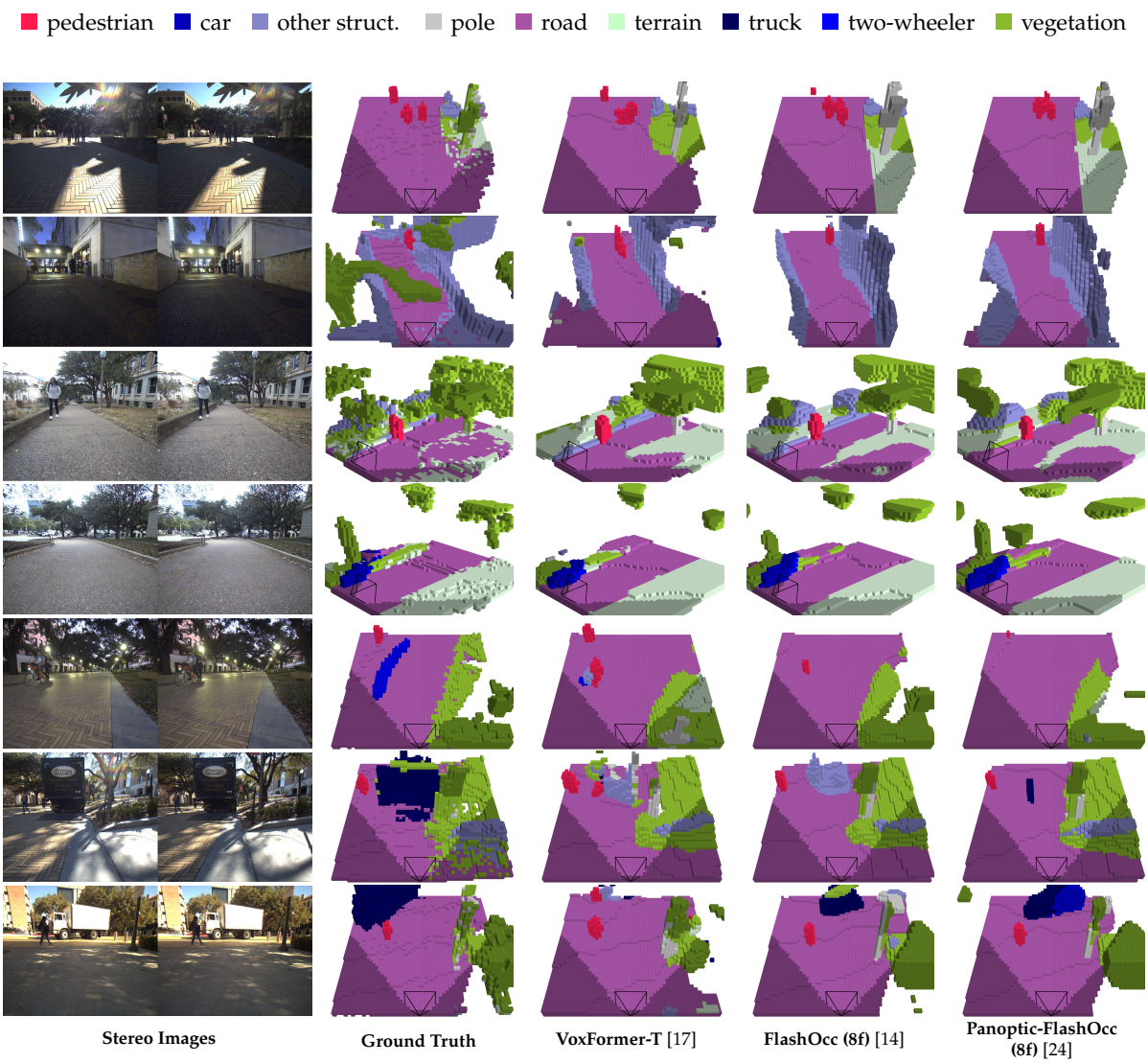


Figure A.10: Qualitative comparison of different baselines under diverse lighting conditions (sunny, night, and cloudy). Performance differences are most visible on pedestrians and the overall scene layout, while some rare categories and fast-moving objects can appear inconsistent due to limited training frequency and motion blur.

