TriO: Tri-Modal Unsupervised Occupancy World Model for Anything Perception

Quinlan Sykora*[1,2], Sourav Biswas*[1,2], Christopher Diehl*[1], Andrew Cunningham[1], Thomas Gilles[1], and Raquel Urtasun[1]

[1] Waabi
[2] University of Toronto,
{qsykora,sbiswas,cdiehl,acunningham,tgilles,urtasun}@waabi.ai

Abstract. We present TriO, a multi-modal unsupervised world model that predicts 4D occupancy, obstacle segmentation, flow and LiDAR. In contrast to prior work, TriO utilizes three distinct sensor modalities (camera, LiDAR, and RADAR) as both inputs and sources of self-supervision, eliminating the need for additional human annotations. Thanks to its novel supervision, the model is able to segment any occupancy from the drivable surface, overcoming the limitations of existing open-set methods in handling long-tail objects. TriO achieves state-of-the-art results in multiple 3D and 4D tasks, including occupancy, flow, and LiDAR prediction, as well as zero-shot road obstacle segmentation across multiple datasets such as Argoverse 2, and Spotting the Unexpected.

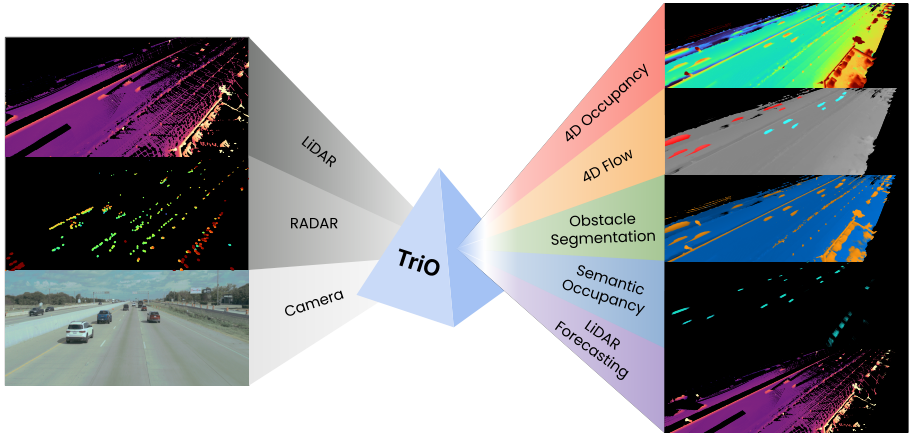

Fig. 1: We present TriO, an unsupervised tri-modal occupancy world model designed to perceive and forecast anything that could be considered path-blocking.

* denotes equal contribution

1 Introduction

In order for a self-driving vehicle (SDV) to safely navigate through complex environments, it must first *perceive anything*, i.e. segment above ground obstacles from drivable areas at the current time, then forecast the evolution of the scene to plan its future motion. This problem has been predominantly tackled with object-based approaches [10, 11, 13, 16, 34, 35, 53, 81, 92, 93], which first detect objects and then forecast their future trajectories. In contrast, object-free occupancy approaches [2,12,17,31,62,74,80] overcome the downsides of thresholding detections and the limited representation capability of a finite set of trajectories.

However, most 3D/4D occupancy approaches [25, 26, 61, 103, 107] rely on dense voxel grid labels [84, 91] which are computationally expensive at high resolutions, and are constructed with human annotations (e.g., semantic LiDAR point annotations or bounding box labels). Due to the cost of the latter, and their difficulty to generate at scale, the volume of the labeled data is commonly orders of magnitude smaller than the amount of unlabeled data [1]. Moreover, the mentioned approaches are usually supervised with a finite set of semantic classes, which do not encapsulate safety-critical rare events.

Conversely, recent occupancy approaches [1, 18, 43] leverage LiDAR self-supervision, pre-training on vast amounts of sensor data and fine-tuning with a small amount of labeled data for downstream applications. However, LiDAR is susceptible to blooming [86], multipath returns [78], and limited range. Additional sensor modalities can be of benefit for self-supervised training, and multi-sensor fusion can increase robustness to sensor failures and adverse conditions.

Other methods have leveraged RGB cameras, either as inputs [71], distilling foundation model features [60], or predicting semantic pseudo-labels [7,8,22,33, 48,54] from camera-based models. Furthermore, many prior works [38,39,76,97, 99] generate semantic labels with open-vocabulary segmentation models queried with text-prompts. However, common failure modes have been shown in open-set detectors [3] and segmentors [89], challenging their overall robustness to rare classes. Finally, the segmentation models require image-text pair annotations.

One generally overlooked modality is RADAR, which can enhance training by introducing radial velocity measurements, improve robustness under adverse weather conditions [49], and increase the observation range. While some works investigate RADAR-based 3D occupancy models using human annotations [20, 73], and others focus on RADAR-only input with self-supervision [46] or cross-modal LiDAR supervision [57], none integrate all aforementioned modalities in a unsupervised tri-modal 4D world model.

Motivated by these downsides, our goal is to develop a model that can leverage all three common sensor modalities (LiDAR, camera, and RADAR) as both input and supervision for occupancy in a self-supervised manner. Towards this goal, we propose TriO: a multi-modal world model that can predict and forecast 4D continuous occupancy, obstacle segmentation, and flow at any continuous (x, y, z, t) coordinate from tri-modal observations. To achieve this, we leverage each sensor as both an input and as a unique form of supervision, ultimately combining the strengths of each modality to produce the best possible represen-

tation of the world. We further introduce a novel multi-sensor fusion architecture leading to better forecasting capabilities. Notably, being pre-trained on a vast amount of raw sensor data, TriO can be effectively leveraged for multiple downstream applications.

Historically, to determine which entities in the scene are relevant to the self-driving task, prior work uses semantic supervision from open-vocabulary models prompted with a closed set of multiple classes [38, 39, 76, 97, 99]. These however do not generalize to long tail events with rare obstacles, and hence do not cover all safety-critical scenarios. We argue the most general and robust requirement for occupancy is to semantically distinguish between what is *occupied but not path-blocking* (e.g., a flat surface that is traversable by the SDV) vs. *occupied and path-blocking* (e.g., a rare object on the road). Towards this end, we define the task of **anything-perception** as the ability to segment any object that exists above the road from the ground surface.

Despite demonstrating impressive open-set detection capabilities, the segmentation models of prior work used for 2D $\rightarrow$ 3D label distillation struggle to segment rare objects [89] (the long tail), even when prompted with the ground truth class, highlighting a lack of semantic understanding and generalization in current VLMs. To circumvent this, we propose a simple yet effective unsupervised[1] *geometric image/LiDAR segmentation* method that leverages representations from image depth model Metric3Dv2 [32], and the LiDAR segmentation heuristic from Patchwork++ [47]. Our method detects any obstruction, leading to significant improvements in generalizable unknown object segmentation. Notably, in contrast to other vision-language based approaches which require at least textual image descriptions [4, 39, 76, 85, 97] or (semantic) mask labels [38, 45, 97, 99], our obstacle segmentation model relies solely on geometric properties rather than semantic annotations.

Contributions: 1) TriO uniquely uses all three common sensor modalities (camera, LiDAR, RADAR), both as input and for self-supervision. **2)** We introduce a novel image/LiDAR-based obstacle segmentation technique that generates pseudo-labels for distillation into TriO, without requiring additional human labels in our training pipeline, while maintaining robustness and generalizability. **3)** We propose a novel architecture fusing all modalities leading to better long-term forecasting. **4)** TriO achieves state-of-the-art performance on a variety of tasks including point cloud forecasting, obstacle segmentation, and both 4D/3D occupancy and flow forecasting, validated on three real-world autonomous driving datasets (*Argoverse2*, *Spotting the Unexpected*, *Long Range RADAR*). Overall, the novelty of our contribution is the unique integration of all three sensor modalities as inputs, and as source of self-supervision for our output representations.

[1] We note that Metric3Dv2 [32] was partially trained on datasets with semi-supervised human annotations. In this context, we define *unsupervised* to mean not requiring direct human annotations. For more discussion refer to the supplementary.

2 Related Work

Semantic Occupancy. The task of *scene completion* [40,66,98,106] involves reconstructing a complete scene from sparse sensor measurements. *Semantic scene completion* [5, 14, 79, 95] further extends this by aiming to infer semantic classes for each completed region, a task closely related to *semantic occupancy prediction*. Contemporary methods in this domain [25,26,61,103,107] typically focus on predicting 3D or 4D occupancy voxel grids with their associated semantic labels. These approaches generally require dense voxel-level supervision [84, 91], which is often derived from costly human-annotated labels, such as 3D bounding boxes or semantic point clouds. More recently, several approaches [7, 9, 54, 100] have proposed distilling 2D pseudo-semantic labels from VLMs. However, these VLMs themselves rely on large-scale human annotations in the form of image-text pairs, and supervising with a closed set of classes inherently restricts generalization in open-world scenarios.

Anything Perception. Open-vocabulary segmentation methods perform classification via text prompts at inference time, overcoming the limitations of fixed, closed-set label spaces. In the 2D domain, two primary paradigms have emerged: one leverages open-set detectors like GroundingDINO [58] or OWLv2 [64] to prompt mask predictors such as SAM [45], while another [38, 97, 99] predicts mask proposals and classifies them by computing the similarity between mask embeddings and CLIP-style [75] text embeddings. These paradigms have been extended to the 3D domain through methods like SAL [69], which distills CLIP features for LiDAR segmentation. Similarly, recent open-vocabulary occupancy methods [37,48,90,102] distill these features into voxel representations to enable open-world prompting. Despite their flexibility, these methods face two critical challenges: (1) they remain fundamentally dependent on models pre-trained on vast image-text and per-pixel mask annotations, and (2) they exhibit significant performance degradation for rare classes [3, 89]. In contrast, TriOs supervision relies on geometric cues from different sensors, enabling a more reliable segmentation of any object relative to the drivable surface. While ground segmentation has been extensively studied for LiDAR point clouds [23, 47, 70, 82], its application to occupancy remains limited. For example, [77] utilizes surface normals from stereo cameras and LiDAR to estimate ground regions *in current time BEV (2D)* occupancy maps, yet is constrained by the range of current measurements and segmentation networks that require pixel-level human annotations. In comparison, we distill fused obstacle segmentation estimates from LiDAR and monocular camera data and TriO has a unique ability of *forecasting these segmentations in 4D*.

Unsupervised Occupancy. Early non-learning-based methods [15, 68, 83] employed classical filtering techniques to estimate occupancy and flow without human annotations. With the advent of deep learning, 4D-Occ [43] proposed a LiDAR-only self-supervised approach to forecast 4D occupancy. However, its explicit 4D voxel grid representation suffers from significant memory and runtime inefficiencies, often forcing a trade-off between spatial resolution and reconstruction quality. To address these limitations, recent implicit occupancy

methods [1, 18, 54, 60, 71] represent scenes in continuous space-time, leveraging self-supervision from LiDAR or camera inputs. Departing from implicit representations, SceneDino [36] predicts 3D geometry and feature fields from monocular input, achieving semantic scene completion through clustering and unsupervised distillation. Other recent works, such as [57], adopt cross-modal strategies using RADAR as input with LiDAR-based supervision. In contrast to all prior work, TrIO utilizes camera, LiDAR, and RADAR data for both input and self-supervision, exploiting the inherent physical synergies between sensors to perform 4D forecasting of occupancy, segmentation, and flow. We show an extensive tabular comparison and further related works in the supplementary.

3 Unsupervised Multi-Modal Occupancy-Flow Method

To learn a robust representation of the world, we leverage the complementary physical properties of the standard SDV sensor suite: active sensors (LiDAR and RADAR) operating on Time-of-Flight principles, and passive cameras. While LiDAR provides high-precision 3D geometry capable of capturing free-space [1] and object surfaces [47], its sparse returns degrade at distance and in adverse weather. In contrast, RADAR uses longer wavelengths that penetrate fog and rain, and provide instantaneous radial velocity via the Doppler effect. Passive RGB cameras offer dense, high-resolution semantic and textural information, but lack native depth. To overcome this limitation, we extract dense depth and surface normal estimates using recent vision foundation models [32], enabling the reliable segmentation of obstacles. In summary, we combine the complementary synergies of each sensor modality and propose TrIO, a 4D world model that learns geometry, dynamics, and object segmentation while remaining unsupervised.

3.1 Unsupervised Task

To define our training objective, let $T \in \mathbb{N}$ denote the future time-horizon and $H \in \mathbb{N}$ the historical observation time steps. For notational clarity, in the following we define $\mathbf{S}_\mathrm{H} = \mathbf{S}_{-H,\dots,0}$ as the historical sensor observations at time $-H, \dots, 0$ and $\mathbf{S}_\mathrm{T}$ as the future observations at time $0, \dots, T$. Here, $\mathbf{S}$ is a placeholder for either multi-view camera images $\mathbf{C} \in \mathbb{R}^{N_\mathrm{C} \times W \times H \times 3}$, LiDAR points $\mathbf{L} \in \mathbb{R}^{N_\mathrm{L} \times d_\mathrm{L}}$, or RADAR targets $\mathbf{R} \in \mathbb{R}^{N_\mathrm{R} \times d_\mathrm{R}}$. N_C denotes the number of camera images with width W and height H. Further, N_L, N_R, d_L, d_R are the number of LiDAR/RADAR points and their feature dimensionalities, respectively. Finally, we define the combined historical and future sensor observations as $\mathbf{X}_\mathrm{H} = (\mathbf{C}_\mathrm{H}, \mathbf{L}_\mathrm{H}, \mathbf{R}_\mathrm{H})$ and $\mathbf{X}_\mathrm{T} = (\mathbf{C}_\mathrm{T}, \mathbf{L}_\mathrm{T}, \mathbf{R}_\mathrm{T})$, respectively. Our goal is to learn a model:

$$\mathbf{y} = (o, s, \mathbf{f}) = f_\theta(\mathbf{X}_\mathrm{H}, \mathbf{q}), \tag{1}$$

parameterized by θ, that can predict occupancy probability $o \in [0, 1]$, obstacle segmentation probability $s \in [0, 1]$ and 3D flow $\mathbf{f} \in \mathbb{R}^3$ at *query point* $\mathbf{q} = (q_x, q_y, q_z, q_t)$ in *continuous* 4D space-time (x, y, z, t).

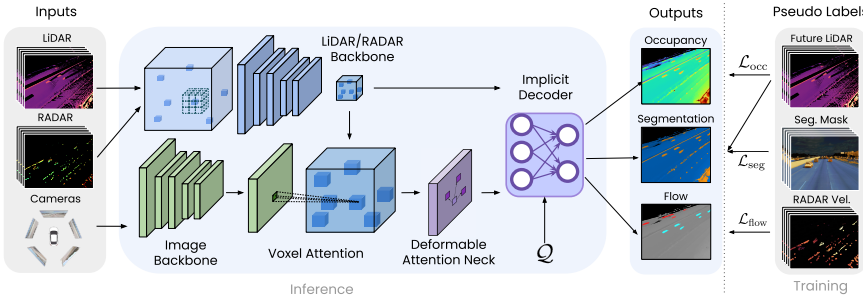

Fig. 2: TrIO's multi-modal architecture fusing LiDAR, RADAR, and camera inputs via voxel-attention. An implicit decoder generates occupancy, segmentation, and flow for a set of query points $\mathcal{Q}$ supervised by pseudo-labels derived from sensor data.

Geometric occupancy and 3D flow offer a simple representation that captures important characteristics of real-world environments (being occupied by a physical object and its velocity). However, a motion planner must distinguish between what is path-blocking (e.g., a child sitting on the street) and what is traversable (e.g., a drivable slope), motivating the prediction of obstacle segmentation s. To train our method, we require labels for each query point $\mathbf{q}$. We construct supervisory signals (pseudo-labels) as $\bar{\mathbf{y}} = g(\mathbf{X}_\mathrm{T}, \mathbf{q})$, which are used in the loss function $\mathcal{L}(\mathbf{y}, \bar{\mathbf{y}})$.

In the remainder of this section, we first detail the architecture of f_θ in Sec. 3.2. We then detail the function g by introducing our segmentation method in Sec. 3.3 and how pseudo-labels are generated from all three sensor modalities in Sec. 3.4. The training procedure is described in Sec. 3.5.

3.2 Unsupervised Multi-Modal Occupancy-Flow Model

Our model follows an encoder-decoder structure and the overall architecture is shown in Fig. 2. First, we encode the measurements of each sensor with two encoders, one for LiDAR-RADAR (f_lr) and one for camera (f_c):

$$\mathbf{Z}_\mathrm{c} = f_\mathrm{c}(\mathbf{C}_\mathrm{H}), \quad \mathbf{Z}_\mathrm{lr} = f_\mathrm{lr}(\mathbf{L}_\mathrm{H}, \mathbf{R}_\mathrm{H}). \tag{2}$$

The camera encoder f_c is a pretrained version of ResNet50 [28], which only takes in the most recent camera frame. The LiDAR-RADAR encoder discretizes locations at the voxel level of multiple historical frames, and encodes them as features in a 3D sparse feature map. The features from the LiDAR $\mathbf{l} = [x, y, z, i, \Delta t]^\intercal \in \mathbb{R}^5$ (including intensity i and relative timestep Δt) and RADAR points $\mathbf{r} = [x, y, z, v_r, \sigma, \Delta t]^\intercal \in \mathbb{R}^6$ (including range rate v_r, RADAR cross section σ, and Δt) are each encoded through separate shallow MLPs.

