RAYLIFT: BEYOND RANK-ONE LIFTING FOR OCCLUSION-AWARE 3D OCCUPANCY PREDICTION

Anonymous authors
Paper under double-blind review

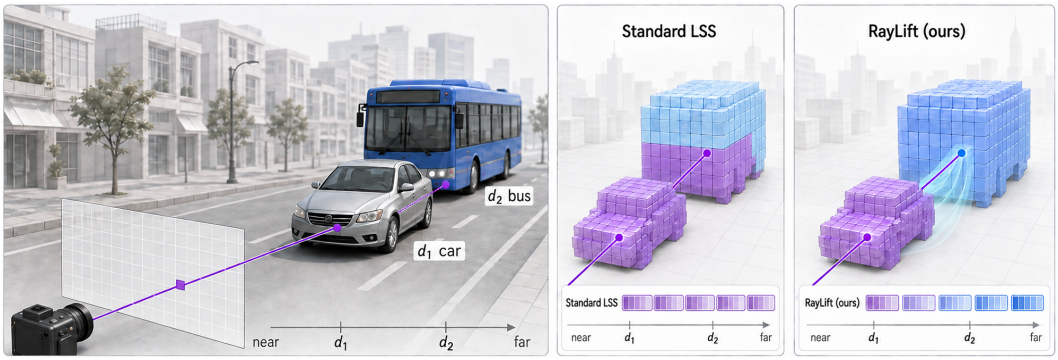

Figure 1: **Beyond rank-one lifting under partial occlusion.** Standard Lift-Splat-Shoot (LSS) produces a rank-one per-ray representation. RayLift relaxes this restriction through multi-feature lifting and adds near-to-far contextual propagation toward occluded voxels.

ABSTRACT

Camera-based 3D semantic occupancy prediction must reconstruct not only visible surfaces, but also geometry and semantics hidden behind them. Consider a pixel ray that first intersects a visible car and then an occluded bus (Fig. 1). Although the image feature at that pixel is dominated by the car, the resulting 3D representation should encode different semantics at the two depths. Standard Lift-Splat-Shoot (LSS) does not provide this distinction during lifting: it copies the same pixel feature to every depth bin and changes only its scalar weight. Consequently, the per-ray lifted representation has rank at most one. When the same depth-independent linear classifier is applied at every depth, all positive-mass depths preserve the same feature-dependent class ordering, so any class-order reversal must arise from subsequent 3D reasoning. We formalize this representational restriction as a rank-collapse obstruction. We introduce **RayLift**, which directly relaxes this restriction and complements it with directed contextual reasoning for occluded regions. Optimal-transport-routed multi-feature lifting assigns each depth bin a depth-dependent mixture of feature heads, enabling different, non-collinear feature vectors along the same ray. Causal ray-radial convolution provides a directed near-to-far pathway for propagating contextual information toward farther, occluded voxels. Visibility-conditioned temporal fusion extends this reasoning across time by conditioning historical voxel features on their predicted visibility. On Occ3D-nuScenes, RayLift achieves $45.4\,\mathrm{RayIoU}$ and $39.2\,\mathrm{mIoU}$, surpassing ALOcc by 1.7 points in overall RayIoU and 1.8 points in $\mathrm{RayIoU}_{1\mathrm{m}}$. On Occ3D-Waymo, RayLift achieves $36.6\,\mathrm{mIoU}$, outperforming ALOcc by 6.5 points under matched camera-only settings.

1 INTRODUCTION

Camera-based 3D semantic occupancy prediction aims to recover a dense voxel grid of geometry and semantics from surrounding images. Such a grid provides a common 3D scene representation for

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

detection (Li et al., 2022; Liu et al., 2022), segmentation (Graham et al., 2018; Tang et al., 2020), and planning, and has therefore become a central representation for autonomous driving (Huang et al., 2023; Wei et al., 2023; Tian et al., 2024). Its main difficulty is occlusion: cameras observe only the nearest visible surfaces, whereas occupancy prediction must also infer the geometry and semantics behind them. For example, a camera ray may first intersect a visible vehicle and then pass through an occluded object at a farther depth. Although these depths may correspond to different semantic classes, they must be inferred from features originating at the same image location. The lifting operation therefore plays a critical role in determining whether the resulting 3D representation can distinguish semantic content at different depths along the ray.

Most camera-based occupancy methods follow the Lift-Splat-Shoot (LSS) paradigm (Philion & Fidler, 2020; Huang et al., 2021). A 2D backbone extracts an image feature map, and a depth head distributes each feature vector over discrete depth bins before the resulting frustum features are splatted into an ego-centric voxel grid for 3D refinement (Li et al., 2023d; Ma et al., 2024; Chen et al., 2025a). Concretely, the lifted feature at depth d is formed as $\mathbf{L}_{u,v,d} = w_{u,v,d}\,\mathbf{f}_{u,v}$, where $\mathbf{f}_{u,v} \in \mathbb{R}^C$ is the feature at image location (u, v) and $w_{u,v,d} \in \mathbb{R}_+$ is its predicted depth weight. The depth weights determine *where* and *how strongly* the feature is placed, but every depth still receives a scaled copy of the same feature vector.