FlashOcc yields a minor mIoU drop relative to FlashOcc, consistent with the expected trade-off introduced by joint semantic and panoptic prediction. VoxFormer-T, which leverages stereo depth cues, remains competitive on pedestrians and achieves the best pedestrian IoU among the three baselines.

Figure A.10 provides qualitative comparisons across diverse lighting conditions (sunny, night, and cloudy). Across methods, road and major occupied structures are recovered consistently in the near-field region, while fine-grained categories such as thin poles or rare vehicle classes remain more challenging.

Panoptic Occupancy and Pedestrian Detection

We evaluate pedestrian detection and panoptic occupancy quality using BEVDet4D and Panoptic-FlashOcc. As shown in Table A.6, BEVDet4D provides pedestrian detections, while Panoptic-FlashOcc predicts voxel-level panoptic labels, enabling evaluation under panoptic metrics. Besides standard PQ, we report $PQ^{\dagger}$ as explained in Section A.3.2, which relaxes the strict IoU matching requirement for stuff classes and is more indicative for vision-only occupancy prediction.

Figure A.11 visualizes Panoptic-FlashOcc predictions, including the predicted semantic occupancy and separated pedestrian instances. The instance-level representation for pedestrians is a key capability for pedestrian motion estimation and complements semantic occupancy prediction.

Method	PQ	PQ†	RQ	SQ	PQ^{Ped}	RQ^{Ped}	SQ^{Ped}	AP^{Ped}
BEVDet4D (8f) [11]	–	–	–	–	–	–	–	41.7
Panoptic-FlashOcc (8f) [24]	19.9	28.1	65.8	32.6	42.5	60.2	70.7	45.5

Table A.6: Baseline performance on 3D panoptic occupancy.

Pedestrian Velocity Prediction
We compare BEVDet4D and Panoptic-FlashOcc-vel on pedestrian velocity prediction. Table A.7 reports AVE under the three evaluation subsets (AVE-T, AVE-D, and AVE-O), together with mIoU for semantic occupancy. Panoptic-FlashOcc-vel achieves comparable velocity accuracy while simultaneously providing voxel-level semantic occupancy. Figure A.12 shows qualitative velocity predictions; the results are visually consistent for most pedestrians, while occasional direction ambiguity (e.g., forward/backward) remains a challenging case.

Method	AVE-T↓	AVE-D↓	AVE-O↓	mIoU↑
BEVDet4D (8f) [29]	1.00	0.36	–	–
Panoptic-FlashOcc-vel (8f) [24]	0.97	0.39	0.67	26.00

Table A.7: Comparison of pedestrian absolute velocity error (AVE).

A.5.3. Qualitative Discussion

Figures A.10–A.12 illustrate that the evaluated vision-based occupancy pipelines recover the overall near-field scene layout consistently across diverse illumination conditions. In particular, pedestrian occupancy and instance separation remain visually stable in many scenes, including cases with pedestrians at relatively large distances, supporting the benchmark emphasis on human-aware perception for mobile robots. At the same time, certain static categories (e.g., thin vertical structures or vegetation-like regions) can be more ambiguous in vision-based occupancy prediction and may be further affected by annotation noise introduced by automatic labeling.