The latent representations $\mathbf{Z}_\mathrm{c}$ and $\mathbf{Z}_\mathrm{lr}$ are then fused by allowing the sparse LiDAR features at the coarsest level to attend to the image features through a voxel attention mechanism (f_voxelatt), similar to what is done in [52], giving $\mathcal{Z}_0 =$

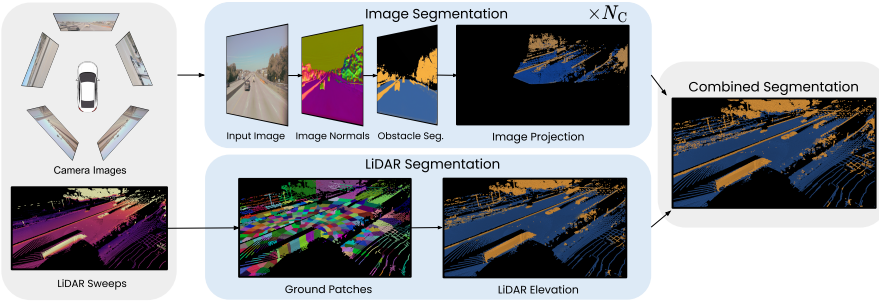

Fig. 3: **Obstacle** and **ground** pseudo-label generation process.

$[\mathbf{V}_{2x}, \mathbf{V}_{4x}, \mathbf{V}_{8x}, \mathbf{M}_{16x}]$, with feature volumes $\mathbf{V}_{2x}, \mathbf{V}_{4x}, \mathbf{V}_{8x}$ and $\mathbf{M}_{16x}$ having been downsampled by 2x, 4x, 8x, and 16x respectively. We then densify the coarsest feature map $\mathbf{M}_{16x}$ and pass it through several deformable attention layers (f_{defatt}) to increase the effective receptive field of the model. Finally, we use the same deformable attention multi-resolution header proposed by [18] for each of the model's predictions. We describe the specific details in the supplementary. The process is summarized as:

$$\mathcal{Z}_0 = f_{\text{voxelatt}}(\mathbf{Z}_{\text{c}}, \mathbf{Z}_{\text{lr}}) \rightarrow \mathcal{Z}_1 = f_{\text{defatt}}(\mathcal{Z}_0) \rightarrow (o, \mathbf{f}, s) = f_{\text{dec}}(\mathcal{Z}_1, \mathbf{Z}_{\text{lr}}, \mathbf{q}). \quad (3)$$

3.3 Obstacle Segmentation Model

To provide supervision for obstacle segmentation s, we propose a novel method for extracting obstacle/ground segmentation labels from image and LiDAR features in a probabilistic manner, combining the strength of each sensor modality. This approach ensures that the objects which are distinctly above ground are captured by the LiDAR segmentation, which is robust due to the 3D information it receives, while also detecting smaller close-to-the-ground obstacles that may only be observable in image space. The overall process is illustrated in Fig. 3.

Image Obstacle Segmentation. We propose using an image foundation model to produce estimates of normals and depths of the image for each camera. Specifically, given a camera image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ we apply Metric3Dv2 [32] to obtain per-pixel metric depth $d \in \mathbb{R}^{H \times W}$, depth confidence $c \in \mathbb{R}^{H \times W}$, and surface normals $\hat{\mathbf{n}} \in \mathbb{R}^{H \times W \times 3}$ with associated confidence $c_n \in \mathbb{R}^{H \times W}$. We then follow a similar process to [63] to produce a segmentation of the obstacles relative to the rest of the scene based on the normals and depth information. In contrast to prior work, which produce discrete labels, we output continuous probabilities for obstacles/ground ($p^{\text{image}}_{\text{obst}}$), in addition to probabilities of the validity of the estimation ($p^{\text{image}}_{\text{valid}}$). Producing these continuous image segmentation labels (ISL) enables the downstream model to learn a more calibrated representation.

LiDAR Obstacle Segmentation. We leverage ideas from Patchwork++ [47] for LiDAR segmentation to discretize the point cloud into polar patches, where

for each patch j, a ground plane is characterized by its centroid $\mathbf{c}_j \in \mathbb{R}^3$ and surface normal $\hat{\mathbf{n}}_j \in \mathbb{R}^3$. For a LiDAR return $\mathbf{l} \in \mathbb{R}^3$, we identify its corresponding patch in the xy-plane. We then get the elevation $e \in \mathbb{R}$ as the point to plane distance using the patch normal, and map this into continuous probabilities for the obstacle $p^{\text{LiDAR}}_{\text{obst}}$ and the valid distribution $p^{\text{LiDAR}}_{\text{valid}}$.

Combined Segmentation. To leverage the complementary strengths of both modalities, we employ a confidence-driven fusion using the per-point validity and obstacle probabilities p_{valid}, p_{obst} from both image (projected onto LiDAR) and LiDAR sources. Formally, the combined probability is

$$\mathbf{p}^{\text{comb}} = \begin{cases} \mathbf{p}^{\text{image}}, & \text{if } (p^{\text{LiDAR}}_{\text{valid}} < 0.5) \ \wedge \ (p^{\text{image}}_{\text{valid}} > p^{\text{LiDAR}}_{\text{valid}}) \\ \mathbf{p}^{\text{LiDAR}}, & \text{otherwise} \end{cases} \tag{4}$$

with $\mathbf{p}^{\text{SOURCE}} = \left[p^{\text{SOURCE}}_{\text{valid}}, p^{\text{SOURCE}}_{\text{obst}}\right]^\intercal$, $\text{SOURCE} \in \{\text{comb}, \text{image}, \text{LiDAR}\}$. The probability vector $\mathbf{p}^{\text{comb}}$ describes the combined soft labels (CSL) to supervise s. By prioritizing image labels for low-confidence LiDAR segmentation results, this approach captures small visual details while maintaining LiDAR's robustness to points that are clearly distinct from the ground plane. Additional details of our segmentation method are in the supplementary.

3.4 Multi-Modal Self-Supervision

We will now outline the pseudo-labels used to supervise TrIO. These can all be represented by a 4D query point $\mathbf{q}$, and a label $\bar{y}$, summarized as $\bar{\mathbf{y}} = (\mathbf{q}, \bar{y})$.

Occupancy. We generate dense occupancy pseudo-labels without any human annotation, following the methodology of UnO [1], DIO [18], and GASP [60]. For a LiDAR return $\mathbf{l}$ we assume the continuous ray segment between the sensor position and $\mathbf{l}$ as the unoccupied region $\mathcal{R}^-$, and a short segment immediately behind $\mathbf{l}$ along the ray direction as the occupied region $\mathcal{R}^+$. The sets of occupied and unoccupied points in the scene are $\mathcal{O}^+ = \bigcup_{\mathbf{l} \in \mathbf{L}} \mathcal{R}^+$ and $\mathcal{O}^- = \bigcup_{\mathbf{l} \in \mathbf{L}} \mathcal{R}^-$, respectively. The set of $2N$ sampled points with occupancy pseudo-labels is

$$\mathcal{Y}_{\text{occ}} = \{(\mathbf{q}, 1) \mid \mathbf{q} \in \mathcal{O}^+\} \cup \{(\mathbf{q}, 0) \mid \mathbf{q} \in \mathcal{O}^-\}. \tag{5}$$

Obstacle Segmentation. In order to train for the segmentation class, we revisit the pseudo-labels generated in Sec. 3.3. Recall that this formulation provides a combined probability vector $\mathbf{p}^{\text{comb}}$ for each point $\mathbf{r} \in \mathcal{P}_{\text{sweep}}$ in LiDAR sweep $\mathcal{P}_{\text{sweep}}$, containing both the validity and obstacle probabilities. We therefore use the locations of these LiDAR points with their time of emission (x, y, z, t) as coordinates for our query $\mathbf{q}$, and their corresponding combined probabilities as the soft labels. The set of points with pseudo-labels $\mathcal{Y}_{\text{seg}}$ is

$$\mathcal{Y}_{\text{seg}} = \{(\mathbf{q}, \mathbf{p}^{\text{comb}}(\mathbf{q})) \mid \mathbf{q} \in \mathcal{P}_{\text{sweep}}\}. \tag{6}$$

Instantaneous Flow. In order to train for unsupervised instantaneous flow, we use the Doppler readings from the RADAR sensor as our supervision source.

As RADAR points can be noisy, we first define a refined set of RADAR points $\mathcal{R}_{\text{clean}}$ where the measurement uncertainty is below a certain threshold. Due to the inherent characteristic that RADAR only captures velocity in the radial direction, our pseudo-labels must include the sensor velocity at the time of measurement $\mathbf{v}_{\text{sensor}}$, the unit radial direction of the RADAR point $\mathbf{d}_{\text{radar}}$, and the scalar Doppler velocity v_{doppler}. We define the flow pseudo-label set $\mathcal{Y}_{\text{flow}}$ as:

$$\mathcal{Y}_{\text{flow}} = \{(\mathbf{q}, (v_{\text{doppler}}, \mathbf{v}_{\text{sensor}}, \mathbf{d}_{\text{radar}})) \mid \mathbf{q} \in \mathcal{R}_{\text{clean}}\}. \tag{7}$$

3.5 Training

We train our model with a multi-task loss for occupancy ($\mathcal{L}_{\text{occ}}$), flow ($\mathcal{L}_{\text{flow}}$), and obstacle probability ($\mathcal{L}_{\text{seg}}$)

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{occ}} + \lambda_2 \mathcal{L}_{\text{flow}} + \lambda_3 \mathcal{L}_{\text{seg}} \tag{8}$$

using the pseudo-labels defined by $\mathcal{Y}_{\text{occ}}$, $\mathcal{Y}_{\text{seg}}$, and $\mathcal{Y}_{\text{flow}}$. Let TrIO's occupancy, obstacle probability, and flow output at a query point $\mathbf{q}$ be denoted as $f_{\theta,\text{occ}}(\mathbf{q}) \in [0,1]$, $f_{\theta,\text{seg}}(\mathbf{q}) \in [0,1]$, and $f_{\theta,\text{flow}}(\mathbf{q}) \in \mathbb{R}^3$ respectively. For notational simplicity, we drop the conditioning on the input $\mathbf{X}_{\text{H}}$. The occupancy loss is the binary cross-entropy loss over all query points:

$$\mathcal{L}_{\text{occ}} = \frac{1}{|\mathcal{Y}_{\text{occ}}|} \sum_{(\mathbf{q}, \bar{y}_{\text{occ}}) \in \mathcal{Y}_{\text{occ}}} \text{BCE}\left(f_{\theta,\text{occ}}(\mathbf{q}),\ \bar{y}_{\text{occ}}\right). \tag{9}$$

The obstacle probability loss is a soft focal loss (SFocalLoss):

$$\mathcal{L}_{\text{seg}} = \frac{1}{|\mathcal{Y}_{\text{seg}}|} \sum_{(\mathbf{q}, p^{\text{comb}}(\mathbf{q})) \in \mathcal{Y}_{\text{seg}}} \text{SFocalLoss}\left(f_{\theta,\text{seg}}(\mathbf{q}),\ p^{\text{comb}}_{\text{obst}}(\mathbf{q}),\ p^{\text{comb}}_{\text{valid}}(\mathbf{q})\right). \tag{10}$$

In contrast to $\mathcal{L}_{\text{occ}}$, which uses hard labels $\bar{y}_{\text{occ}} \in \{0,1\}$, $\mathcal{L}_{\text{seg}}$ operates on soft targets $p^{\text{comb}}_{\text{obst}} \in [0,1]$, with each point's contribution scaled by its label confidence $p^{\text{comb}}_{\text{valid}}$. We find this soft parameterization to be important for generating a well-calibrated model and elaborate further in Sec. 4. Full loss definitions are provided in the supplementary material. Finally, the flow supervision is determined as follows. We begin by moving the model's instantaneous 3D flow prediction $f_{\theta,\text{flow}}(\mathbf{q})$ into the sensor's frame of reference by subtracting the sensor's velocity $\mathbf{v}_{\text{sensor}}$, yielding the relative flow $\mathbf{f}_{\text{rel}}(\mathbf{q}) = f_{\theta,\text{flow}}(\mathbf{q}) - \mathbf{v}_{\text{sensor}}$. We then project this relative flow onto the radial direction vector $\mathbf{d}_{\text{radar}}$ to obtain the predicted radial velocity: $f_{\text{rad}}(\mathbf{q}) = \mathbf{d}_{\text{radar}} \cdot \mathbf{f}_{\text{rel}}(\mathbf{q})$. Finally, we supervise this value using the Doppler measurements with an L_1 loss over the cleaned RADAR points $\mathcal{R}_{\text{clean}}$:

$$\mathcal{L}_{\text{flow}} = \frac{1}{|\mathcal{R}_{\text{clean}}|} \sum_{\mathbf{q} \in \mathcal{R}_{\text{clean}}} |f_{\text{rad}}(\mathbf{q}) - v_{\text{doppler}}|. \tag{11}$$

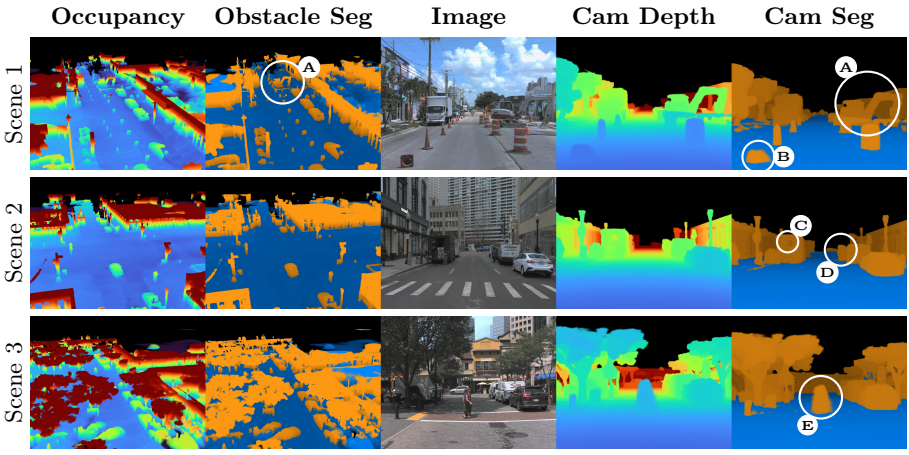

Fig. 4: Visualization of TriO on AV2. Each row shows (left to right): perspective-view occupancy, perspective-view obstacle segmentation, camera image, rendered depth, and rendered obstacle segmentation. TriO captures key elements of the scenes: **A)** excavator, **B)** fallen cone, **C)** inside of truck, **D)** open car door, **E)** pedestrians.

4 Experiments

In this section, we investigate the following research questions while comparing against the state-of-the-art: **Q1**: Can TriO reliably perceive unknown objects in a zero-shot, open-set manner? **Q2**: Does TriO's architecture lead to a better 4D geometric representation? **Q3**: Does unsupervised multi-modal pre-training improve downstream task performance via fine-tuning?

Datasets. We use the *Argoverse 2* (AV2) [94] *Sensor* split (700 train/ 150 val sequences) and evaluate on the AV2 Occupancy Forecasting leaderboard. The dataset *Spotting the Unexpected* (STU) [67] uniquely provides 3D LiDAR point labels for rarely seen objects on the road, aiming to test models on anything perception. We utilize all 22 available sequences with labels from STU (train and val) for evaluation. Note, STU is only for *zero-shot evaluation* – i.e. we train TriO and baselines on AV2 and apply it without any fine-tuning on STU, containing different sensor characteristics. Given the limited label range and no RADAR in previous datasets, we use an unreleased dataset *Long Range RADAR* (LRR) where we label bounding boxes up to 350m in front of the SDV. LRR has 2280 sequences (highway and urban). See supplementary for details.

4.1 Zero-Shot Anything Perception (Q1)

We evaluate our method's ability to perceive unusual, path-blocking objects on STU. First, we assess pseudo-label quality by comparing our ISL and CSL against STU ground truth. Using the correspondence between image and LiDAR labels in STU, we benchmark our soft pseudo-labels against open-set 2D and 3D

Table 1: Pseudo-label model segmentation performance on STU. MCPP refers to MaskCLIP++.

Method	mIoU ↑	F1 ↑	Recall ↑
MCPP + SAM	22.2	38.5	51.8
MCPP + SAM (Oracle)	22.2	38.3	51.9
MCPP + FC-CLIP	24.8	30.3	43.9
MCPP + FC-CLIP (Oracle)	27.4	35.1	52.4
MCPP + MAFT+	25.6	34.0	51.6
MCPP + MAFT+ (Oracle)	28.0	37.8	59.5
GroundedSAM	22.0	27.9	35.3
GroundedSAM (Oracle)	26.8	18.7	55.2
OWLv2 + SAM	20.2	21.9	23.4
OWLv2 + SAM (Oracle)	28.6	29.5	57.2
Patchwork++	42.1	51.4	62.8
ISL (Ours)	38.6	35.0	76.3
CSL (Ours)	**44.6**	**57.8**	**79.9**

Table 2: Semantic occupancy prediction performance of distilled methods. † is QueryOcc+ with LiDAR input.