Viewed across depth, these vectors form a matrix of rank at most one: all positive-mass depths receive positive scalar multiples of the same feature vector. We refer to this restriction as a *rank-collapse obstruction*. This structure limits the semantic variation available during lifting and places additional burden on subsequent 3D reasoning. First, under a shared depth-independent linear classifier, all positive-mass depth bins preserve the same class ordering, preventing lifting alone from encoding different semantic preferences at different depths. Second, under the shared local backbone assumption in Sec. 3, a confident class-order reversal requires the local 3D input patches at the two depths to differ sufficiently. Because scalar lifting confines target-ray contributions to the one-dimensional subspace $\mathrm{span}\{\mathbf{f}_{u,v}\}$, any additional separation beyond the variation available within this subspace must be supplied by surrounding context.

We introduce **RayLift**, an occlusion-aware lifting framework that addresses both aspects directly. To increase the representational diversity along each ray, *optimal-transport-routed multi-feature lifting* generates multiple feature heads at each image location and assigns each depth bin a depth-dependent mixture of them. Different depths along the same ray can therefore receive different, non-collinear feature vectors, rather than only scaled copies of a single feature vector. A balanced transport constraint discourages routing collapse, while geometry-guided routing focuses representational capacity on relevant depth regions. A frustum-level semantic loss further encourages these additional feature vectors to encode informative semantic variation across depth. To propagate surrounding context toward occluded regions, *causal ray-radial convolution* resamples the voxel grid into ego-centric polar coordinates and aggregates features causally along the radial dimension, providing an explicit near-to-far pathway toward farther, occluded voxels. As a temporal extension, *visibility-conditioned temporal fusion* augments each historical voxel feature with a predicted visibility cue, allowing the model to distinguish likely visible historical features from those inferred under occlusion.

On Occ3D-nuScenes (Caesar et al., 2020; Tian et al., 2024), RayLift achieves $45.4\,\mathrm{RayIoU}$ and $39.2\,\mathrm{mIoU}$, surpassing ALOcc (Chen et al., 2025a) by $+1.7$ points in overall RayIoU and $+1.8$ points in $\mathrm{RayIoU}_{1\mathrm{m}}$. On Occ3D-Waymo (Sun et al., 2020; Tian et al., 2024), RayLift achieves $36.6\,\mathrm{mIoU}$, outperforming ALOcc by $+6.5$ points under matched camera-only settings. On SurroundOcc (Wei et al., 2023), RayLift achieves $35.5\,\mathrm{IoU}$ and $24.3\,\mathrm{mIoU}$. Under the standard Occ3D camera-visible-mask protocol, RayLift additionally achieves $46.2\,\mathrm{mIoU}$, exceeding ALOcc by $+0.7$ points.

Our contributions are:

- We identify a *rank-collapse obstruction* in scalar LSS lifting and show that, under a shared depth-independent linear classifier, positive-mass depths preserve the same class ordering. We further characterize when additional separation must be supplied by cross-ray context.
- We propose *optimal-transport-routed multi-feature lifting*, which enables depth-dependent non-collinear feature vectors, while balanced routing discourages head collapse and frustum-level supervision encourages semantic specialization across depth.

- We introduce *causal ray-radial convolution* to propagate spatial context toward occluded voxels and *visibility-conditioned temporal fusion* to condition historical features on predicted visibility.
- RayLift achieves state-of-the-art RayIoU on Occ3D-nuScenes, including a $+1.8$-point gain at the strict $1\mathrm{m}$ threshold, and outperforms ALOcc by $+6.5\,\mathrm{mIoU}$ on Occ3D-Waymo.

2 Related Work

Camera-based 3D occupancy prediction. Camera-based 3D semantic occupancy prediction reconstructs dense geometry and semantics from monocular or surround-view images. Early semantic scene completion methods focused on monocular indoor and driving scenes (Cao & De Charette, 2022), while recent methods scale occupancy prediction to multi-camera autonomous-driving settings (Huang et al., 2023; Zhang et al., 2023; Wei et al., 2023; Tong et al., 2023; Tian et al., 2024; Wang et al., 2023). Recent approaches improve 3D reasoning through transformer-based, factorized, low-rank, or sparse representations (Huang et al., 2023; Zhang et al., 2023; Ma et al., 2024; Zhao et al., 2024; Liu et al., 2024a; Wang et al., 2024; Liao et al., 2025). These methods primarily alter the representation or reasoning performed after image features have been mapped into 3D. RayLift instead revisits the feature structure produced during camera-to-3D lifting itself.

Camera-to-3D lifting. Lift-Splat-Shoot (LSS) (Philion & Fidler, 2020) transforms image features into 3D or BEV by combining each pixel feature with a predicted depth distribution. This depth-based lifting paradigm underlies many camera-based perception methods (Huang et al., 2021; Li et al., 2023c;b;d). Subsequent work improves this interface through stronger depth estimation, temporal stereo, view transformation, depth-aware sampling, and adaptive lifting (Li et al., 2023c;b;a;d; Chen et al., 2025a). In particular, ALOcc introduces occlusion-aware adaptive lifting and semantic consistency between image and voxel features (Chen et al., 2025a). These methods primarily improve *where* and *how strongly* image features are placed along depth, while largely retaining scalar modulation of a shared pixel feature. RayLift instead constructs depth-dependent feature representations through multi-feature routing, directly relaxing the rank-one per-ray structure induced by scalar lifting.