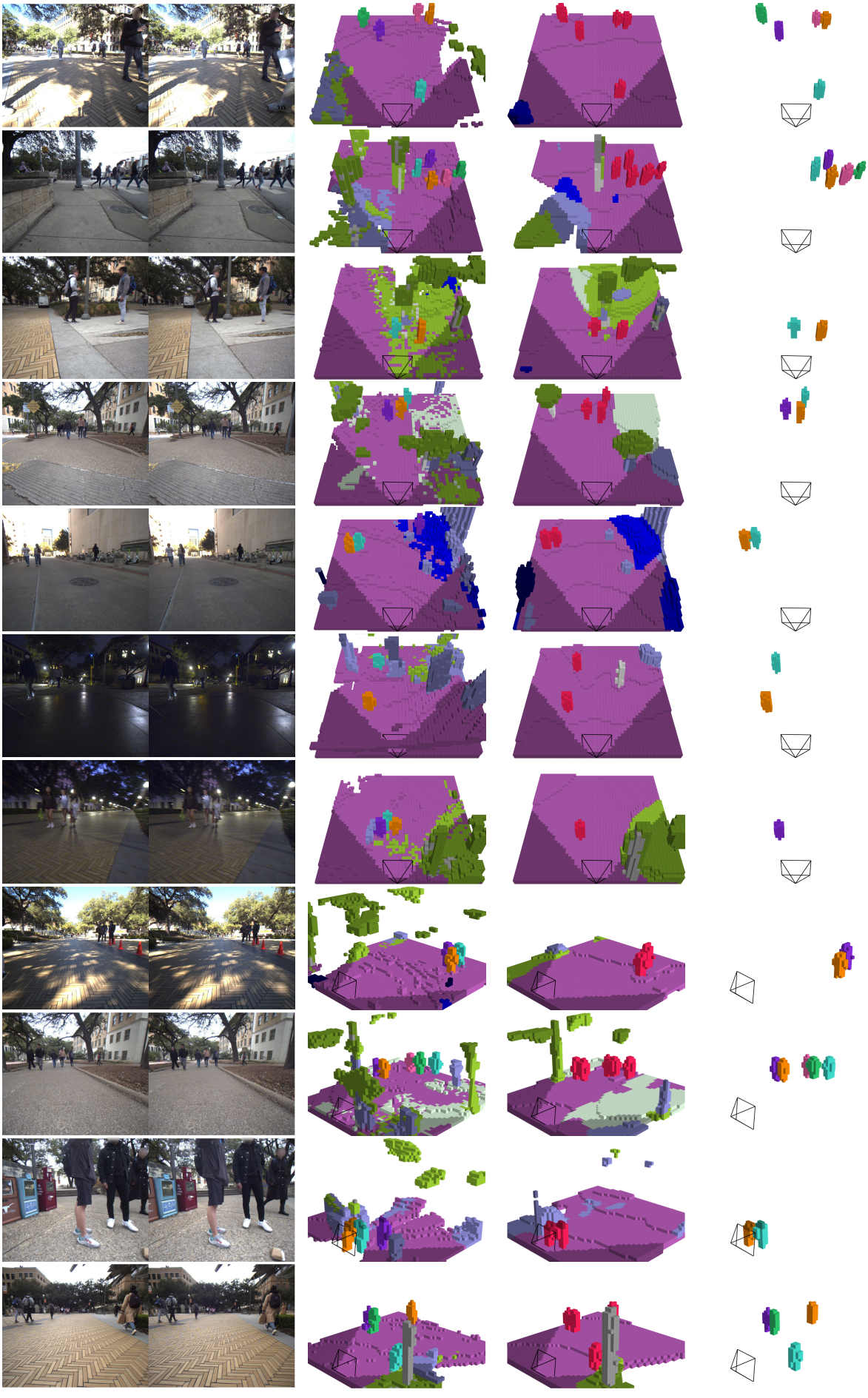


Figure A.11: Panoptic occupancy prediction performance using Panoptic-FlashOcc (8f) [24].