Method	mIoU ↑	AP ↑	F1 ↑	Rec. ↑
UnO (ISL)	40.1	7.4	11.6	23.5
DiO (ISL)	49.7	29.3	34.3	50.2
UnO (CSL)	40.6	23.1	29.8	25.5
DiO (CSL)	49.8	74.8	72.9	65.6
TriO (MCPP + SAM)	27.9	25.7	27.5	24.0
TriO (MCPP + FC-CLIP)	23.1	2.2	5.2	56.7
TriO (MCPP + MAFT+)	20.6	10.2	14.8	10.0
TriO (GroundedSAM)†	50.4	50.1	52.1	54.8
TriO (OWLv2 + SAM)	50.1	72.3	70.1	68.8
TriO (Patchwork++)	39.4	74.8	70.8	59.5
TriO (ISL) (Ours)	41.8	57.1	53.7	59.9
TriO (CSL Hard) (Ours)	50.3	76.1	71.7	64.5
TriO (CSL Soft) (Ours)	**54.9**	**81.6**	**77.7**	**71.3**

baselines on equal footing. We also provide qualitatives results on AV2 in Fig. 4.

Anything Perception Baselines. We benchmark against state-of-the-art open-vocabulary methods, including *detect-then-segment* (GroundedSAM [76], OWLv2 [64]+SAM [45]) and *segment-then-classify* (MaskCLIP++ [99] with FC-CLIP [97], MAFT+ [38], or SAM) approaches. Following [69], we use standard prompts for ground ($\mathcal{P}_{\text{ground}}$) and non-ground ($\mathcal{P}_{\text{obstacle}}$) classes; see supplementary material for details. To establish an upper performance bound, we include an (Oracle) variant where scenes are manually inspected and specific unknown object classes are appended to $\mathcal{P}_{\text{obstacle}}$. Finally, we also include Patchwork++ [47] as a LiDAR-based baseline. For the second set of experiments, we train TriO's segmentation objective ($\mathcal{L}_{\text{seg}}$) using each of the pseudo-label from the baselines. This allows us to measure how effectively open-world knowledge can be distilled into the downstream world model, and in turn how well this model generalizes when evaluated on a different domain. Notably, our model under GroundedSAM supervision serves as a LiDAR-enhanced version of QueryOcc+ [54]. Finally, we compare against LiDAR-only models (UnO, DiO) trained with ISL and CSL.

Anything Perception Metrics. We report *mIoU* (mean Intersection over Union) [84, 91] for semantic predictions and *F1* score to evaluate pseudo-label quality at a fixed operating point without unfairly advantaging soft probability methods. We emphasize *Recall* to capture safety-critical missed detections, while Average Precision (*AP*) evaluates continuous probabilities in distilled occupancy models. See supplementary for further details.

Comparison of Pseudo-Label Methods. Our generated pseudo-labels (CSL) achieve the best balance of detecting the obstacles (measured by recall) on the road without too many false positive detections (F1 and mIoU) as shown in Tab. 1. Notably, this strong performance is not solely reliant on multi-modal fusion; even our image-only pseudo-labels (ISL) surpass all evaluated baselines in terms of recall. We also find that text-based segmentation baselines, even when prompted with the exact text description of the objects in the road, struggle to

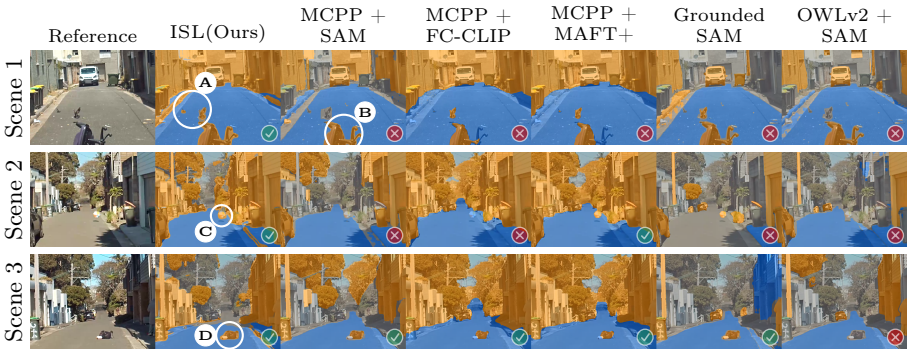

Fig. 5: Segmentation results on the STU validation set. Each scene contains an unexpected obstacle: **A)** two pigeons and **B)** chair, **C)** ball, and **D)** bookbag. An obstacle segmentation is correct if the road is labeled traversable (**blue**) and the obstacle is fully segmented as an obstacle (**orange**). ISL robustly segments long-tail obstacles.

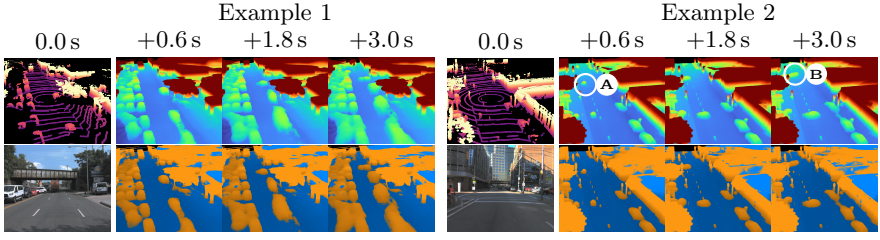

Fig. 6: Scene forecasting results of TrIO on AV2. LiDAR and camera sensor inputs are displayed at present time for reference. TrIO's forecasts maintain coherent obstacle segmentation throughout time, distinguishing occupied from free space even as agents move. TrIO captures a bicyclist **A)** and predicts its future multimodality **B)**.

generalize to these rarely seen classes, reinforcing the findings of [89]. We provide visual evidence for that claim in Fig. 5 and the supplementary. In contrast, our model, relying on geometry, is able to generalize in a more robust manner.

Semantic Occupancy Evaluation. As shown in Tab. 2 TrIO performs the best after the obstacle labels have been distilled into our occupancy model. We also find that using soft probabilistic labels (CSL Soft) tends to result in better performance than when trained with thresholded discretized labels (CSL Hard) due to better calibration (see AP, mIoU). TrIO improves also over LiDAR-only methods highlighting the benefit of using multi-modal sensor information.

4.2 Geometric Occupancy/ LiDAR Point Cloud Forecasting (Q2)

We evaluate TrIO on the AV2 Occupancy Forecasting benchmark; the goal of this benchmark is to determine how TrIO performs on occupancy prediction (visualized in Fig. 6), evaluated via LiDAR point cloud forecasting. Following

Table 3: LiDAR forecasting on the Argoverse 2 leaderboard.

	L1 (m) ↓	AbsRel (%) ↓	NFCD (m^2) ↓	CD (m^2) ↓
RayTracing	4.88	35.00	3.62	17.03
occformer_ep15	3.57	22.00	3.35	91.61
4D-Occ	3.22	19.00	2.45	72.74
occformer	2.97	17.00	2.03	71.33
Progressive ARM	2.32	13.00	1.81	71.41
UnO	2.24	12.00	0.86	**8.10**
DIO	2.11	11.00	**0.85**	13.96
TrIO	**1.82**	**10.63**	0.93	10.90

prior work [1,18,43], after training the world model f_θ, we freeze all parameters θ and introduce a LiDAR forecasting header f_ψ to predict LiDAR depth. We provide more details in the supplementary.

LiDAR Forecasting Baselines. We compare our method against the state-of-the-art from the Argoverse 2 leaderboard. Notably, 4D-Occ [43] serves as a representative unsupervised baseline for explicit occupancy prediction via voxel grid forecasting. In contrast, UnO [1] and DiO [18], also perform implicit occupancy prediction. However, unlike TrIO, these methods do not incorporate camera input, representing architectural differences in sensor modality.

LiDAR Forecasting Metrics: For point cloud forecasting we use well-established metrics of the AV2 Occupancy Forecasting leaderboard, namely Chamfer distance (CD), Near Field Chamfer Distance (*NFCD*), depth L1 error (*L1*), depth relative *L1* error (AbsRel) as our metrics.

Comparison against state-of-the-art. Tab. 3 compares TrIO against all the AV2 leaderboard baselines. TrIO is best in the L1 and AbsRel metrics, while competitive with baselines in other metrics. This underlines TrIO's ability to accurately forecast occupancy and its effectiveness as a world model.

4.3 Fine-tuning for Downstream Occupancy Tasks (Q3)

We show the versatility of the TrIO pre-trained feature extractor by evaluating its performance on various downstream tasks. We follow [1,60] and finetune the model to forecast semantic 3D (AV2) and semantic 2D BEV (LRR) occupancy and flow over time supervised with bounding box labels for the vehicle class.

Downstream Occupancy Baselines. For fine-tuning experiments, we pre-train UnO and DiO baselines via LiDAR self-supervision then fine-tune on 2D or 3D bounding box labels. We also compare against ImplicitO [2] which does not utilize unsupervised pre-training.

Downstream Occupancy Metrics. We use well-established metrics to evaluate the performance of the different methods on these downstream tasks by following prior work [1, 2, 18, 62]. We report mAP, SoftIoU [62] for occupancy and End Point Error (EPE) for flow. For LRR, we also evaluate each metric at over 200m in front of the SDV. We hypothesize that RADAR specifically having a longer effective range is able to improve our model performance at these dis-

Table 4: Fine-tuning downstream performance on AV2 and LRR.

	AV2			LRR					
				All			200m+		
Method	mAP ↑	Soft-IoU ↑	EPE ↓	mAP ↑	Soft-IoU ↑	EPE ↓	mAP ↑	Soft-IoU ↑	EPE ↓
Implicit-O	45.3	18.9	3.04	49.1	22.6	5.69	29.5	10.7	10.2
UnO	59.0	29.2	2.04	51.6	21.2	4.92	27.2	8.30	8.29
DIO	57.6	26.2	**1.43**	58.0	31.5	3.01	36.2	13.8	4.40
TrIO	**63.3**	**31.9**	1.97	**60.5**	**35.7**	**2.86**	**42.6**	**21.4**	**4.01**

tances, where other sensor readings may be too sparse to be consistently useful. This emphasizes the use case of LRR as only it is able to provide validation labels at a distance of 350m in front of the SDV.

Semantic Occupancy and Flow Forecasting. Tab. 4 and Tab. 5 show the BEV and 3D occupancy and flow forecasting results on AV2 and LRR. Our method outperforms all the baselines when finetuned to the supervised occupancy task. The drop in EPE performance in Tab. 4 compared to DIO is explained as TriO on AV2 has no flow supervision (due to the absence of RADAR data) during pre-training, whereas DIO does. Notably, our model outperforms the baselines significantly at range. The results show an improvement of 6.4 AP over the state-of-the-art showcasing the benefits of using RADAR at long-range.

Ablation Study. We evaluate the contributions of pre-training, sensor modalities, and loss objectives in Tab. 5. Unsupervised pre-training provides consistent gains across all metrics, validating the world model objectives. Omitting the radar loss leads to degradation in flow metrics across all ranges. Conversely, adding camera inputs improves far range occupancy performance at the expense of EPE; this suggests that single-frame camera features, which lack temporal context, may introduce noise into the flow estimation process. Removing RADAR as an input modality significantly degrades performance on long range metrics, supporting our hypothesis that this sensor modality directly lends itself to long range tasks.

Scaling Labeled Data. In Fig. 7 we scale the number of labeled training samples when fine-tuning pre-trained models (TrIO, UnO, DIO) and models trained from scratch (ImplicitO-4D). TrIO outperforms at all levels of supervision, and with 10^2 labeled training examples, reaches similar performance to the strongest baselines with 10^4 examples. TrIO is *two-orders of magnitude* more sample efficient, underscoring the generalizability of our method. Due to its novel obstacle segmentation supervision TrIO better understands obstacles and free space. This underlines the strong representation learned during pre-training and TrIO's ability to encode the geometry, semantics, and dynamics of the scene.

Limitations, Discussion, and Future Work. Currently, TrIO is limited to using RADAR for flow self-supervision, but future work could add occupancy supervision from RADAR. While our model uses RADAR to generalize occupancy

Table 5: Ablation studies of TriO on LRR.

Method	All			200m+		
	mAP ↑	Soft-IoU ↑	EPE ↓	mAP ↑	Soft-IoU ↑	EPE ↓
TriO (no pretraining)	59.7	33.9	4.09	41.0	19.8	4.70
TriO (no radar input)	57.7	31.9	3.04	34.5	14.3	4.59
TriO (no camera input)	**60.5**	**35.7**	**2.68**	42.2	20.9	**3.64**
TriO (no radar loss)	**60.5**	35.6	4.35	**42.6**	21.3	6.02
TriO	**60.5**	**35.7**	2.86	**42.6**	**21.4**	4.01

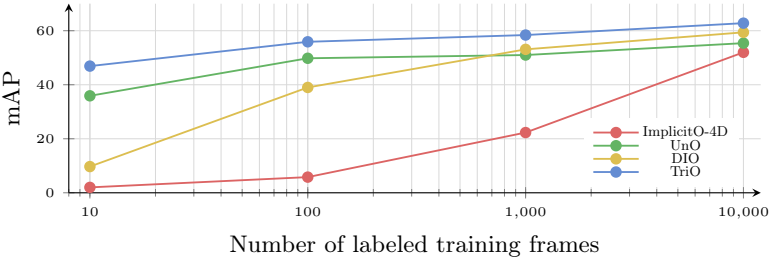

Fig. 7: Occupancy fine-tuning under a varying number of labelled frames.

predictions beyond the LiDAR range, RADAR can also be used as additional occupancy supervision in adverse weather to robustify predictions. This requires techniques beyond the line-of-sight assumption done in [46] due to nature of RADAR returns. Furthermore, the ability to perceive and forecast the movement of rare objects is an understudied problem in the SDV space. We see a general need for additional larger scale benchmarks focusing on the task of *Anything Perception* to ensure the overall robustness of many pre-existing systems.

5 Conclusion

We present TriO, a novel unsupervised tri-modal occupancy model that learns to predict 4D occupancy, obstacle semantics, and flow by leveraging the complementary physical synergies of camera, LiDAR, and RADAR data. It achieves state-of-the-art performance across multiple tasks, including zero-shot anything perception, LiDAR forecasting, and 3D/4D occupancy prediction, while being unsupervised and independent of additional costly human annotations. We observe failure-cases of open-vocabulary methods for rare classes, and show that TriO outperforms these methods in open-world scenarios. Notably, it achieves robust performance in perceiving anything relative to the drivable surface, including rare classes, showcasing its potential for a scalable deployment of SDVs.

References

1. Agro, B., Sykora, Q., Casas, S., Gilles, T., Urtasun, R.: Uno: Unsupervised occupancy fields for perception and forecasting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14487–14496 (June 2024)
2. Agro, B., Sykora, Q., Casas, S., Urtasun, R.: Implicit occupancy flow fields for perception and prediction in self-driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1379–1388 (2023)
3. Anonymous: Fantastic tractor-dogs and how not to find them with open-vocabulary detectors. In: The Fourteenth International Conference on Learning Representations (2026), https://openreview.net/forum?id=jUuXNrG7wh
4. Barsellotti, L., Bianchi, L., Messina, N., Carrara, F., Cornia, M., Baraldi, L., Falchi, F., Cucchiara, R.: Talking to dino: Bridging self-supervised vision backbones with language for open-vocabulary segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 22025–22035 (October 2025)
5. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9296–9306 (2019)
6. Biswas, S., Casas, S., Sykora, Q., Agro, B., Sadat, A., Urtasun, R.: Quad: Query-based interpretable neural motion planning for autonomous driving. In: ICRA (2022)
7. Boeder, S., Gigengack, F., Risse, B.: Gaussianflowocc: Sparse and weakly supervised occupancy estimation using gaussian splatting and temporal flow. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 24943–24954 (October 2025)
8. Boeder, S., Gigengack, F., Roesler, S., Caesar, H., Risse, B.: Shelfocc: Native 3d supervision beyond lidar for vision-based occupancy estimation (2025), https://arxiv.org/abs/2511.15396
9. Cao, A.Q., de Charette, R.: Monoscene: Monocular 3d semantic scene completion. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3981–3991 (2022). https://doi.org/10.1109/CVPR52688.2022.00396
10. Casas, S., Agro, B., Mao, J., Gilles, T., Cui, A., Li, T., Urtasun, R.: Detra: A unified model for object detection and trajectory forecasting. In: ECCV (2024)
11. Casas, S., Luo, W., Urtasun, R.: Intentnet: Learning to predict intention from raw sensor data. In: CoRL (2018)
12. Casas, S., Sadat, A., Urtasun, R.: Mp3: A unified model to map, perceive, predict and plan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14403–14412 (2021)
13. Cui, A., Casas, S., Sadat, A., Liao, R., Urtasun, R.: Lookout: Diverse multi-future prediction and planning for self-driving. In: ICCV (2021)
14. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2432–2443 (2017)
15. Diehl, C., Feicho, E., Schwambach, A., Dammeier, T., Mares, E., Bertram, T.: Radar-based dynamic occupancy grid mapping and object detection. In:

2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). pp. 1–6 (2020)
16. Diehl, C., Klosek, T., Krueger, M., Murzyn, N., Osterburg, T., Bertram, T.: Energy-based potential games for joint motion forecasting and control. In: Proceedings of The 7th Conference on Robot Learning. pp. 3112–3141 (2023)
17. Diehl, C., Sievernich, T.S., Krüger, M., Hoffmann, F., Bertram, T.: Uncertainty-aware model-based offline reinforcement learning for automated driving. IEEE Robotics and Automation Letters pp. 1167–1174 (2023)
18. Diehl, C., Sykora, Q., Agro, B., Gilles, T., Casas, S., Urtasun, R.: Dio: Decomposable implicit 4d occupancy-flow world model. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 27456–27466 (2025). https://doi.org/10.1109/CVPR52734.2025.02557
19. Ding, F., Pan, Z., Deng, Y., Deng, J., Lu, C.X.: Self-supervised scene flow estimation with 4-d automotive radar. IEEE Robotics and Automation Letters **7**(3), 8233–8240 (2022). https://doi.org/10.1109/LRA.2022.3187248
20. Ding, F., Wen, X., Zhu, Y., Li, Y., Lu, C.X.: Radarocc: Robust 3d occupancy prediction with 4d imaging radar. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 101589–101617. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-3222, https://proceedings.neurips.cc/paper_files/paper/2024/file/b81d83165e3145a2e7d33bb5e33ea913-Paper-Conference.pdf
21. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (Jun 1981). https://doi.org/10.1145/358669.358692, https://doi.org/10.1145/358669.358692
22. Gan, W., Liu, F., Xu, H., Mo, N., Yokoya, N.: Gaussianocc: Fully self-supervised and efficient 3d occupancy estimation with gaussian splatting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 28980–28990 (October 2025)
23. Gomes, T., Matias, D., Campos, A., Cunha, L., Roriz, R.: A survey on ground segmentation methods for automotive lidar sensors. Sensors **23**(2) (2023). https://doi.org/10.3390/s23020601
24. Graham, B., van der Maaten, L.: Submanifold sparse convolutional networks. arXiv preprint arXiv:1706.01307 (2017)
25. Gu, S., Yin, W., Jin, B., Guo, X., Wang, J., Li, H., Zhang, Q., Long, X.: Dome: Taming diffusion model into high-fidelity controllable occupancy world model (2024), https://arxiv.org/abs/2410.10429
26. Guo, E., An, P., Yang, Y., Liu, Q., Liu, A.A.: Fsf-net: Enhance 4d occupancy forecasting with coarse bev scene flow for autonomous driving (2024), https://arxiv.org/abs/2409.15841
27. Hao, Y., Madani, S., Guan, J., Alloulah, M., Gupta, S., Hassanieh, H.: Bootstrapping autonomous driving radars with self-supervised learning. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15012–15023 (2024). https://doi.org/10.1109/CVPR52733.2024.01422
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
29. Hess, G., Jaxing, J., Svensson, E., Hagerman, D., Petersson, C., Svensson, L.: Masked autoencoder for self-supervised pre-training on lidar point clouds. In:

Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 350–359 (2023)
30. Hoffmann, D.T., Raza, S.H., Jiang, H., Tananaev, D., Klingenhoefer, S., Meinke, M.: Floxels: Fast unsupervised voxel based scene flow estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22328–22337 (June 2025)
31. Hu, A., Murez, Z., Mohan, N., Dudas, S., Hawke, J., Badrinarayanan, V., Cipolla, R., Kendall, A.: Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In: ICCV (2021)
32. Hu, M., Yin, W., Zhang, C., Cai, Z., Long, X., Chen, H., Wang, K., Yu, G., Shen, C., Shen, S.: Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
33. Huang, Y., Zheng, W., Zhang, B., Zhou, J., Lu, J.: Selfocc: Self-supervised vision-based 3d occupancy prediction. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19946–19956 (2024). https://doi.org/10.1109/CVPR52733.2024.01885
34. Ivanovic, B., Leung, K., Schmerling, E., Pavone, M.: Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach. RA-L (2020)
35. Ivanovic, B., Pavone, M.: The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: ICCV (2019)
36. Jevtić, A., Reich, C., Wimbauer, F., Hahn, O., Rupprecht, C., Roth, S., Cremers, D.: Feed-forward scenedino for unsupervised semantic scene completion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6784–6796 (October 2025)
37. Jiang, H., Liu, L., Cheng, T., Wang, X., Lin, T., Su, Z., Liu, W., Wang, X.: Gausstr: Foundation model-aligned gaussian transformer for self-supervised 3d spatial understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11960–11970 (June 2025)
38. Jiao, S., Zhu, H., Huang, J., Zhao, Y., Wei, Y., Humphrey, S.: Collaborative vision-text representation optimizing for open-vocabulary segmentation. In: European Conference on Computer Vision (2024)
39. Jose, C., Moutakanni, T., Kang, D., Baldassarre, F., Darcet, T., Xu, H., Li, D., Szafraniec, M., Ramamonjisoa, M., Oquab, M., Siméoni, O., Vo, H.V., Labatut, P., Bojanowski, P.: Dinov2 meets text: A unified framework for image- and pixel-level vision-language alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 24905–24916 (June 2025)
40. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing. p. 61–70. Eurographics Association, Goslar, DEU (2006)
41. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., Luiten, J., Lopez-Antequera, M., Bulò, S.R., Richardt, C., Ramanan, D., Scherer, S., Kontschieder, P.: MapAnything: Universal feed-forward metric 3D reconstruction. In: International Conference on 3D Vision (3DV). IEEE (2026)
42. Khurana, T., Hu, P., Dave, A., Ziglar, J., Held, D., Ramanan, D.: Differentiable raycasting for self-supervised occupancy forecasting. In: European Conference on Computer Vision. pp. 353–369. Springer (2022)

43. Khurana, T., Hu, P., Held, D., Ramanan, D.: Point cloud forecasting as a proxy for 4d occupancy forecasting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1116–1124 (2023)
44. Kim, J., Woo, J., Shin, U., Oh, J., Im, S.: Flow4d: Leveraging 4d voxel network for lidar scene flow estimation. IEEE Robotics and Automation Letters pp. 1–8 (2025)
45. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4015–4026 (October 2023)
46. Kung, P.C., Harisha, S., Vasudevan, R., Eid, A., Skinner, K.A.: Radarsplat: Radar gaussian splatting for high-fidelity data synthesis and 3d reconstruction of autonomous driving scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 27596–27606 (October 2025)
47. Lee, S., Lim, H., Myung, H.: Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3d point cloud. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 13276–13283 (2022). https://doi.org/10.1109/IROS47612.2022.9981561
48. Li, P., Ding, S., Zhou, Y., Zhang, Q., Inak, O., Triess, L., Hanselmann, N., Cordts, M., Zell, A.: Ago: Adaptive grounding for open world 3d occupancy prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8645–8655 (October 2025)
49. Li, X., Cheng, X., Ju, X., Peng, Y., Hu, J., Li, J.: Optimization on the polarization and waveform of radar for better target detection performance under rainy condition. Remote Sensing **16**(14) (2024). https://doi.org/10.3390/rs16142557, https://www.mdpi.com/2072-4292/16/14/2557
50. Li, X., Kaesemodel Pontes, J., Lucey, S.: Neural scene flow prior. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 7838–7851. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper_files/paper/2021/file/41263b9a46f6f8f22668476661614478-Paper.pdf
51. Li, X., Zheng, J., Ferroni, F., Pontes, J.K., Lucey, S.: Fast neural scene flow. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9878–9890 (October 2023)
52. Li, Y., Yu, Z., Choy, C., Xiao, C., Alvarez, J.M., Fidler, S., Feng, C., Anandkumar, A.: Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9087–9098 (June 2023)
53. Liang, M., Yang, B., Zeng, W., Chen, Y., Hu, R., Casas, S., Urtasun, R.: Pnpnet: End-to-end perception and prediction with tracking in the loop. In: CVPR (2020)
54. Lilja, A., Lan, J., Fu, J., Hammarstrand, L.: Queryocc: Query-based self-supervision for 3d semantic occupancy (2025), https://arxiv.org/abs/2511.17221
55. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
56. Liu, D., Liu, D., Li, X., Lin, S., xie, H., Wang, B., Chang, X., Chu, L.: Self-supervised multi-frame neural scene flow (2024), https://arxiv.org/abs/2403.16116

57. Liu, R., Wu, X., Chen, X., Hu, L., Lou, Y.: 4d-rolls: 4d radar occupancy learning via lidar supervision. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 13602–13609 (2025). https://doi.org/10.1109/IROS60139.2025.11246471
58. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: European Conference on Computer Vision (ECCV) (2024)
59. Liu, Y., Mou, L., Yu, X., Han, C., Mao, S., Xiong, R., Wang, Y.: Let occ flow: Self-supervised 3d occupancy flow prediction. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Proceedings of The 8th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 270, pp. 2895–2912. PMLR (06–09 Nov 2025), https://proceedings.mlr.press/v270/liu25e.html
60. Ljungbergh, W., Lilja, A., Ling, A.T.A.L., Lindström, C., Verbeke, W., Fu, J., Petersson, C., Hammarstrand, L., Felsberg, M.: Gasp: Unifying geometric and semantic self-supervised pre-training for autonomous driving (2025), https://arxiv.org/abs/2503.15672
61. Ma, J., Chen, X., Huang, J., Xu, J., Luo, Z., Xu, J., Gu, W., Ai, R., Wang, H.: Cam4DOcc: Benchmark for Camera-Only 4D Occupancy Forecasting in Autonomous Driving Applications. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
62. Mahjourian, R., Kim, J., Chai, Y., Tan, M., Sapp, B., Anguelov, D.: Occupancy flow fields for motion forecasting in autonomous driving. IEEE Robotics and Automation Letters **7**(2), 5639–5646 (2022)
63. Man, Y., Weng, X., Kitani, K.M.: Groundnet: Segmentation-aware monocular ground plane estimation with geometric consistency. CoRR **abs/1811.07222** (2018), http://arxiv.org/abs/1811.07222
64. Minderer, M., Gritsenko, A., Houlsby, N.: Scaling open-vocabulary object detection. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 72983–73007. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/e6d58fc68c0f3c36ae6e0e64478a69c0-Paper-Conference.pdf
65. Ming, Z., Stephany Berrio, J., Shan, M., Worrall, S.: Occfusion: Multi-sensor fusion framework for 3d semantic occupancy prediction. IEEE Transactions on Intelligent Vehicles **10**(5), 3421–3433 (2025). https://doi.org/10.1109/TIV.2024.3453293
66. Nealen, A., Igarashi, T., Sorkine, O., Alexa, M.: Laplacian mesh optimization. In: Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia. p. 381–389 (2006)
67. Nekrasov, A., Burdorf, M., Worrall, S., Leibe, B., Berrio Perez, J.S.: Spotting the unexpected (stu): A 3d lidar dataset for anomaly segmentation in autonomous driving. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11875–11885 (2025). https://doi.org/10.1109/CVPR52734.2025.01109
68. Nuss, D., Reuter, S., Thom, M., Yuan, T., Krehl, G., Maile, M., Gern, A., Dietmayer, K.: A random finite set approach for dynamic occupancy grid maps with real-time application. The International Journal of Robotics Research **37**(8), 841–866 (2018). https://doi.org/10.1177/0278364918775523, https://doi.org/10.1177/0278364918775523

69. Osep, A., Meinhardt, T., Ferroni, F., Peri, N., Ramanan, D., Leal-TaixÃ©, L.: Better call sal: Towards learning to segment anything in lidar. In: European Conference on Computer Vision (ECCV) (2024)
70. Paigwar, A., Erkent, O., Sierra-Gonzalez, D., Laugier, C.: Gndnet: Fast ground plane estimation and point cloud segmentation for autonomous vehicles. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2150–2156 (2020). https://doi.org/10.1109/IROS45743.2020.9340979
71. Palladin, E., Brucker, S., Ghilotti, F., Narayanan, P., Bijelic, M., Heide, F.: Self-supervised sparse sensor fusion for long range perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 27498–27509 (October 2025)
72. Palmer, P., Krüger, M., Schütte, S., Altendorfer, R., Adam, G., Bertram, T.: Lerojd: Lidar extended radar-only object detection. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision – ECCV 2024. pp. 379–396. Springer Nature Switzerland, Cham (2025)
73. Pavković, T., Mahani, M.A.N., Niedermayer, J., Betz, J.: Gaussianfusionocc: A seamless sensor fusion approach for 3d occupancy prediction using 3d gaussians (2025), https://arxiv.org/abs/2507.18522
74. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: ECCV (2020)
75. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021), https://arxiv.org/abs/2103.00020
76. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., Zhang, L.: Grounded sam: Assembling open-world models for diverse visual tasks (2024), https://arxiv.org/abs/2401.14159
77. Richter, S., Bieder, F., Wirges, S., Stiller, C.: A dual evidential top-view representation to model the semantic environment of automated vehicles. IEEE Transactions on Intelligent Vehicles **9**(1), 2688–2700 (2024). https://doi.org/10.1109/TIV.2023.3284400
78. Rife, J.H., Li, Y.: Characterizing lidar range-measurement ambiguity due to multiple returns. In: Proceedings of the 38th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2025). pp. 1949–1963. Baltimore, Maryland (Sep 2025). https://doi.org/10.33012/2025.20444, https://doi.org/10.33012/2025.20444
79. Roldao, L., de Charette, R., Verroust-Blondet, A.: 3d semantic scene completion: a survey pp. 1978—-2005 (2022)
80. Sadat, A., Casas, S., Ren, M., Wu, X., Dhawan, P., Urtasun, R.: Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In: ECCV (2020)
81. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: ECCV (2020)
82. Steinke, N., Goehring, D., Rojas, R.: Groundgrid: Lidar point cloud ground segmentation and terrain estimation. IEEE Robotics and Automation Letters **9**(1), 420–426 (2024). https://doi.org/10.1109/LRA.2023.3333233
83. Tanzmeister, G., Thomas, J., Wollherr, D., Buss, M.: Grid-based mapping and tracking in dynamic environments using a uniform evidential environment repre-

sentation. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 6090–6095 (2014). https://doi.org/10.1109/ICRA.2014.6907756
84. Tian, X., Jiang, T., Yun, L., Mao, Y., Yang, H., Wang, Y., Wang, Y., Zhao, H.: Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 64318–64330 (2023)
85. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., Hénaff, O., Harmsen, J., Steiner, A., Zhai, X.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features (2025), https://arxiv.org/abs/2502.14786
86. Uttarkabat, S., Appukuttan, S., Gupta, K., Nayak, S., Palo, P.: Bloomnet: Perception of blooming effect in adas using synthetic lidar point cloud data. In: 2024 IEEE Intelligent Vehicles Symposium (IV). pp. 1886–1892 (2024). https://doi.org/10.1109/IV55156.2024.10588461
87. Vedder, K., Peri, N., Chodosh, N.E., Khatri, I., EATON, E., Jayaraman, D., Liu, Y., Ramanan, D., Hays, J.: Zeroflow: Scalable scene flow via distillation. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=FRCHDhbxZF
88. Vedder, K., Peri, N., Khatri, I., Li, S., Eaton, E., Kocamaz, M.K., Wang, Y., Yu, Z., Ramanan, D., Pehserl, J.: Neural eulerian scene flow fields. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=0CieWy9ONY
89. Šarić, J., Martinović, I., Kristan, M., Šegvić, S.: What holds back open-vocabulary segmentation? In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops. pp. 4256–4266 (October 2025)
90. Wang, L., Kim, S.W., Yang, J., Yu, C., Ivanovic, B., Waslander, S.L., Wang, Y., Fidler, S., Pavone, M., Karkus, P.: DistillneRF: Perceiving 3d scenes from single-glance images by distilling neural fields and foundation model features. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)
91. Wang, X., Zhu, Z., Xu, W., Zhang, Y., Wei, Y., Chi, X., Ye, Y., Du, D., Lu, J., Wang, X.: Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17850–17859 (October 2023)
92. Weng, X., Wang, J., Held, D., Kitani, K.: 3d multi-object tracking: A baseline and new evaluation metrics. In: IROS (2020)
93. Weng, X., Yuan, Y., Kitani, K.: Ptp: Parallelized tracking and prediction with graph neural networks and diversity sampling. RA-L (2021)
94. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., et al.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. arXiv preprint arXiv:2301.00493 (2023)
95. Yan, X., Gao, J., Li, J., Zhang, R., Li, Z., Huang, R., Cui, S.: Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 3101–3109 (2021)
96. Yang, H., Zhang, S., Huang, D., Wu, X., Zhu, H., He, T., Tang, S., Zhao, H., Qiu, Q., Lin, B., et al.: Unipad: A universal pre-training paradigm for autonomous driving. arXiv preprint arXiv:2310.08370 (2023)

97. Yu, Q., He, J., Deng, X., Shen, X., Chen, L.C.: Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 32215–32234. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/661caac7729aa7d8c6b8ac0d39ccbc6a-Paper-Conference.pdf
98. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 2018 International Conference on 3D Vision (3DV). pp. 728–737 (2018)
99. Zeng, Q.S., Li, Y., Zhou, D., Li, G., Hou, Q., Cheng, M.M.: High-quality mask tuning matters for open-vocabulary segmentation (2025), https://arxiv.org/abs/2412.11464
100. Zhang, C., Yan, J., Wei, Y., Li, J., Liu, L., Tang, Y., Duan, Y., Lu, J.: Occnerf: Advancing 3d occupancy prediction in lidar-free environments (2024), https://arxiv.org/abs/2312.09243
101. Zhang, Q., Yang, Y., Li, P., Andersson, O., Jensfelt, P.: Seflow: A self-supervised scene flow method in autonomous driving. In: European Conference on Computer Vision. Springer (2024)
102. Zhao, L., Luo, Y., Hay, J., Gan, L.: Shelfgaussian: Shelf-supervised open-vocabulary gaussian-based 3d scene understanding (2025), https://arxiv.org/abs/2512.03370
103. Zheng, W., Chen, W., Huang, Y., Zhang, B., Duan, Y., Lu, J.: Occworld: Learning a 3d occupancy world model for autonomous driving. In: ECCV (2024)
104. Zhu, H., Dong, Z., Topollai, K., Sha, B., Choromanska, A.: Self-supervised representation learning with joint embedding predictive architecture for automotive lidar object detection (2025), https://arxiv.org/abs/2501.04969
105. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results (2018), https://arxiv.org/abs/1811.11168
106. Zimmermann, K., Petrícek, T., Šalanský, V., Svoboda, T.: Learning for active 3d mapping. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 1548–1556 (2017)
107. Zuo, S., Zheng, W., Huang, Y., Zhou, J., Lu, J.: Gaussianworld: Gaussian world model for streaming 3d occupancy prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6772–6781 (June 2025)

Supplementary Materials

TriO: Tri-Modal Unsupervised Occupancy World Model for Anything Perception

This supplementary material provides additional related work, implementation details, the experimental setup, and further results.