Occlusion, ray reasoning, and contextual aggregation. Camera-only occupancy prediction must infer geometry and semantics beyond directly visible surfaces. Prior work improves such reasoning through temporal aggregation, rendering-based supervision, occlusion-aware lifting, and sparse ray-oriented occupancy prediction (Pan et al., 2024; Liu et al., 2024a; Chen et al., 2025a;b). RenderOcc enforces consistency between predicted occupancy and image observations through differentiable rendering (Pan et al., 2024), while SparseOcc introduces sparse occupancy prediction together with ray-oriented evaluation (Liu et al., 2024a). Most occupancy backbones aggregate spatial context through isotropic convolutions, BEV encoders, or transformers (Wei et al., 2023; Zhang et al., 2023; Li et al., 2023d; Ma et al., 2024). RayLift instead couples the lifting-stage rank analysis with a directed near-to-far contextual pathway and explicit visibility conditioning for temporal fusion.

3 The Rank-Collapse Obstruction

We analyze a structural limitation of Lift-Splat-Shoot (LSS) lifting and its consequences for distinguishing semantics at different depths. We first establish the rank-one structure induced by scalar lifting. Sec. 3.1 shows that this structure preserves class ordering along a ray and characterizes the separation required to reverse that ordering downstream. Sec. 3.2 then identifies the portion of this separation that must be supplied by cross-ray context.

Preliminaries. Let $\mathbf{f}_{u,v} \in \mathbb{R}^C$ denote the C-dimensional image feature at pixel (u, v). Scalar lifting constructs the frustum feature at depth d as

$$\mathbf{L}_{u,v,d} = w_{u,v,d}\,\mathbf{f}_{u,v} \in \mathbb{R}^C, \tag{3.1}$$

where $w_{u,v,d} \geq 0$ denotes the scalar depth mass assigned to bin d, and $\mathbf{w}_{u,v} := [w_{u,v,1}, \ldots, w_{u,v,D}]^\top \in \mathbb{R}^D_+$ collects the depth masses along the ray. We index the lifted volume in frustum coordinates (u, v, d), where a fixed pixel (u, v) and varying depth d define a camera

ray. For a fixed ray, stacking the lifted features across depth gives $\mathbf{L}_{u,v,:} = \mathbf{w}_{u,v}\mathbf{f}_{u,v}^{\top}$, and hence $\mathrm{rank}(\mathbf{L}_{u,v,:}) \leq 1$. Thus, all positive-mass depth bins receive positive scalar multiples of the same image feature. Scalar lifting can therefore vary where and how strongly the image feature is placed, but cannot introduce independent feature variation across depth. The corresponding derivation and corollary are provided in Appendix B.1.

3.1 Lifting-Stage Class-Order Invariance

For a local 3D input patch P_d centered at the voxel intersected by ray (u, v) at depth d, let the shared 3D backbone produce $\phi(P_d)$. Let $\mathbf{c}_k \in \mathbb{R}^C$ denote the weight vector of the shared linear classifier for semantic class k. The corresponding feature-dependent class score is

$$s_{u,v,d,k} = \langle \phi(P_d), \mathbf{c}_k \rangle. \tag{3.2}$$

The precise backbone assumptions are given in Appendix B.2.

Scalar lifting preserves class ordering. To isolate the semantic variation available at the lifting stage, we consider a shared depth-independent linear classifier applied directly to the lifted features. Since $\mathbf{L}_{u,v,d} = w_{u,v,d}\,\mathbf{f}_{u,v}$,

$$s^{\mathrm{lift}}_{u,v,d,k} = w_{u,v,d}\,\langle \mathbf{f}_{u,v}, \mathbf{c}_k \rangle. \tag{3.3}$$

Proposition 1 (Depth-wise invariance of feature-dependent class ordering)**.** *For any two classes a, b and any depth with $w_{u,v,d} > 0$,*

$$\mathrm{sign}(s^{\mathrm{lift}}_{u,v,d,a} - s^{\mathrm{lift}}_{u,v,d,b}) = \mathrm{sign}(\langle \mathbf{f}_{u,v}, \mathbf{c}_a - \mathbf{c}_b \rangle). \tag{3.4}$$

Hence, all positive-mass depths preserve the same feature-dependent class ordering.

The proof is provided in Appendix B.3. Thus, with the same linear classifier applied at every depth, changing only the scalar depth weight cannot make one depth prefer class a while another prefers class b; any such reversal must arise from subsequent 3D reasoning.