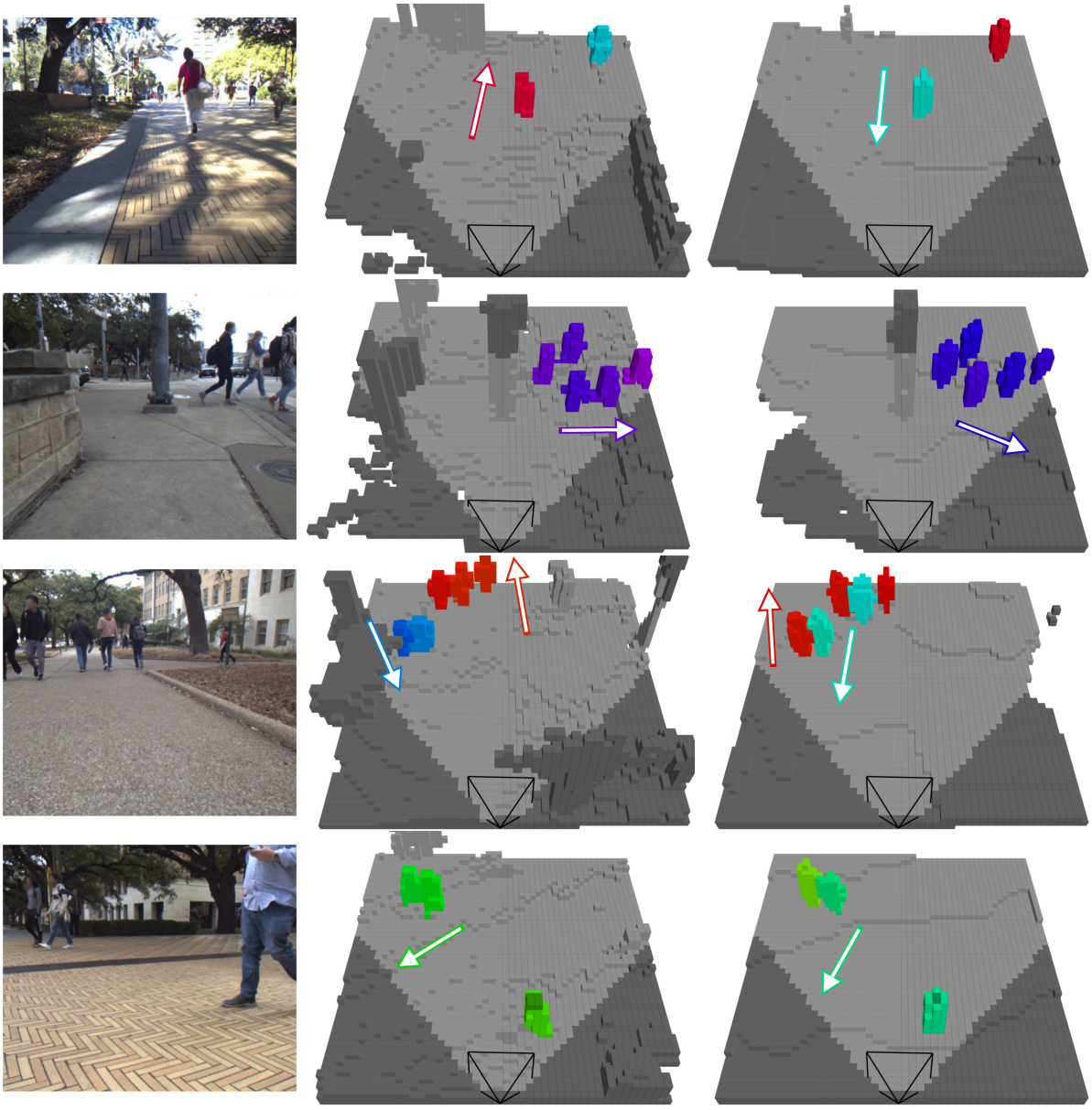


Figure A.12: Pedestrian velocity prediction using Panoptic-FlashOcc-vel (8f). The directions are shown in colors and arrows.

References for Appendix

[1] Yanan Zhang et al. “Vision-based 3d occupancy prediction in autonomous driving: a review and outlook”. In: *Frontiers of Computer Science* 20.1 (2026), p. 2001301.

[2] Xiaoyu Tian et al. “Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 64318–64330.

[3] Wenwen Tong et al. “Scene as occupancy”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 8406–8415.

[4] Yiming Li et al. “Sscbench: A large-scale 3d semantic scene completion benchmark for autonomous driving”. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 13333–13340.