Contents

1 Introduction 2
2 Related Work 4
3 Unsupervised Multi-Modal Occupancy-Flow Method 5
3.1 Unsupervised Task 5
3.2 Unsupervised Multi-Modal Occupancy-Flow Model 6
3.3 Obstacle Segmentation Model 7
3.4 Multi-Modal Self-Supervision 8
3.5 Training 9
4 Experiments 10
4.1 Zero-Shot Anything Perception (Q1) 10
4.2 Geometric Occupancy/ LiDAR Point Cloud Forecasting (Q2) 12
4.3 Fine-tuning for Downstream Occupancy Tasks (Q3) 13
5 Conclusion 15
A Extended Related Works 24
B Implementation Details 26
B.1 Network Architecture 26
B.2 Obstacle Segmentation Model 28
B.3 Occupancy World Model Loss Functions 30
B.4 Training of Downstream Task 31
B.5 Training Hyperparameters 32
B.6 Extended Discussion of Unsupervised Framing 32
C Experimental Setup 32
C.1 Occupancy World Model Pre-training (AV2/LRR) 32
C.2 Zero-Shot Anything Perception Evaluation (STU) 33
C.3 LiDAR Forecasting (AV2) 36
C.4 Downstream Occupancy Fine-tuning (AV2/LRR) 36
D Additional Experiments 37
D.1 Quantitative Results 37
D.2 Qualitative Results 39
E Extended Limitations and Future Work 40

A Extended Related Works

This section provides an additional comparison to related work. We begin by revisiting occupancy-related approaches described in the main paper. Tab. 6

Table 6: Comparison of TriO against the most closely related methods. * Only DIO-∅ does not use human labels. †QueryOcc+ uses both camera and LiDAR supervision. ‡The used segmentation models rely annotations like per pixel semantic masks or image-text pairs annotations. ◇ Continuous Gaussian coordinates discretized via voxelization

Method	Input			Supervision			Semantics	Continuous 4D	Unsupervised	
	Camera	LiDAR	RADAR	Camera	LiDAR	RADAR				
ImplicitO [2]	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗
Occ4D [43]	✗	✓	✗	✗	✓	✗	✗	✓	✓	✓
UnO [1]	✗	✓	✗	✗	✓	✗	✗	✓	✓	✓
DIO [18]	✗	✓	✗	✗	✓	✗	✓	✓	✓	(✗)*
GASP [60]	✗	✓	✗	✓	✓	✗	✗	✓	✓	✗
MonoScene [9]	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
GaussianWorld [107]	✓	✗	✗	✗	✗	✗	✓	(✓)◇	✓	✗
SceneDINO [36]	✓	✗	✗	✓	✗	✗	✓	✓	✗	✓
OccNerf [100]	✓	✗	✗	✓	✗	✗	✓	✓	✗	(✓)‡
GaussianFlowOcc [7]	✓	✗	✗	✓	✗	✗	✓	✗	✗	(✓)‡
QueryOcc [54]	✓	✗	✗	✓	(✓)†	✗	✓	✓	✗	(✓)‡
LRS4Fusion [71]	✓	✓	✗	✗	✓	✗	✗	✓	✓	✓
RadarOcc [20]	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
4DRolls [57]	✗	✗	✓	✗	✓	✗	✗	✗	✓	✓
RadarSplat [46]	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓
OccFusion [65]	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗
GaussianFusionOcc [73]	✓	✓	✓	✗	✗	✗	✓	(✓)◇	✗	✗
ShelfGaussian [102]	✓	✓	✓	✓	✓	✗	✓	(✓)◇	✗	(✓)‡
TriO (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

compares the most closely related methods in terms of input modalities, supervision signals, and model properties. TriO is the only method that uses all three modalities for both input and self-supervision, while remaining unsupervised and modeling continuous occupancy in 4D.

Self-Supervised Flow Predictions. TriO is also related to self-supervised scene flow methods, which can be grouped into two categories. The first solves global optimization problems using test-time optimization techniques to estimate point flow across multiple frames [51, 88]. One downside of this approach is its high runtime, which often makes these methods unsuitable for real-time applications. The second category uses feed-forward neural networks to predict point flow across frames. This is achieved through self-supervised losses [30, 44, 50, 56, 101] or distillation from test-time optimization techniques [87]. The previously mentioned methods primarily rely on multiple frames of LiDAR point cloud data. However, RADAR data can provide a partial supervision signal for velocity estimation via the Doppler effect. For example, [19] uses RADAR data, including radial velocity measurements from 4D RADARs. Other methods also combine occupancy and flow estimation: [18] proposes a occupancy consistency loss between 3D occupancy predictions to infer flow while relying solely

on LiDAR. [59] estimates 3D occupancy and occupancy flow using only camera inputs, with self-supervision via differentiable rendering.

In contrast, our method is not only able to predict 4D occupancy and obstacle segmentations, but also uniquely combines multiple sensor modalities as input and self-supervised flow learning with RADAR velocity, while additionally predicting future 4D flow.

Pre-Training for Perception. Representation learning methods aim to leverage large-scale unlabeled data to enhance performance on downstream tasks. Beyond the occupancy-based approaches [1, 18, 43, 54, 60, 71], which utilize LiDAR or camera self-supervision, several general-purpose representation learning frameworks have emerged. For instance, VoxelMAE [29] employs masked auto-encoding on voxelized point clouds as an effective pre-training strategy for 3D object detection. UniPAD [96] integrates both camera and LiDAR inputs within a 3D differentiable rendering framework to reconstruct color and depth. More recently, AD-L-JEPA [104] introduced a Joint Embedding Predictive Architecture (JEPA) for LiDAR, demonstrating superior downstream detection performance compared to traditional generative masked auto-encoders. In the RADAR domain, RaDICaL [27] proposed RADAR-to-RADAR and RADAR-to-VISION contrastive losses as a self-supervised pre-training objective. LEROjD [72] introduces a LIDAR-to-RADAR cross-modal distillation and LiDAR thin-out strategies for downstream RADAR-only object detection.

In contrast to these single- or dual-modality approaches, TrIO utilizes pre-training across all three modalities simultaneously. Furthermore, while existing methods focus primarily on representations for instantaneous downstream tasks (e.g. detection), TrIO specifically targets representations optimized for temporal 4D forecasting.

B Implementation Details

B.1 Network Architecture

In this subsection we revisit the architecture of TrIO and provide additional details on the implementation.

Encoder: We instantiate the camera encoder f_c as a ResNet50 [28] operating on N_C input images $\mathbf{C}$ of size 1920×1080. The LiDAR-RADAR branch voxelizes the multi-frame point measurements and encodes them into a sparse 3D feature grid. The voxel ROI spans $[-50, 350]$ m along the longitudinal axis, $[-50, 50]$ m along the lateral axis, both with a resolution of 0.15625m, and $[-2, 4]$ m in the vertical dimension, with a height resolution of 0.2m. Beyond spatial coordinates relative to the center of their corresponding voxel cell, $(x, y, z)_r \in \mathbb{R}^3$, the features from the LiDAR points $\mathbf{l} = [x, y, z, i, \Delta t]^\intercal \in \mathbb{R}^5$ (including intensity i and relative timestep Δt) and RADAR points $\mathbf{r} = [x, y, z, v_r, \sigma, \Delta t]^\intercal \in \mathbb{R}^6$ (including range rate v_r, RCS σ, and Δt) are passed through separate two-layer MLPs with

group norm to encode the input features of dimension 5 or 6 outputting features of size 15. The LiDAR intensity is rasterized in BEV and concatenated such that the final output has size 16. We aggregate points in the same voxel with mean aggregation. The result is a sparse feature grid $\mathbf{Z}_{\text{voxel}} \in \mathbb{R}^{30\times640\times2560}$. This tensor is passed through a series of sparse convolutions (LiDAR-RADAR backbone) [24] progressively halving the spatial resolution and doubling the channel dimension (16→32→64→128 channels), yielding $\mathbf{Z}_{\text{lr}} \in \mathbb{R}^{320\times80\times4\times128}$ at 8× spatial downsampling, which is ultimately fused with the image features $\mathbf{Z}_{\text{c}}$.

Feature Fusion. In order to fuse the features from the RADAR-LiDAR encoder and the image encoder $\mathbf{Z}_{\text{c}}, \mathbf{Z}_{\text{lr}}$, we employ an image to sparse voxel attention mechanism, inspired by [52] giving $\mathcal{Z}_0 = [\mathbf{V}_{2x}, \mathbf{V}_{4x}, \mathbf{V}_{8x}, \mathbf{M}_{16x}]$, with feature volumes $\mathbf{V}_{2x}, \mathbf{V}_{4x}, \mathbf{V}_{8x}$ and BEV feature map $\mathbf{M}_{16x}$ having resolution downsampled by 2x, 4x, 8x, and 16x, respectively:

$$\mathcal{Z}_0 = f_{\text{voxelatt}}(\mathbf{Z}_{\text{c}}, \mathbf{Z}_{\text{lr}}) \tag{12}$$

However, in our implementation, we allow each of the sparse features to attend using deformable attention back into the image features, using their projected location in image space as the center of the deformable attention mechanism. We also encode the position features in both the LiDAR and image space before running through the attention mechanism. $\mathbf{Z}_{\text{c}} \in \mathbb{R}^{N_{\text{cam}}\times128\times H_c\times W_c}$ denotes the camera feature maps from the ResNet50 image backbone, and $\mathbf{M}_{16x} \in \mathbb{R}^{256\times80\times320}$ is the resulting fused dense BEV feature map. We then get $\mathcal{Z}_0$ by using $\mathbf{M}_{16x}$ and the intermediate feature maps of the previously mentioned LiDAR-RADAR backbone.

Deformable Attention Neck. The fused BEV feature map $\mathcal{Z}_0$ is passed through a multi-scale neck $f_{\text{deformatt}}$ composed of a ResNet with deformable convolution blocks [105] followed by a feature pyramid network (FPN) [55], yielding:

$$\mathcal{Z}_1 = f_{\text{deformatt}}(\mathcal{Z}_0). \tag{13}$$

Each deformable convolution block learns offsets, enabling the network to adaptively expand its receptive field beyond the fixed local neighborhood of standard convolutions. The FPN aggregates these features producing a rich BEV representation at the original spatial resolution of $\mathbf{M}_{16x}$. This is reminiscent of what is done in DIO [18], but with the additional deformable convolution operations enabling larger effective receptive fields, resulting in better forecasting of fast-moving entities.

Decoder. For the decoder f_{dec}, we use the multiscale deformable attention mechanism of [18],

$$(o, \mathbf{f}, s) = f_{\text{dec}}(\mathcal{Z}_1, \mathbf{Z}_{\text{lr}}, \mathbf{q}). \tag{14}$$

Specifically, given a query point $\mathbf{q}$, the header attends to each level of the sparse feature maps $(\mathbf{V}_{2x}, \mathbf{V}_{4x}, \mathbf{V}_{8x})$ and the dense feature map in $\mathcal{Z}_1$ at the location

of this query point in 3D space. We then perform deformable attention to attend to a series of different locations, querying each of them across all feature maps, and combining them using the same deformable attention mechanism. The final features are then passed through a shallow MLP which decodes the occup

ancy probability o, the instantaneous motion (flow) $\mathbf{f}$ of the point, and the probability s of that point being classified as an obstacle. This operation can be done efficiently in parallel for a batch of query points.

B.2 Obstacle Segmentation Model

This section provides implementation details of the obstacle segmentation pipeline used to supervise s, including the per-modality pseudo-label construction and fusion rule.

Image Obstacle Segmentation. For each camera image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ with known intrinsics $K \in \mathbb{R}^{3\times 3}$ and extrinsic pose $T_{\text{cam}\to\text{veh}}$, we use Metric3D v2 [32] to obtain per-pixel metric depth $d \in \mathbb{R}^{H\times W}$, depth confidence $c \in \mathbb{R}^{H\times W}$, and surface normals $\hat{\mathbf{n}} \in \mathbb{R}^{H\times W\times 3}$ with associated confidence $c_n \in \mathbb{R}^{H\times W}$.

We then convert each pixel to a 3D point and transform it, together with the normals, into vehicle frame. Let $\mathbf{p}$ and $\mathbf{n}$ denote a resulting 3D point and surface normal vector, respectively. On a high-level we assume that obstacle regions are defined by points with upwards (in z) pointing normals and smooth normal gradients.

Soft Probabilistic Obstacle Criteria. Rather than applying hard thresholds we convert each geometric criterion into a soft probability using a parameterized sigmoid:

$$\sigma_{\tau,k}(x) = 1 - \frac{1 + e^{-k\tau}}{1 + e^{k(x-\tau)}}, \tag{15}$$

where τ is the transition point and k controls the steepness. This function transitions smoothly from 0 at x=0 to ≈1 for $x \gg \tau$. For criteria where high values indicate invalidity, we use the complement $\bar{\sigma}_{\tau,k}(x) = 1 - \sigma_{\tau,k}(x)$. We compute four independent per-pixel probabilities:

1. **Normal orientation**: $p_{\text{norm}} = \sigma_{\tau_n,k_n}(n_z)$, assigning high probability to upward-facing surfaces.
2. **Normal smoothness**: $p_{\text{grad}} = \bar{\sigma}_{\tau_g,k_g}\big(\|\nabla \mathbf{n}\|\big)$, where $\|\nabla \mathbf{n}\|$ is the normalized spatial image gradient of the normal field. We assume ground regions exhibit spatially smooth normals, while object boundaries produce large normalized gradients.
3. **Depth confidence**: $p_{\text{conf}} = \sigma_{\tau_c,k_c}(c)$, assigning low probability for points where depth estimates are unconfident.
4. **Range**: $p_{\text{dist}} = \bar{\sigma}_{\tau_d,k_d}(p_x)$, assigning low probability for distant points where monocular depth degrades.

RANSAC Plane Refinement. To further disambiguate ground from elevated flat surfaces (*e.g.*, vehicle rooftops), we fit a ground plane to candidate 3D points using RANSAC [21]. We select candidate ground points as those whose combined probability exceeds a threshold τ_π:

$$p_{\text{conf}} \cdot p_{\text{dist}} \cdot p_{\text{norm}} \cdot p_{\text{grad}} > \tau_\pi. \tag{16}$$

After applying RANSAC we compute the *point-to-plane distance* d_{plane}, and the soft probability:

$$p_{\text{plane}} = \bar{\sigma}_{\tau_r,k_r}\left(d_{\text{plane}}\right). \tag{17}$$

Final Obstacle and Validity Probabilities. We define the ground probability as the product of the normal-based criteria, and the obstacle probability as the inverse

$$p_{\text{ground}} = p_{\text{norm}} \cdot p_{\text{grad}} \tag{18}$$

$$p_{\text{obst}} = 1 - p_{\text{ground}} \tag{19}$$

and the validity probability, which captures whether a pixel carries reliable geometric information or belong to regions like the sky:

$$p_{\text{valid}} = \sigma(c_n) \cdot p_{\text{conf}} \cdot p_{\text{dist}} \cdot \big(1 - (1 - p_{\text{plane}}) \cdot p_{\text{ground}}\big), \tag{20}$$

where $\sigma(c_n)$ is the standard sigmoid of the normal confidence.

The resulting per-pixel distribution over three classes is:

$$p^{\text{image}}_{\text{invalid}} = 1 - p_{\text{valid}}, \tag{21}$$

$$p^{\text{image}}_{\text{obst}} = p_{\text{valid}} \cdot (1 - p_{\text{ground}}), \tag{22}$$

$$p^{\text{image}}_{\text{ground}} = p_{\text{valid}} \cdot p_{\text{ground}}. \tag{23}$$

These soft image-space labels (ISL) are projected onto 3D LiDAR points via camera–LiDAR calibration to produce dense supervisory signals for the model output s. The output is a set of LiDAR points with confidences and their probability of being associated with the ground class, defined by $p^{\text{image}}_{\text{valid}}$ and $p^{\text{image}}_{\text{ground}}$, respectively.