Separation required for class-order reversal. Let $m_{a,b}(d) := s_{u,v,d,a} - s_{u,v,d,b}$. Under the shared L_g-Lipschitz backbone assumption formalized in Appendix B.2, we obtain:

Proposition 2 (Separation required for class-order reversal)**.** *For any $\gamma > 0$ and classes a, b with $\mathbf{c}_a \neq \mathbf{c}_b$, if*

$$m_{a,b}(d_1) \geq \gamma, \qquad m_{a,b}(d_2) \leq -\gamma, \tag{3.5}$$

then

$$\|\phi(P_{d_1}) - \phi(P_{d_2})\| \geq \frac{2\gamma}{\|\mathbf{c}_a - \mathbf{c}_b\|}, \qquad \|P_{d_1} - P_{d_2}\|_F \geq \underbrace{\frac{2\gamma}{L_g\,\|\mathbf{c}_a - \mathbf{c}_b\|}}_{\Delta_{\mathrm{req}}}. \tag{3.6}$$

The proof is provided in Appendix B.4. Intuitively, reversing the class preference between two depths with margin γ requires a change of at least 2γ in their class margins. Since the shared 3D backbone is L_g-Lipschitz, the corresponding input patches must differ by at least Δ_{req}. However, scalar lifting restricts the target ray to scaled copies of a single feature vector, so any required variation beyond this one-dimensional subspace must come from surrounding context.

3.2 Contextual Separation under Rank-One Lifting

We next characterize when the required patch separation must include variation outside the subspace spanned by the target ray's image feature. For $\mathbf{f}_{u,v} \neq \mathbf{0}$, let $\Pi_{u,v}, \Pi^{\perp}_{u,v} \in \mathbb{R}^{C\times C}$ denote the projections onto the subspace spanned by $\mathbf{f}_{u,v}$ and its orthogonal complement, respectively:

$$\Pi_{u,v} := \frac{\mathbf{f}_{u,v}\,\mathbf{f}_{u,v}^{\top}}{\|\mathbf{f}_{u,v}\|^2}, \qquad \Pi^{\perp}_{u,v} := I - \Pi_{u,v}. \tag{3.7}$$

Define

$$\Delta_{\parallel} := \|(P_{d_1} - P_{d_2})\,\Pi_{u,v}\|_F. \tag{3.8}$$

Here, $\Delta_{\|}$ measures the difference between the two input patches within the subspace spanned by $\mathbf{f}_{u,v}$.

We decompose each local patch as $P_d = P_d^{\text{ray}} + P_d^{\text{off}}$, where the two terms collect contributions from the target ray and all other rays, respectively. Since scalar lifting confines target-ray contributions to $\text{span}\{\mathbf{f}_{u,v}\}$, we have $P_d^{\text{ray}}\Pi_{u,v}^{\perp} = 0$. Combining this restriction with Proposition 2 gives:

Corollary 1 (Cross-ray separation requirement)**.** *If the downstream class ordering reverses as in Proposition 2, then*

$$\|(P_{d_1}^{\text{off}} - P_{d_2}^{\text{off}})\,\Pi_{u,v}^{\perp}\|_F \geq \sqrt{\left[\Delta_{\text{req}}^2 - \Delta_{\|}^2\right]_+}, \tag{3.9}$$

where $[x]_+ := \max(x, 0)$.

The proof is provided in Appendix B.5. When $\Delta_{\|} < \Delta_{\text{req}}$, a confident class-order reversal requires additional variation orthogonal to the target ray's image feature. Since scalar lifting cannot produce this variation, it must be supplied by surrounding rays.

These results motivate two complementary components of RayLift. Multi-feature lifting enables depth-dependent, non-collinear feature vectors along the target ray, while causal ray-radial convolution propagates surrounding context from nearer to farther voxels.

4 RayLift

We present RayLift, an occlusion-aware framework for lift-splat occupancy prediction that addresses the two limitations identified in Sec. 3. Optimal-transport-routed multi-feature lifting (Sec. 4.1) relaxes the rank-one restriction by assigning depth-dependent mixtures of feature heads before splatting, enabling non-collinear feature vectors along the same ray. Causal ray-radial convolution (Sec. 4.2) provides a directed near-to-far pathway for propagating spatial context toward occluded voxels. Visibility-conditioned temporal fusion (Sec. 4.3) extends this reasoning across time by conditioning historical voxel features on their predicted visibility. Figure 2 summarizes the framework.

4.1 Optimal-Transport-Routed Multi-Feature Lifting

The analysis in Sec. 3 shows that scalar lifting restricts all positive-mass depths on a ray to scaled copies of the same image feature. To relax this rank-one restriction, RayLift transforms the image feature $\mathbf{f}_{u,v} \in \mathbb{R}^C$ at pixel (u, v) into G feature heads:

$$\mathbf{F}_{u,v}^{(g)} = \Phi_g(\mathbf{f}_{u,v}) \in \mathbb{R}^C, \qquad g = 1, \ldots, G, \tag{4.1}$$

where Φ_g is a learned feature transformation for head g. We assign each depth bin d a conditional routing distribution $p_{u,v}(g \mid d)$, which specifies the weight of feature head g at that depth. Let $\boldsymbol{\sigma}_{u,v} \in \mathbb{R}_+^D$ denote the lifting masses predicted by the geometry branch over the D depth bins. We construct the lifted feature as

$$\mathbf{L}_{u,v,d} = \sigma_{u,v,d} \sum_{g=1}^{G} p_{u,v}(g \mid d)\, \mathbf{F}_{u,v}^{(g)}, \tag{4.2}$$

where $\sigma_{u,v,d}$ controls the contribution at depth bin d. By varying the routing distribution across depths, RayLift can produce non-collinear feature vectors along the same ray.