[5] Jens Behley et al. “Semantickitti: A dataset for semantic scene understanding of lidar sequences”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9297–9307.

[6] Yuqi Wang et al. “Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 17158–17168.

[7] Ziyue Zhu et al. “Voxelsplat: Dynamic gaussian splatting as an effective loss for occupancy and flow prediction”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 6761–6771.

[8] Yuqi Wu et al. “Embodiedocc: Embodied 3d occupancy prediction for vision-based online scene understanding”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2025, pp. 26360–26370.

[9] Hao Shi et al. “OneOcc: Semantic Occupancy Prediction for Legged Robots with a Single Panoramic Camera”. In: *arXiv preprint arXiv:2511.03571* (2025).

[10] Jonah Philion and Sanja Fidler. “Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d”. In: *European conference on computer vision*. Springer. 2020, pp. 194–210.

[11] Junjie Huang et al. “Bevdet: High-performance multi-camera 3d object detection in bird-eye-view”. In: *arXiv preprint arXiv:2112.11790* (2021).

[12] Yinhao Li et al. “Bevdepth: Acquisition of reliable depth for multi-view 3d object detection”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 37. 2. 2023, pp. 1477–1485.

[13] Yinhao Li et al. “Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 2. 2023, pp. 1486–1494.

[14] Zichen Yu et al. “Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin”. In: *arXiv preprint arXiv:2311.12058* (2023).

[15] Zhiqi Li et al. “Bevformer: learning bird’s-eye-view representation from lidar-camera via spatiotemporal transformers”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[16] Yuan-Ko Huang et al. “Tri-Perspective View for Vision-Based 3D Semantic Occupancy Prediction”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), pp. 9223–9232.

[17] Yiming Li et al. “Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 9087–9098.

[18] Zhiqi Li et al. “Fb-occ: 3d occupancy prediction based on forward-backward view transformation”. In: *arXiv preprint arXiv:2307.01492* (2023).

[19] Pin Tang et al. “Sparseocc: Rethinking sparse latent representation for vision-based semantic occupancy prediction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 15035–15044.

[20] Yunpeng Zhang, Zheng Zhu, and Dalong Du. "Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 9433–9443.

[21] Wanshui Gan et al. "GaussianOcc: Fully Self-supervised and Efficient 3D Occupancy Estimation with Gaussian Splatting". In: *ArXiv* abs/2408.11447 (2024).

[22] Simon Boeder, Fabian Gigengack, and Benjamin Risse. "Gaussianflowocc: Sparse and weakly supervised occupancy estimation using gaussian splatting and temporal flow". In: *arXiv preprint arXiv:2502.17288* (2025).

[23] Haoyi Jiang et al. "Gausstr: Foundation model-aligned gaussian transformer for self-supervised 3d spatial understanding". In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 11960–11970.

[24] Zichen Yu et al. "Panoptic-flashocc: An efficient baseline to marry semantic occupancy with panoptic via instance center". In: *arXiv preprint arXiv:2406.10527* (2024).

[25] Junseo Kim et al. "MobileOcc: A Human-Aware Semantic Occupancy Dataset for Mobile Robots". In: *arXiv preprint arXiv:2511.16949* (2025).

[26] Arthur Zhang et al. "Towards Robust Robot 3D Perception in Urban Environments: The UT Campus Object Dataset". In: *arXiv preprint arXiv:2309.13549* (2023).

[27] Lorenzo Porzi et al. "Seamless scene segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 8277–8286.

[28] Holger Caesar et al. "nuscenes: A multimodal dataset for autonomous driving". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11621–11631.

[29] Junjie Huang and Guan Huang. "Bevdet4d: Exploit temporal cues in multi-camera 3d object detection". In: *arXiv preprint arXiv:2203.17054* (2022).

[30] Faranak Shamsafar et al. "Mobilestereonet: Towards lightweight deep networks for stereo matching". In: *Proceedings of the ieee/cvf winter conference on applications of computer vision*. 2022, pp. 2417–2426.