LiDAR Obstacle Segmentation. We take inspiration from Patchwork++ [47]; however, rather than using its final obstacle segmentation output, we utilize its core ideas for local ground patch estimation. We first discretize the point cloud into polar patches, where for each patch j, a ground plane is characterized by its centroid $\mathbf{c}_j \in \mathbb{R}^3$ and surface normal $\hat{\mathbf{n}}_j \in \mathbb{R}^3$. For a LiDAR return $\mathbf{l} \in \mathbb{R}^3$, we identify its corresponding patch in the xy-plane. We then get the elevation $e \in \mathbb{R}$ as the point to plane distance using the patch normal, which we finally convert into a probability using a parametrized sigmoid

$$p^{\text{LiDAR}}_{\text{obst}} = \sigma_{\tau_e,k_e}(e_i), \tag{24}$$

Table 7: Obstacle segmentation hyperparameters.

Symbol	Description	Value
τ_n	Normal orientation transition threshold	0.85
k_n	Normal orientation sigmoid steepness	20.0
τ_g	Normal smoothness transition threshold	0.05
k_g	Normal smoothness sigmoid steepness	70.0
τ_c	Depth confidence transition threshold	0.7
k_c	Depth confidence sigmoid steepness	100.0
τ_d	Distance transition threshold	150.0
k_d	Distance sigmoid steepness	0.05
τ_π	Candidate ground probability threshold	0.5
τ_r	Plane-distance transition threshold	0.5
k_r	Plane-distance sigmoid steepness	2.0
τ_e	LiDAR elevation transition threshold	0.2
k_e	LiDAR elevation sigmoid steepness	20.0
λ	Entropy scaling factor for $p_{\text{valid}}^{\text{LiDAR}}$	3
γ	Entropy offset for $p_{\text{valid}}^{\text{LiDAR}}$	1

To define the validity probability, we quantify the uncertainty of the obstacle classification using a scaled binary entropy $\mathcal{H}$. This ensures that points with ambiguous geometric features contribute less to the supervisory signal. The LiDAR validity is

$$p_{\text{valid}}^{\text{LiDAR}} = -\lambda \mathcal{H}(p_{\text{obst}}^{\text{LiDAR}}) + \gamma, \tag{25}$$

where λ and γ scale this to range between 0 and 1, ensuring that a probability of 0.5 results in a low validity score. This provides us with the LiDAR segmentation probabilities.

Fused Obstacle Segmentation. We use a confidence-based fusion rule to select the supervision source per LiDAR point from the image or LiDAR based segmentation. Specifically, if the LiDAR segmentation validity is low and the image validity is higher, we use image labels; otherwise, we keep LiDAR labels:

$$\mathbf{p}^{\text{comb}} = \begin{cases} \mathbf{p}^{\text{image}}, & \text{if } p_{\text{valid}}^{\text{LiDAR}} < 0.5 \ \wedge \ p_{\text{valid}}^{\text{image}} > p_{\text{valid}}^{\text{LiDAR}} \\ \mathbf{p}^{\text{LiDAR}}, & \text{otherwise} \end{cases} \tag{26}$$

with $\mathbf{p}^{\text{SOURCE}} = \left[p_{\text{valid}}^{\text{SOURCE}}, \ p_{\text{obst}}^{\text{SOURCE}}\right]^\intercal$, $\text{SOURCE} \in \{\text{comb}, \text{image}, \text{LiDAR}\}$. The probability vector $\mathbf{p}^{\text{comb}}$ defines the final combined soft labels (our combined labels), which we use for supervision of obstacle semantics. Table 7 summarizes the hyperparameters used in the obstacle segmentation pipeline.

B.3 Occupancy World Model Loss Functions

We provide additional loss function details for the occupancy ($\mathcal{L}_{\text{occ}}$) and obstacle segmentation ($\mathcal{L}_{\text{seg}}$) loss: The occupancy loss is a binary cross-entropy loss

summed across all query points:

$$\mathcal{L}_{\text{occ}} = -\frac{1}{|\mathcal{Q}^+ + \mathcal{Q}^-|}\left(\sum_{\mathbf{q}\in\mathcal{Q}^-}\log(1 - f_{\theta,\text{occ}}(\mathbf{q})) + \sum_{\mathbf{q}\in\mathcal{Q}^+}\log(f_{\theta,\text{occ}}(\mathbf{q}))\right). \quad (27)$$

The groundness probability loss is similarly defined as a *soft* focal loss function, weighted by the validity probability:

$$\begin{aligned}\mathcal{L}_{\text{seg}} = -\frac{1}{|\mathcal{P}_{\text{sweep}}|}\sum_{\mathbf{q}\in\mathcal{P}_{\text{sweep}}} p^{\text{comb}}_{\text{valid}}(\mathbf{q})\Big(&\alpha\, p^{\text{comb}}_{\text{ground}}(\mathbf{q})\big(1 - f_{\theta,\text{seg}}(\mathbf{q})\big)^{\gamma}\log\big(f_{\theta,\text{seg}}(\mathbf{q})\big) \\ &+ (1-\alpha)\big(1 - p^{\text{comb}}_{\text{ground}}(\mathbf{q})\big)\big(f_{\theta,\text{seg}}(\mathbf{q})\big)^{\gamma}\log\big(1 - f_{\theta,\text{seg}}(\mathbf{q})\big)\Big).\end{aligned} \quad (28)$$

with $\alpha = 0.25$ and $\gamma = 2.0$.

B.4 Training of Downstream Task

LiDAR Forecasting. For LiDAR forecasting, we apply techniques from prior work [1, 18, 42]. We first train the occupancy world model f_θ, followed by a training of a LiDAR forecasting header f_ψ with frozen occupancy parameters θ and new parameters ψ. This model is specifically trained to predict the expected depth of a LiDAR for the task of lidar forecasting. We use an L_1 loss to supervise the depth of the model under the assumption of previous works that the direction of the LiDAR ray can be assumed to be known, and that our goal is to render where the ray terminates. Therefore, given a set of LiDAR rays $\mathcal{P}_{\text{ray}}$ with LiDAR ray returns $\mathbf{l}$ we wish to sample, we formulate the loss as:

$$\mathcal{L}_{\text{lidar}} = \frac{1}{|\mathcal{P}_{\text{ray}}|}\sum_{\mathbf{l}\in\mathcal{P}_{\text{ray}}} \left|f_\psi(f_{\theta,\text{occ}}(\mathbf{l})) - \|\mathbf{l}\|_2\right|. \quad (29)$$

In practice, during training we randomly sample 50 LiDAR rays, then further sample 2000 points per ray at 0.1m resolution to calculate our LiDAR forecasting loss.

BEV/ 3D Supervised Semantic Occupancy. For supervised occupancy training, we simply use binary cross entropy to train the model using the rasterized box labels of the objects of interest. For the sake of evaluating how well the model forecasts the occupancy of fast moving objects, we focus on supervising exclusively on the vehicle class. Therefore, for a set of points $\mathcal{U}$ uniformly sampled in 2D BEV (LRR) or 3D (AV2) over the region of interest, and a label function $\text{Ras}(\mathbf{q})$ that returns 1 if the query point lies within a bounding box and 0 if not, with the loss function defined in Equation (27).

B.5 Training Hyperparameters

We train TrIO on 16 GPUs using the AdamW optimizer with a cosine learning rate schedule. The learning rate begins at 8.0×10^{-5} and follows a 1,000-iteration warmup period to reach a peak of 8.0×10^{-4}. We set $N = |\mathcal{Q}^+| = |\mathcal{Q}^-| = 7500$, $|\mathcal{P}_{\text{sweep}}| = 10000$, and $|\mathcal{R}_{\text{clean}}| = 2000$, with loss weights configured as $\lambda_1 = 1500, \lambda_2 = 10$, and $\lambda_3 = 125$. Note for a fair comparison, hyperparameters of baselines are also tuned.

B.6 Extended Discussion of Unsupervised Framing

Throughout this paper, we use *unsupervised* to denote that TrIO's training pipeline requires no human-provided annotations (e.g., bounding boxes, semantic labels, or text prompts). The pre-trained model used for pseudo-label generation (Metric3Dv2 [32]) was supervised on geometric properties (depth and normals), which can be derived from LiDAR and reconstructions. Nevertheless, some datasets (e.g., [14]) used by [32] (see Tab. 5 in the reference) contain further human-in-the-loop work like CAD model alignment; hence, we restrict our unsupervised claim to TrIO's *own training pipeline and not the entire system.*

C Experimental Setup

In this section we provide additional details for our experimental setup on Argoverse 2 (AV2) [94], Spotting the Unexpected (STU) [67] and the unreleased dataset Long Range Radar (LRR).

C.1 Occupancy World Model Pre-training (AV2/LRR)

AV2 is a dataset collected in six cities in the United States, especially mined for interesting urban scenarios. We use the official split with 750 train and 150 val sequences. Each sequence spans 15 s with 150 frames. We train TrIO on a ROI of $[-100, 150]$m in the x dimension $[-100, 100]$ in the y dimension and $[-4.5, 4.5]$m in z dimension. For generating the segmentation labels, we use all eight cameras of the camera rig ([94]). We train our model with the front left and front right cameras ($N_{\text{C}} = 2$) for memory efficiency and use one LiDAR ($N_{\text{L}} = 1$). We use $H = 3.0$ s of past data sampled at 5Hz and pre-train to predict $T = 3.0$ s into the future. We also ablated using four cameras (D.1), which led to slight performance gains.

LRR is a dataset with urban and highway driving with 2280 sequences each 20 seconds long, whereas we use 19 sequences for evaluation. TrIO uses $N_{\text{C}} = 5$ cameras, $N_{\text{L}} = 6$ LiDAR and $N_{\text{R}} = 4$ RADAR. We use $H = 0.5$ s of past data sampled at 10 Hz and pre-train with to predict $T = 1.5$ s into the future. We use an ROI of $[-50, 350]$ m on the x dimension $[-50, 50]$ on the y dimension and $[-2, 4]$ m in z dimension.

C.2 Zero-Shot Anything Perception Evaluation (STU)

We evaluate *zero-shot* obstacle segmentation on long-tail obstacles using the Spotting the Unexpected (STU) dataset, where we use one LiDAR ($N_{\mathrm{L}} = 1$) and one front camera ($N_{\mathrm{C}} = 1$) featuring sensor characteristics that differ from those on AV2. The STU dataset is split into three parts: a train set, a validation set, and a non-public test set. The train set consists of two sequences and represents naturalistic driving data with fully labeled scenes. The validation set contains 19 sequences and features a variety of unexpected obstacles placed in the road, but does not contain labels for the ground. We combine the train and validation sets into a single dataset for our evaluations in the following manner: first we retain only points with negative (ground) labels from the train set, second we retain only points with positive (obstacle) labels from the validation set. This combined set is only used for evaluation and not for training or fine-tuning the models trained on AV2. The combination allows for an evaluation focused on how well rare obstacles can be segmented from the road.

Next we detail pre-processing steps used in our evaluation process. First, labeled LiDAR points are filtered to the height range $z \in [-3, 5]$ m and to those projecting into the front camera's field of view via pinhole projection with the provided intrinsics, extrinsics, and distortion parameters. For the implicit world model evaluations (TriO), each LiDAR point $\mathbf{l} = (l_x, l_y, l_z)$ is augmented with a zero timestamp to form a query point $\mathbf{q} = (l_x, l_y, l_z, l_t{=}0)$. Then, the decoder is queried with these LiDAR points to obtain predictions for occupancy and groundness. Points with predicted occupancy of less than 0.5 are assigned to ground. For image-based baselines, obstacle segmentation prediction is obtained by projecting LiDAR points onto the segmentation image associated with the front camera. For image methods with binarized outputs, ignore/invalid classifications are mapped to ground. For image methods with probabilistic outputs, the valid probability channel is ignored when determining the pixel label (leaving only $p^{\mathrm{image}}_{\mathrm{obst}}$ and $p^{\mathrm{image}}_{\mathrm{ground}}$). Thus, we produce a set of predictions and their associated labels that can be used for evaluation of both implicit models and image space models.

Metrics. Given these predictions and their associated labels, a precision–recall curve is computed over the combined dataset. The operating-point threshold is selected to maximize the F_1 score. All metrics requiring an operating point are reported using this optimal threshold. Models that output already binarized have their binarized output used directly. We report four metrics: (i) mean Intersection over Union (mIoU) (ii) F_1 score; (iii) recall; and (iv) Average Precision (AP).

- **mIoU** [%]: This metric is the mean Intersection over Union, and represents the number of points that are correctly predicted to be the desired class, divided by the union of the prediction and the labels for said class. This is averaged over the two classes equally and is generally a standard metric for semantic occupancy [91].

Table 8: Default terms/ text prompts used for open-vocabulary label mapping. Those colored as orange correspond to an obstacle class, and those colored as blue correspond to a ground class. Table design inspired from [69].

Class	Text prompts
	Super classes
vehicle	vehicle, car, truck, bicycle, motorcycle, other-vehicle, jeep, SUV, van, bike, moped, pickup truck, caravan, trailer, bus, tram, train, construction vehicle, crane, excavator
human	human, person, bicyclist, motorcyclist, pedestrian, bicycle rider, motorcycle rider
ground	ground, road, sidewalk, parking, other-ground, driveable area, service lane, bike lane, parking lot, curb, driveway, traffic island
structure	structure, building, garage, wall, window, stair
nature	nature, vegetation, trunk, terrain, bush, shrub, foliage, treetop, tree trunk, grass, soil
object	object, fence, pole, traffic-sign, lamp post, traffic-sign pole, traffic-sign mounting, separator, small wall, crash barrier, traffic cone, bench

- **F1** [%]: This formally represents the harmonic mean over precision and recall. It represents how well the model can balance both of these metrics at a single optimal operating point. This metric is more suitable for evaluating the segmentation pseudo-labels than other metrics such as mean average precision, due to the fact that they evaluate at a single operating point, and therefore do not unfairly advantage soft metrics that return the probability of each class.
- **Recall** [%]: Recall tracks the proportion of ground-truth obstacle points that are correctly classified as obstacle points. This is important for evaluating the safety of autonomous driving systems, as missing even a single obstacle could lead to a critical failure of the overall system.
- **AP**[%]: Average Precision represents the area under the precision recall curve, and is often used to determine how well a model is able to interplay between true positives, false positives, and false negatives. However, the precision recall curve is computed over a series of different thresholds, and therefore this metric is designed with continuous probability models in mind. As a result, we only use this metric on our occupancy baselines, where a direct comparison is fair.

Baselines We benchmark against several state-of-the-art open-vocabulary segmentation methods: GroundedSAM [76] employs an open-set object detector, Grounding DINO [58], with various text prompts to query SAM [45]. Similarly, we introduce an OWLv2+SAM baseline that replaces the Grounding DINO detector with OWLv2 [64]. These methods follow a *detect-then-segment* paradigm.

Table 9: Oracle obstacle text prompts/ terms hand picked for STU, separated by category. These prompts are added to $\mathcal{P}_{\text{obstacle}}$ got the oracle baselines.

Category	Oracle term/ text prompt
Seating	Chair, Office chair, Roller chair, Gaming chair
Bags	Dior bag, Shopping bag, Paper bag, Plastic bag, Polkadot bag, Banana bag, Zebra bag, Inflatable bag, Briefcase, Sleeping bag, Workout bag, Backpack
Inflatables	Inflatable toy, Inflatable ball, Inflatable, Pool noodle
Balls	Ball, Beach ball
Mats & Foam	Yoga mat, Foam mat, Foam, Noodle
Animals & Toys	Kangaroo, Panda, Model animal, Road kill, RC car, Model car, Toy, Donuts, Banana
Umbrellas	Umbrella, Open umbrella
Vegetation	Tree branch, Branch, Foliage, Leaves
Misc. Debris	Cardboard, Debris, Toolbox

Conversely, we also evaluate *segment-then-classify* approaches, where the image is first partitioned into masks which are subsequently classified. We utilize the state-of-the-art MaskCLIP++ [99] framework, which we combine with various mask predictors. Specifically, we evaluate mask predictors from FC-CLIP [97] and MAFT+ [38] and SAM. We also compare against more heuristic-based ground segmentation models, such as Patchwork++ [47], which uses estimated LiDAR elevation to compare against what is and what isn't ground.

For the second set of experiments, we also train TriO's segmentation objective ($\mathcal{L}_{\text{seg}}$) using each of the pseudo-labels from the baselines. The goal is to determine how well this information can be distilled into a downstream world model, and in turn how well this model generalizes when evaluated on a different domain. Notably, our model under GroundedSAM supervision can be viewed as an extension of the recent work QueryOcc+ [54], albeit with a distinct architecture and additional LiDAR input. Finally, we also add baselines training the LiDAR-only based models DIO [18] and UnO [1] with the pseudo-labels from our method (ISL, CSL).

Text Prompts. For all open-vocabulary methods, we utilize a consistent set of prompts for ground ($\mathcal{P}_{\text{ground}}$) and obstacle ($\mathcal{P}_{\text{obst}}$) classes based on [69]. Simply put, we start with those prompts and then map them to either the obstacle or the ground class as shown in Table 8. Following common practices, the baselines also wrap each text prompt into a full sentence, resulting in prompts like *a photo of a pedestrian*. The MaskCLIP++ (MCPP) baselines first generate a set of mask, which are later classified. To avoid assigning regions like the sky to either ground or obstacle, we prompt those models additionally with the prompt *sky* and map this later to an ignore class, which enables a more fair comparison.