Geometry-conditioned optimal transport. To obtain depth-dependent routing, the distribution $p_{u,v}(g \mid d)$ should reflect both the image content and the predicted geometry at each depth. We therefore construct head–depth affinities $\mathbf{A}_{u,v} \in \mathbb{R}^{G \times D}$ from image-conditioned head queries and geometry-conditioned depth keys, where $A_{u,v,g,d}$ measures the compatibility between feature head g and depth bin d.

We formulate routing as a balanced optimal-transport problem that distributes the normalized predicted depth mass among the feature heads according to these affinities. Specifically, the transport plan $\boldsymbol{\Gamma}_{u,v} \in \mathbb{R}_+^{G \times D}$ assigns mass $\Gamma_{u,v,g,d}$ from depth bin d to head g:

$$\begin{aligned} \boldsymbol{\Gamma}_{u,v} &= \arg\max_{\boldsymbol{\Gamma} \geq 0} \langle \boldsymbol{\Gamma}, \mathbf{A}_{u,v} \rangle + \tau H(\boldsymbol{\Gamma}) \\ &\text{s.t.} \quad \boldsymbol{\Gamma}\mathbf{1}_D = \boldsymbol{\mu}, \quad \boldsymbol{\Gamma}^\top \mathbf{1}_G = \boldsymbol{\nu}_{u,v}. \end{aligned} \tag{4.3}$$

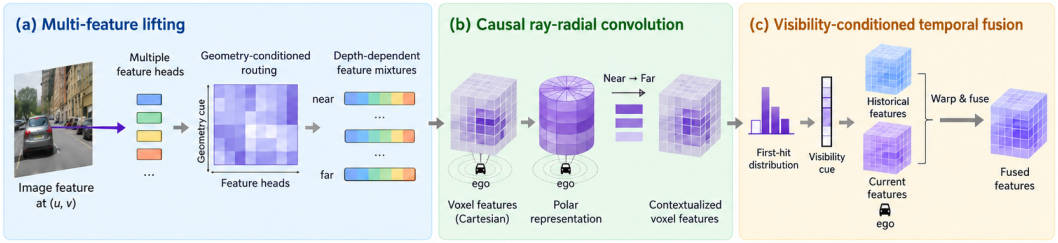

Figure 2: **Overview of RayLift.** (a) Geometry-conditioned affinities define a balanced optimal-transport plan that routes depth bins among multiple feature heads, enabling depth-dependent feature mixtures before splatting. (b) Causal ray-radial convolution resamples voxel features into polar coordinates and propagates context from smaller to larger radius through a residual branch. (c) Visibility-conditioned temporal fusion derives a voxel visibility cue from the first-hit distribution and augments historical voxel features with this cue before warping and fusion.

Here, H denotes transport entropy and $\tau > 0$ controls its strength. The head marginal $\boldsymbol{\mu}$ is uniform with $\mu_g = 1/G$, while the depth marginal $\boldsymbol{\nu}_{u,v}$ is obtained by normalizing the predicted lifting masses $\boldsymbol{\sigma}_{u,v}$. The all-ones vectors $\mathbf{1}_D$ and $\mathbf{1}_G$ sum over depths and heads, respectively. The objective favors compatible head–depth assignments, while the constraints preserve the predicted depth distribution and balance total mass across heads.

For depth bins with $\nu_{u,v,d} > 0$, normalizing each depth column of the transport plan gives the conditional routing distribution

$$p_{u,v}(g \mid d) = \frac{\Gamma_{u,v,g,d}}{\nu_{u,v,d}}. \tag{4.4}$$

These weights determine the depth-dependent feature mixtures in Eq. 4.2. Affinity construction, numerical stabilization, and the Sinkhorn solver are detailed in Appendix C.1.

Per-ray rank capacity. Let $\mathbf{R}_{u,v} \in \mathbb{R}^{D\times G}$ contain the routing weights, with $[\mathbf{R}_{u,v}]_{d,g} = p_{u,v}(g \mid d)$. Let $\mathbf{H}_{u,v} \in \mathbb{R}^{G\times C}$ stack the feature heads as rows, with $[\mathbf{H}_{u,v}]_{g,:} = (\mathbf{F}^{(g)}_{u,v})^\top$. The per-ray lifted representation can then be written as

$$\mathbf{L}_{u,v,:} = \operatorname{diag}(\boldsymbol{\sigma}_{u,v})\, \mathbf{R}_{u,v}\, \mathbf{H}_{u,v}, \qquad \operatorname{rank}(\mathbf{L}_{u,v,:}) \le G. \tag{4.5}$$

Unlike scalar lifting, MFL can therefore produce higher-rank representations along the same ray. When $G = 1$, the representation has rank at most one, recovering the original restriction.

Frustum-level supervision. The higher-rank capacity provided by MFL does not itself guarantee meaningful depth-wise specialization. We therefore supervise semantic predictions from the routed frustum features before splatting using the same mask and dice objectives as the occupancy head:

$$\mathcal{L}_{\text{frust}} = \mathcal{L}^{\text{frust}}_{\text{mask}} + \mathcal{L}^{\text{frust}}_{\text{dice}}. \tag{4.6}$$

Depth-specific target construction, sampling, and loss weighting are detailed in Appendix D.2.

4.2 Causal Ray-Radial Convolution

Increasing on-ray feature diversity alone does not explicitly propagate surrounding context toward occluded voxels. Standard 3D convolutions aggregate local features without enforcing the near-to-far asymmetry induced by occlusion. We therefore introduce a causal radial branch that propagates spatial context from smaller to larger ego-centric radii after splatting.

Causal radial aggregation. Given an ego-centric voxel feature volume $\mathbf{V} \in \mathbb{R}^{C\times X\times Y\times Z}$, where C is the feature dimension and X, Y, and Z are the Cartesian grid dimensions, we resample it into a polar representation $\mathbf{P} \in \mathbb{R}^{C\times N_R\times N_\theta\times Z}$, where N_R and N_θ are the numbers of radial and azimuthal bins. We then apply a causal convolution along the radial dimension:

$$\mathbf{P}'_{:,r,\theta,z} = \psi\left(\sum_{k=0}^{K-1} \mathbf{W}_k\, \mathbf{P}_{:,r-k\delta,\theta,z}\right), \tag{4.7}$$

where r, θ, and z index radius, azimuth, and height, K is the radial kernel size, δ is the dilation, $\mathbf{W}_k$ are learnable channel-mixing weights, and ψ denotes normalization followed by activation. The causal structure restricts each output to features at smaller or equal radii, inducing near-to-far propagation. After multi-camera splatting, ego-centric radius serves as a shared approximation to viewing-direction depth in the ground plane.

The processed features are resampled back to the Cartesian grid and added residually:

$$\mathbf{V}_{\text{out}} = \mathbf{V} + \alpha\, \mathcal{S}_{\text{cart}}(\mathbf{P}'), \tag{4.8}$$

where $\mathcal{S}_{\text{cart}}$ denotes polar-to-Cartesian resampling and α is a learnable scalar initialized to zero. Implementation details of the polar grid and CRRC placement are provided in Appendix C.2.

4.3 Visibility-Conditioned Temporal Fusion

Historical frames can provide direct evidence for voxels that are currently occluded, but feature-only temporal fusion does not explicitly distinguish such observations from features inferred behind an earlier surface. We therefore augment each frame's voxel features with a predicted visibility cue.

Let $\boldsymbol{\rho}_{u,v} \in \Delta^{D-1}$ denote the first-hit depth distribution predicted by the stereo geometry branch at pixel (u, v). The probability that depth bin d is not behind an earlier predicted surface is

$$T_{u,v,d} = 1 - \sum_{i<d} \rho_{u,v,i}. \tag{4.9}$$

For each frame t, we splat this quantity into a voxel visibility volume $v_t \in [0,1]^{X\times Y\times Z}$ using the same frustum-to-voxel mapping. We augment the voxel features by channel-wise concatenation:

$$\widetilde{\mathbf{V}}_t = [\mathbf{V}_t; v_t]. \tag{4.10}$$

The augmented historical features are then ego-motion-warped into the current frame and fused with the current augmented features through the existing temporal pathway, allowing the model to distinguish likely visible historical evidence from features inferred under occlusion. Voxel visibility pooling, gradient detachment, and temporal warping are detailed in Appendix C.3.

5 Experiments

5.1 Experimental Setup

Dataset and metrics. We evaluate RayLift on Occ3D-nuScenes (Tian et al., 2024), which contains 700 training and 150 validation scenes with dense semantic occupancy annotations. The voxel grid covers $[-40, 40]\,\mathrm{m}$ in the horizontal plane and $[-1, 5.4]\,\mathrm{m}$ vertically at a resolution of $0.4\,\mathrm{m}$, yielding a $200 \times 200 \times 16$ grid. We use surround-view camera inputs only. All Occ3D-nuScenes experiments in the main paper are trained without camera-visible-mask supervision. Our primary metric is RayIoU (Liu et al., 2024a), which compares the semantic class and first-hit depth obtained by ray casting through the predicted and ground-truth occupancy volumes. We report $\mathrm{RayIoU}_{1\mathrm{m}}$, $\mathrm{RayIoU}_{2\mathrm{m}}$, $\mathrm{RayIoU}_{4\mathrm{m}}$, and their mean. We additionally report voxel mIoU over the 17 semantic classes.

Implementation details. We use a ResNet-50 image backbone with an FPN neck at an input resolution of 704×256. The main model uses a 16-frame temporal configuration. The stereo geometry branch predicts $D = 88$ depth bins over $[1, 45)\,\mathrm{m}$ with a spacing of $0.5\,\mathrm{m}$. Multi-feature lifting uses $G = 8$ feature heads and solves the transport problem using $n_{\text{sink}} = 5$ log-domain Sinkhorn iterations with temperature $\tau = 1.0$. CRRC uses a radial kernel size of 5, dilation 2, and 180 azimuth bins, and is inserted after the second and third stages of the 3D backbone. We train for 12 epochs using AdamW with a total batch size of 16 across eight GPUs and maintain an exponential moving average of the model weights. Further architectural, optimization, temporal-fusion, and runtime details are provided in Appendix D.

Table 1: RayIoU (Liu et al., 2024a) results on Occ3D-nuScenes (Tian et al., 2024). All methods are trained without camera-visible-mask supervision. FPS is measured using a single A100 GPU with PyTorch FP32 backend for all experiments. Baseline results are taken from Yu et al. (2024); Wang et al. (2024) or the respective original papers.