Finally, recognizing that a closed-set of predefined prompts are inherently limited and risks omitting rare object classes, to establish an upper bound for

open-vocabulary performance, we include an **(Oracle)** variant where scenes are manually inspected and specific unknown object classes are appended to $\mathcal{P}_{\text{obstacle}}$. We show these prompts in Table 9.

C.3 LiDAR Forecasting (AV2)

Following the Argoverse 2 leaderboard, we report the following four metrics using the same evaluation protocol:

- **L1** [m] The average absolute error between the depth predicted by the occupancy model and the ground truth LiDAR ray depth. This is the primary metric for the AV2 leaderboard rankings.
- **AbsRel** [%] The absolute error of the LiDAR ray normalized by the ground truth depth.
- **CD** [m^2]: The Chamfer Distance between the predicted LiDAR point cloud and the ground truth. It is defined as the mean squared distance from every point in the predicted point cloud to its nearest neighbor in the true point cloud.
- **NFCD** [m^2]: The Near Field Chamfer Distance, which measures the Chamfer distance between the predicted and ground truth point clouds within a restricted 70m$\times$ 70m range.

C.4 Downstream Occupancy Fine-tuning (AV2/LRR)

On AV2, the evaluation ROI is same as training except only 25m to the sides due to memory constraints and at a resolution of 0.5 m every 0.1 s. LRR uses the same evaluation ROI (x, y) as described in Section C.1. We evaluate on a grid of query points with resolution 0.3125 m every 0.3 s.

Metrics. The fine-tuning experiments for BEV/3D semantic occupancy/flow forecasting use the following metrics:

- **mAP** [%]: This mean average precision metric is the average area under the precision recall curve when the predicted occupancy is evaluated against the rasterized bounding box labels. This can be done in both 2D BEV and in 3D.
- **IoU** [%]: Soft Intersection Over Union is a metric originally proposed by [62] as a more calibrated soft evaluation method for occupancy. It is the probabilistic equivalent of IoU, previously introduced.
- **EPE** [%]: End point error represents the L_2 distance between the ground truth flow vector, and the predicted flow vector. This can be computed in 2D for 2D BEV models, and in 3D.

D Additional Experiments

D.1 Quantitative Results

Runtime and Memory. In Table 10, we provide the runtime and memory usage for TriO and the baselines on an NVIDIA RTX 5000 GPU. We also report the number of parameters for each method. We distinguish between two settings: The first setting (eval) is evaluated on a full grid of query points (x, y, t), totaling 2,048,000 points; this setting is also used to compute our metrics. We observe that UnO and DIO achieve lower runtime and memory usage, which can be attributed to the addition of the new camera and RADAR sensors in TriO. Specifically, the camera encoding contributes to an increased parameter count, motivating the need for more efficient architectures in future work. In the second setting, we query the model with 20,000 points and measure the resulting runtime and memory. This is motivated by the number of query points used by a state-of-the-art planner, as reported by QuAD [6]. We observe a significant reduction in runtime across all methods when transitioning from the grid (eval: 2,048k) to the planner (20k) query setting. Notably, TriO (no camera) achieves the best runtime, rendering the method real-time capable.

Table 10: Model parameters, runtime, and GPU memory for two query-point settings (Planner [6]: 20k points; Eval: 2048k points), and fine-tuning downstream performance on LRR.

	All			200m+					Planner [6]	Eval
Method	mAP ↑	Soft-IoU ↑	EPE ↓	mAP ↑	Soft-IoU ↑	EPE ↓	# Params	Mem (MB) ↓	Runtime (ms) ↓	Runtime (ms) ↓
UnO	51.6	21.2	4.92	27.2	8.30	8.29	39,742,221	**2,493**	63.92	1774.41
DIO	58.0	31.5	3.01	36.2	13.8	4.40	8,205,109	2,930	68.21	1776.86
TriO (no camera)	**60.5**	**35.7**	**2.68**	42.2	20.9	**3.64**	12,930,483	3,449	**60.81**	**1622.54**
TriO (no radar)	57.7	31.9	3.04	34.5	14.3	4.59	54,224,584	9,907	488.06	2048.88
TriO	**60.5**	**35.7**	2.86	**42.6**	**21.4**	4.01	54,224,967	9,915	489.32	2054.63

Additional Camera Ablations Tab. 11 shows marginal gains from 4 cameras on AV2 in occupancy, flow, and LiDAR metrics. However, these were not enough to justify the added cost during our main experiments, which aligns with our no-camera ablations in the main paper.

Expanding on our no camera ablation study in the main paper on LRR, we find camera inputs increases occupancy performance at range lending to the idea that camera features are more dense than other sensor modalities such as LiDAR at these distances. Howeve, since we only pass in one timestep of camera data as input to the model, we hypothesize this may be responsible for the slight regression in flow estimation since a single timestep of camera is insufficient to estimate flow by itself. The improvement in flow in AV2, contrasting with the decline seen in the LRR "no-camera" ablation, is likely because AV2 (urban) contains significantly more static vehicles. In contrast, LRR (highway/urban)

includes many high-velocity objects for which single-frame camera input is less informative. Furthermore, LiDAR provides a dominant signal for geometric labels (occupancy, obstacle segmentation) except at long range, where sparsity increases; this aligns with the observed occupancy improvements at range when incorporating camera data. Future work could explore camera pre-training to better leverage multi-modal features.

Table 11: Camera Ablation on AV2

	4D Semantic Occupancy			LiDAR Forecasting			
Model	mAP (%) ↑	SoftIoU (%) ↑	EPE (m) ↓	L1 (m) ↓	AbsRel (%) ↓	NFCD (m^2) ↓	CD (m^2) ↓
TRiO (2 Cam)	**63.3**	31.9	1.97	**1.82**	10.63	0.93	**10.90**
TRiO (4 Cam)	**63.3**	**32.3**	**1.91**	**1.82**	**10.48**	**0.90**	11.06

Unsupervised Flow Estimation In Tab. 12, we compare the unsupervised flow estimation of TriO against that of DIO [18], which uses a warping consistency loss. We observe that our method, directly supervised with RADAR, provides the best unsupervised flow estimation. We find that DIO only learns a minimal amount of flow on the surface of objects, leaving the occupancy in the interior stationary. This is also reflected in the EPE (static) metric, where DIO performs well due to predicting zero flow. As a result, our RADAR supervised flow results in significantly more accurate predictions, especially for dynamic obstacles as shown by the EPE (dynamic).

Table 12: Unsupervised flow forecasting results on LRR.

	All			200m+		
Method	EPE ↓	EPE (static) ↓	EPE (dynamic) ↓	EPE ↓	EPE (static) ↓	EPE (dynamic) ↓
DIO	27.8	**0.01**	28.8	28.4	**0.01**	29.3
TRiO	**19.4**	0.17	**20.3**	**19.6**	0.16	**20.5**

Breakdown of AV2 Prediction Metrics Tab. 13 shows a metrics breakdown in near and far field. We contacted the leaderboard creators and confirmed the published L1 and AbsRel metrics are implicitly near field. Tab. 13 shows our re-implemented UnO and DIO baselines, have slightly different metrics (better L1, worse CD) than the results from the leaderboard in the main paper, but in general we do better close by and worse at range.

Sensitivity Analysis Fig. 8 shows the current 0.5 threshold is a good balance between recall and F1 score. The method is robust balance across all metrics;

Table 13: AV2 LiDAR forecasting. Leaderboard metrics.

Model	NF-L1 ↓	FF-L1 ↓	NF-AbsRel (%) ↓	FF-AbsRel (%) ↓	NFCD ↓	FFCD ↓	CD ↓
UnO	2.07	**2.07**	11.69	**3.01**	1.03	79.28	11.41
DIO	1.87	2.20	11.28	3.28	0.97	87.07	12.43
TRiO	**1.82**	2.13	**10.63**	3.31	**0.93**	**76.52**	**10.90**

performance only degrades at extreme values. We also perform loss sensitivity analysis on LRR. Scaling each loss weight by 4× and 0.25× shows that TriO is robust to this, with min/max deviations of 0.1/0.2% AP, 0.0/0.4% SoftIoU, 0.03/0.44 m EPE.

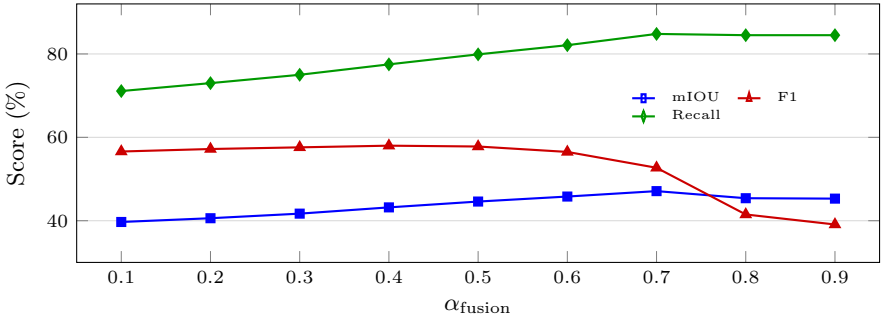

Fig. 8: Sensitivity to fusion parameter α_{fusion} (STU)

.

D.2 Qualitative Results

We provide additional qualitative visualizations to highlight different aspects of TriO and the baselines. Figures 10 and 11 show diverse AV2 scenes, illustrating the model's generalization across a wide range of scenarios and obstacle types, including VRUs, construction vehicles, animals, infrastructure, and challenging weather conditions like snow.

Figures 12 and 13 compare our pseudo-labels against several open-vocabulary VLM baselines on a broad range of uncommon obstacles, emphasizing both the strong generalization of our model and representative failure cases of current state-of-the-art VLMs, even when supplied with 'ground-truth' text prompts (Oracle).

Figures 14 and 15 present additional zero-shot results of our model, trained on AV2, via roll-outs on STU that showcase obstacle segmentation for rare objects. In addition Figure 16 shows an AV2 rollout containing a vulnerable road user on a scooter. Figure 17 visualizes forecasted LiDAR vs ground-truth LiDAR

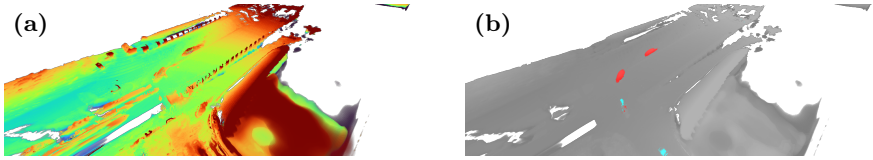

Fig. 9: a) Warped occupancy by flow (hard scenario). b) Flow degeneracy: longitudinal bias failing to resolve orthogonal motion.

on a challenging scenario with different traffic participants. Figures 18 and 19 summarize typical failure modes on AV2 and STU, respectively.

Meanwhile, Figure 20 showcase how our model successfully extends to edge cases such as speed bumps and hilly roads.

E Extended Limitations and Future Work

For obstacle segmentation pseudo-labels, we use a late-fusion approach that combines image- and LiDAR-based segmentation, where pseudo-labels are first generated independently for each modality. Since both segmentation approaches rely primarily on surface normals, future work could explore unified early-fusion methods such as [41]. Although including cameras as an input improves long-range occupancy performance, it also increases runtime and memory usage. Future work should therefore investigate more efficient camera encoding approaches.

While TrIO also generalizes to other weather conditions, for example as shown in Figure 10, we sometimes observe false-positive occupancy and obstacle segmentation predictions caused by snow and exhaust (Figure 18). Incorporating more occupancy self-supervision from cameras and RADAR could help mitigate these failures. We acknowledge that open-loop metrics may not fully capture final closed-loop driving performance. Although this is beyond the scope of the present paper, we are interested in studying how TrIO can be used directly in downstream motion planning tasks, since the ability to detect and avoid rarely seen objects is vital for self-driving.

Another area for future work is improving upon the flow representation. Forecasted occupancy warped via 3D flow (Fig. 9 a) reveals inconsistencies between both. While RADAR flow is a strong pretraining signal (Tab. 3) outperforming consistency-baselines (Tab. 6, Supp.), it is insufficient for standalone metric flow. TriO resolves 3D flow from 1D radial Doppler with spatially and temporally diverse measurements: Multiple RADAR sensors capture distinct radial components of the same velocity vector from different viewpoints, resolving the full 3D flow via geometric triangulation. A single object can also yield multiple returns across its spatial extent, and we supervise with future sensor frames, making learning only radial flow non-trivial. We hypothesize that the location invariance in the CNN backbone also helps. However, tangential failures (Fig. 9 b) persist, likely due to highway motion biases.

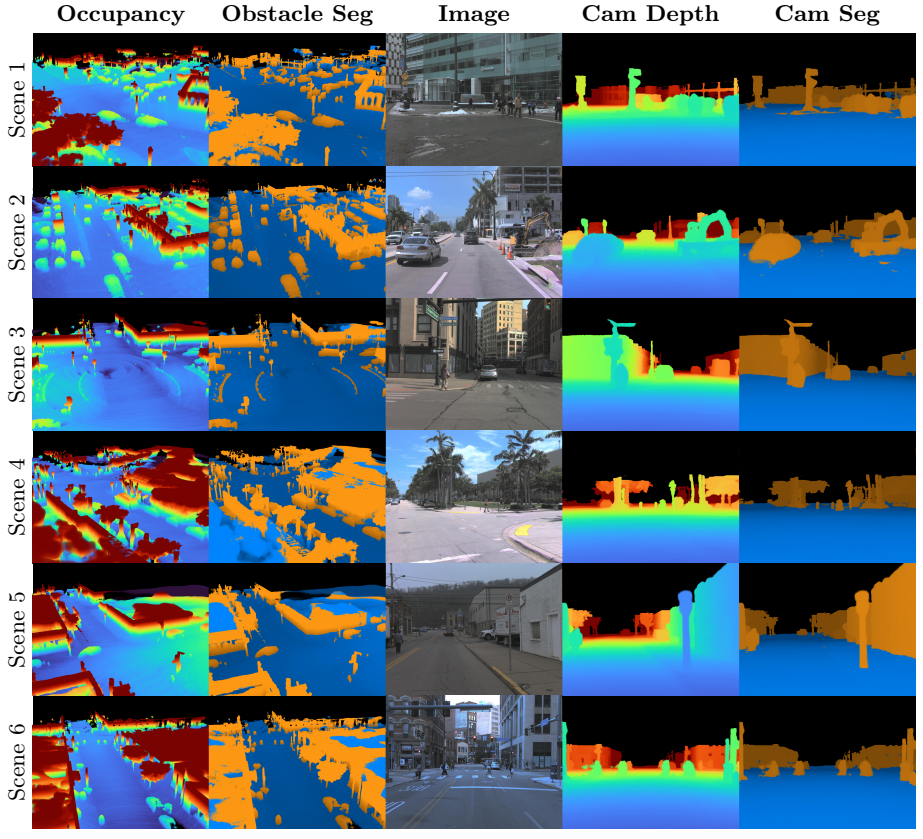

Fig. 10: TriO qualitative results on AV2 across a variety of diverse scenes: 1) large group of pedestrians in the snow, 2) excavator, 3) pedestrian at street corner and detailed structures, 4) pedestrian with trees, 5) different vehicle types and pedestrian walking on street, 6) pedestrians crossing street at intersection.

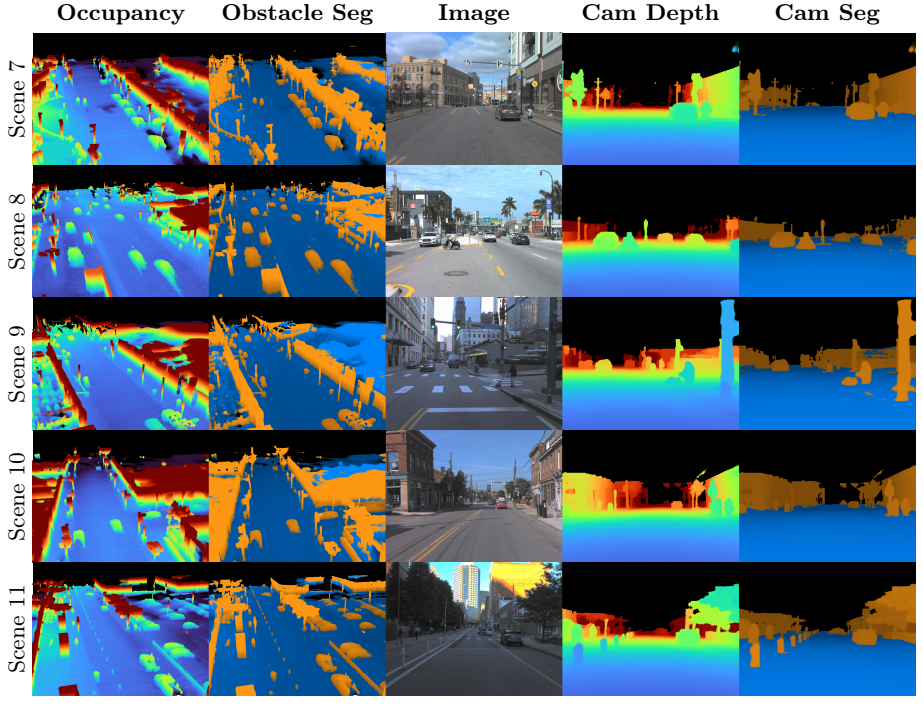

Fig. 11: Additional TrIO qualitative results on AV2: 7) closeby pedestrians and detailed static structures, 8) scooter in middle of road, 9) pedestrian pulling a box, 10) vehicles and pedestrians on sidewalk, 11) pylons, parked vehicles, and pedestrians on sidewalk.