Method	Backbone	Input Size	mIoU(%)↑	RayIoU(%)↑	RayIoU_{1m}(%)↑	RayIoU_{2m}(%)↑	RayIoU_{4m}(%)↑	FPS
OccFormer (Zhang et al., 2023)	R101	960 × 640	21.9	-	-	-	-	-
TPVFormer (Huang et al., 2023)	R101	960 × 640	27.8	-	-	-	-	-
BEVFormer (Li et al., 2022)	R101	1600 × 900	23.7	32.4	26.1	32.9	38.0	3.0
BEVDetOcc-SF (Huang & Huang, 2022)	R50	704 × 384	24.3	35.2	31.2	35.9	38.4	4.4
FB-Occ (Li et al., 2023d)	R50	704 × 256	31.1	39.0	33.0	40.0	44.0	10.3
SparseOcc (Liu et al., 2024a)	R50	704 × 256	30.6	35.1	29.1	35.8	40.3	12.5
Panoptic-FlashOcc (Yu et al., 2024)	R50	704 × 256	31.6	38.5	32.8	39.3	43.4	35.6
OPUS (Wang et al., 2024)	R50	704 × 256	36.2	41.2	34.7	42.1	46.7	7.2
ALOcc (Chen et al., 2025a)	R50	704 × 256	38.0	43.7	37.8	44.7	48.8	6.0
STCOcc (Liao et al., 2025)	R50	704 × 256	-	41.7	36.2	42.7	46.4	-
HBEVOcc (Lyu et al., 2026)	R50	704 × 256	36.5	41.5	36.1	42.2	46.3	4.5
RayLift (Ours)	R50	704 × 256	**39.2**	**45.4**	**39.6**	**46.4**	**50.2**	5.2

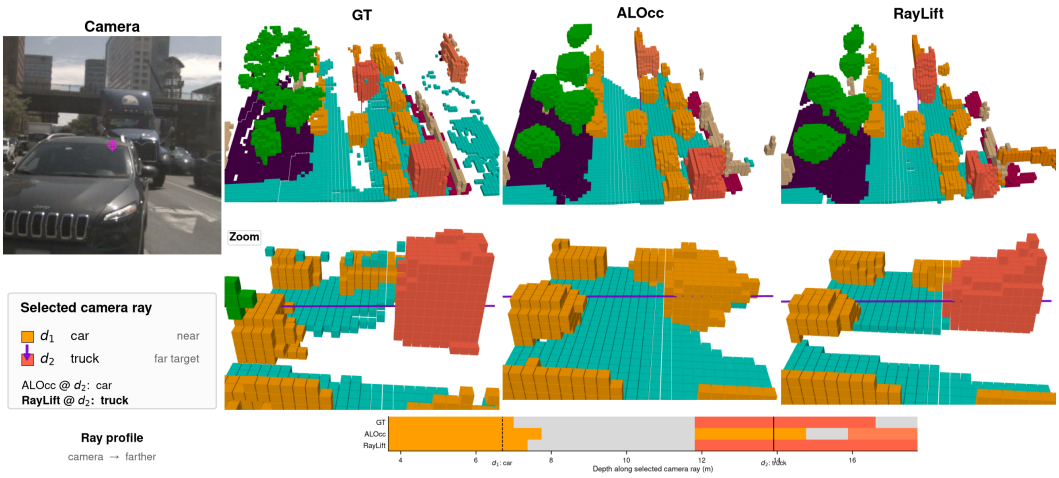

Figure 3: **Qualitative comparison on a multi-class occluded ray.** The selected camera ray intersects a nearer car at d_1 and a farther truck at d_2. ALOcc predicts the nearer car class again at d_2, whereas RayLift recovers the farther truck semantics. The magenta marker and purple line indicate the selected ray, and the bottom profile shows the corresponding predictions along depth.

5.2 Main Results

Table 1 compares RayLift with recent camera-only occupancy methods. With a ResNet-50 backbone and 704×256 input resolution, RayLift achieves 45.4 RayIoU and 39.2 mIoU. It exceeds the strongest comparable baseline, ALOcc (Chen et al., 2025a), by 1.7 points in overall RayIoU and 1.8 points in RayIoU_{1m}. For comparison, results with camera-visible-mask supervision are reported separately in Appendix E.1.

Additional benchmarks. We further evaluate RayLift under matched camera-only settings on SurroundOcc and Occ3D-Waymo. On SurroundOcc, RayLift achieves 35.5 occupied IoU and 24.3 mIoU, compared with 34.7 and 24.0 for ALOcc. On Occ3D-Waymo, RayLift achieves 36.6 mIoU, outperforming ALOcc by 6.5 points. Full benchmark settings and per-class results are provided in Appendix E.2 and Appendix E.3.

Qualitative results. Figure 3 shows a representative multi-class occlusion case matching the along-ray setting motivating RayLift. ALOcc repeats the nearer car label at d_2, whereas RayLift recovers the farther truck semantics. The ray profile shows the same behavior directly along depth. Additional qualitative comparisons and visualizations are provided in Appendix E.5.

Table 2: **Ablation of the two primary design components of RayLift.** MFL denotes optimal-transport-routed multi-feature lifting, while CRRC denotes causal ray-radial convolution.