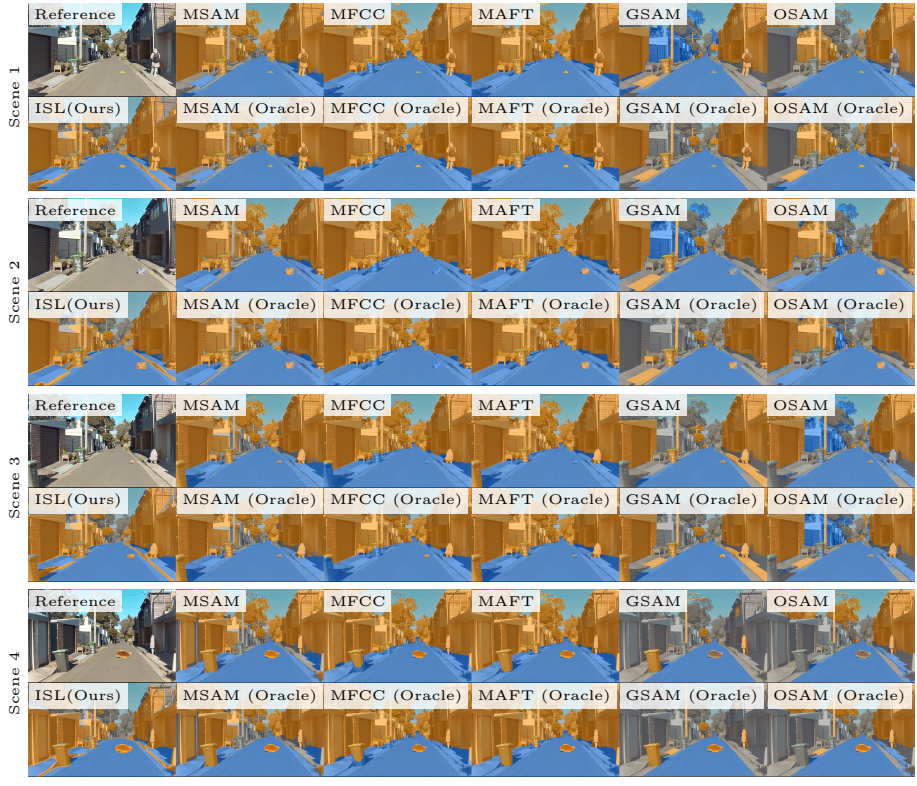

Fig. 12: Pseudo-labels of our method in comparison to baselines across a variety of scenarios. Oracle denotes upper-performance bound baselines with additional text prompts of Table 9. To fit all the labels into the image we use the following naming convention: MCPP + SAM → MSAM, MCPP + FC-CLIP → MFCC, MCPP + MAFT+ → MAFT, Grounded SAM → GSAM, OWLv2 + SAM → OSAM. In this figure, there are a number of scenes including: 1) a pedestrian with debris on the road, 2) a box on the road, 3) a pedestrian with trash on the road, 4) umbrella with object behind.

Fig. 13: Additional pseudo-labels of our method in comparison to baselines, following the same naming convention as in Figure 12: 5) fallen object and pedestrian crouching, 6) toy car in the road, 7) pile of garbage, 8) box with debris on it.

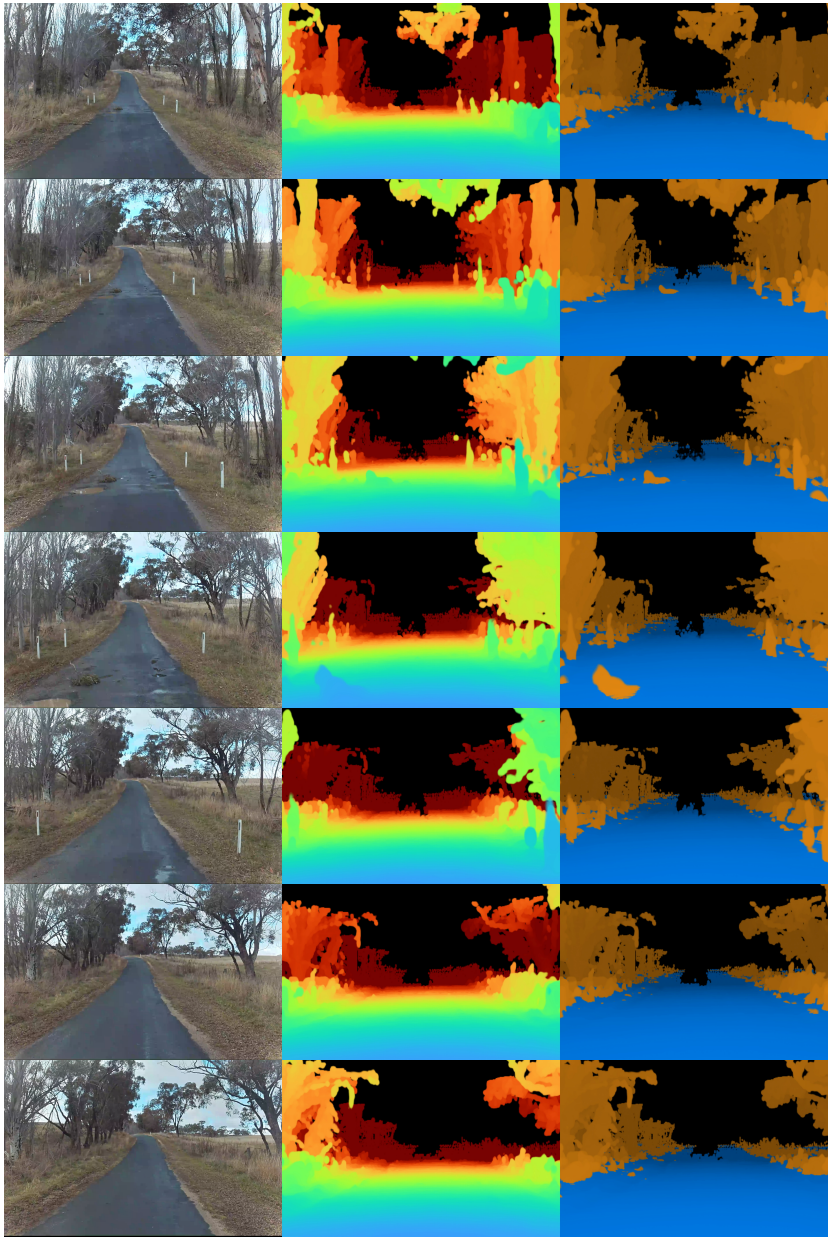

Fig. 14: Zero-shot TriO outputs on a roll-out of frames from a STU validation sequence. The columns include (from left to right): the reference image, the TriO rendered depth image, and the TriO rendered obstacle segmentation image. This example showcases TriO's ability to generalize to unseen country road scenes, even though it was trained on urban data (AV2).

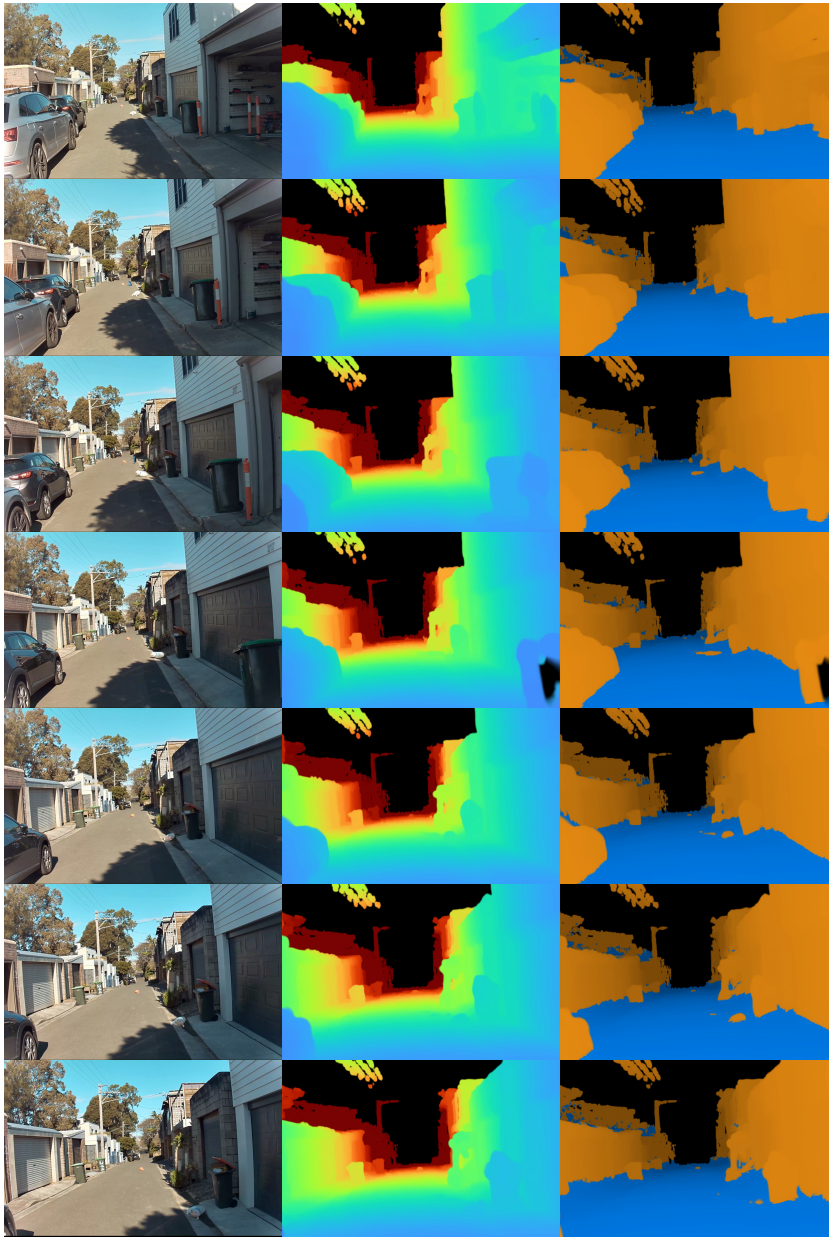

Fig. 15: Zero-shot TrIO outputs on a roll-out of frames from a STU validation sequence. The columns include (from left to right): the reference image, the TrIO rendered depth image, and the TrIO rendered obstacle segmentation image. This figure showcases an example of small debris being perceived as an obstacle - where even moving debris is properly segmented.

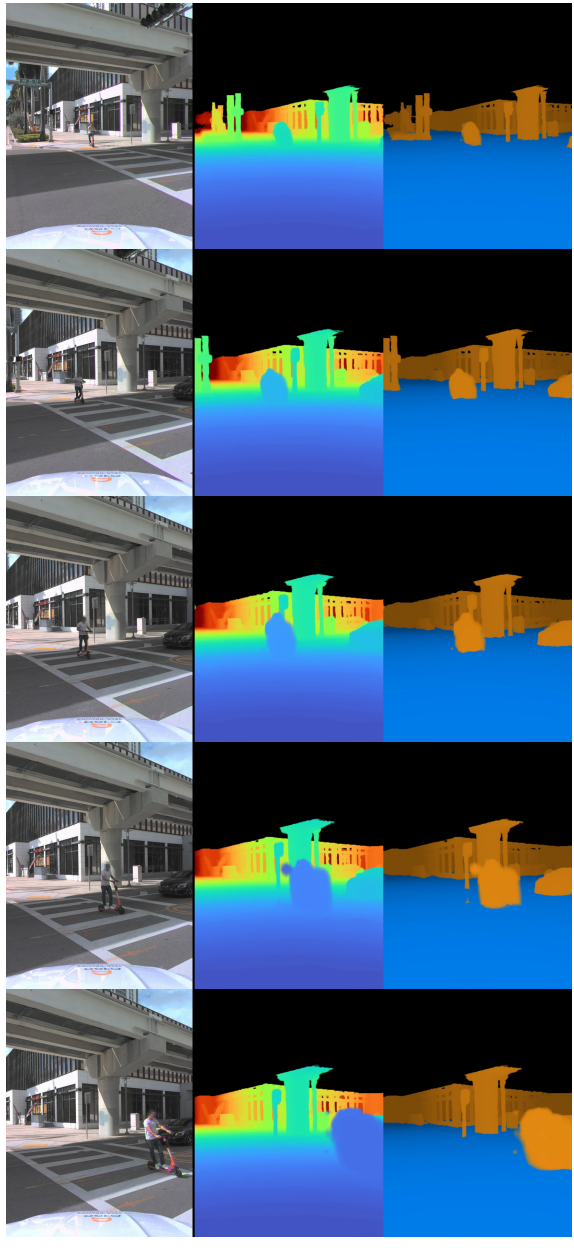

Fig. 16: A qualitative visualization of TrIO outputs on AV2 across a rolled out sequence. The columns include (from left to right): the reference image, the TrIO rendered depth image, and the TrIO rendered obstacle segmentation image. This figure showcases an example of TrIO perceiving a vulnerable road user.

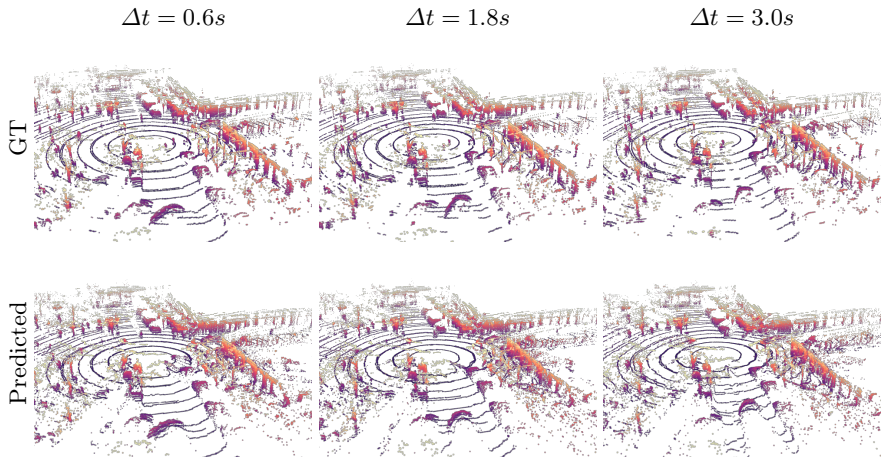

Fig. 17: Ground truth (GT) vs predicted LiDAR of TrIO on AV2.

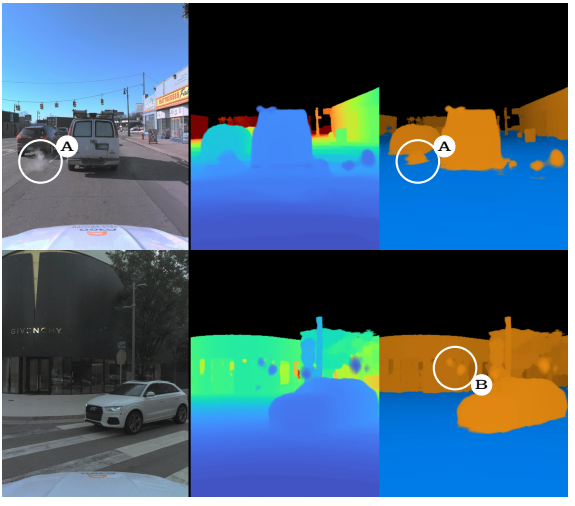

Fig. 18: Visualization of failure cases of TrIO on AV2 . **A)** a false positive obstacle detected from vehicular exhaust. **B)** false positive obstacles detected from snow.

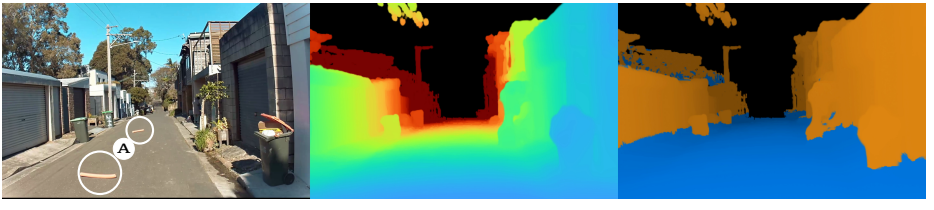

Fig. 19: Visualization of a failure case of TrIO on STU. **A)** what appears to be foam pool noodles that are segmented as ground - a false negative.

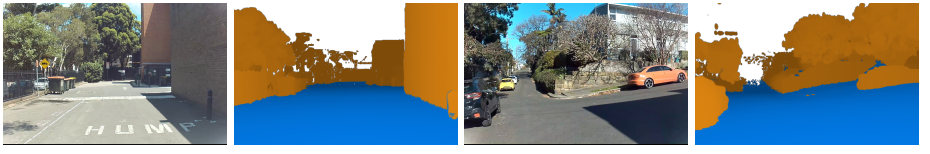

Fig. 20: TriO inference results for 1. speed bump, 2. hilly roads.