MFL	CRRC	mIoU	RayIoU	$\mathrm{RayIoU}_{1\mathrm{m}}$	$\mathrm{RayIoU}_{2\mathrm{m}}$	$\mathrm{RayIoU}_{4\mathrm{m}}$
		38.2	43.9	38.2	44.8	48.8
✓		38.8	44.8	39.2	45.8	49.4
	✓	38.8	44.4	38.5	45.3	49.2
✓	✓	**39.2**	**45.4**	**39.6**	**46.4**	**50.2**

Table 3: **Ablation of supporting supervision and temporal visibility conditioning.**

Variant	mIoU	RayIoU	$\mathrm{RayIoU}_{1\mathrm{m}}$	$\mathrm{RayIoU}_{2\mathrm{m}}$	$\mathrm{RayIoU}_{4\mathrm{m}}$
RayLift	**39.2**	**45.4**	**39.6**	**46.4**	**50.2**
w/o frustum loss	39.0	45.0	39.2	46.0	49.9
w/o visibility fusion	38.9	44.9	39.0	45.8	49.9

5.3 Ablation Study

Unless otherwise specified, all ablations use the same ResNet-50 backbone, 16-frame input, and 12-epoch training schedule. Table 2 retains visibility fusion and uses the default MFL frustum supervision; both are isolated in Table 3.

Multi-feature lifting and contextual propagation. Table 2 evaluates MFL and CRRC. The rank-one baseline obtains $43.9\,\mathrm{RayIoU}$. MFL provides the larger standalone gain, reaching 44.8, while CRRC alone reaches 44.4. Adding CRRC on top of MFL further improves performance to 45.4, suggesting that richer on-ray representations and additional spatial context are complementary. At the strict $\mathrm{RayIoU}_{1\mathrm{m}}$ threshold, the complete model improves from 38.2 to 39.6, indicating that the gain is retained under tighter first-hit depth tolerance.

Per-ray representation analysis. To examine whether the trained model uses the additional representational capacity of MFL, we analyze the lifted features on GT-selected multi-class rays before voxel pooling. Compared with scalar lifting, RayLift increases the effective rank from 1.00 to 3.52 and reduces cross-class inter-depth cosine similarity from 1.00 to 0.68 (Appendix E.4).

Frustum supervision and temporal visibility. Table 3 evaluates the two supporting components. Removing frustum supervision reduces RayIoU from 45.4 to 45.0, suggesting that directly supervising the routed frustum features helps MFL learn useful depth-dependent representations before splatting. Removing visibility-conditioned temporal fusion reduces RayIoU to 44.9, with a larger drop at the strict $\mathrm{RayIoU}_{1\mathrm{m}}$ threshold from 39.6 to 39.0. This supports using predicted visibility as an additional cue when fusing historical voxel features.

6 Conclusion

We identified a rank-collapse obstruction in lift-splat occupancy prediction: scalar lifting restricts the features along each camera ray to scaled copies of a single image representation, limiting depth-dependent semantic variation before 3D reasoning. RayLift addresses this restriction through optimal-transport-routed multi-feature lifting, directed near-to-far contextual propagation, and visibility-conditioned temporal fusion. On Occ3D-nuScenes, RayLift achieves $45.4\,\mathrm{RayIoU}$ and $39.2\,\mathrm{mIoU}$, including a 1.8-point gain over ALOcc at the strict $\mathrm{RayIoU}_{1\mathrm{m}}$ threshold. On Occ3D-Waymo, RayLift further improves mIoU over ALOcc by 6.5 points under matched camera-only settings. Together, the representation analysis and experimental results support depth-dependent feature lifting as a complementary design direction to spatial and temporal reasoning for 3D occupancy prediction.

REFERENCES

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

Anh-Quan Cao and Raoul De Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022.

Dubing Chen, Jin Fang, Wencheng Han, Xinjing Cheng, Junbo Yin, Chenzhong Xu, Fahad Shahbaz Khan, and Jianbing Shen. Alocc: adaptive lifting-based 3d semantic occupancy and cost volume-based flow prediction. In *ICCV*, 2025a.

Dubing Chen, Huan Zheng, Yucheng Zhou, Xianfei Li, Wenlong Liao, Tao He, Pai Peng, and Jianbing Shen. Semantic causality-aware vision-based 3d occupancy prediction. In *ICCV*, 2025b.

Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018.

Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022.

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

Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, 2023.

Hongyang Li, Hao Zhang, Zhaoyang Zeng, Shilong Liu, Feng Li, Tianhe Ren, and Lei Zhang. Dfa3d: 3d deformable attention for 2d-to-3d feature lifting. In *ICCV*, pp. 6684–6693, 2023a.

Jinke Li, Xiao He, Chonghua Zhou, Xiaoqiang Cheng, Yang Wen, and Dan Zhang. Viewformer: Exploring spatiotemporal modeling for multi-view 3d occupancy perception via view-guided transformers. In *ECCV*, 2024.

Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *AAAI*, 2023b.

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

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

Zhiqi Li, Zhiding Yu, Wenhai Wang, Anima Anandkumar, Tong Lu, and Jose M Alvarez. Fb-bev: Bev representation from forward-backward view transformations. In *ICCV*, pp. 6919–6928, 2023d.

Zhimin Liao, Ping Wei, Shuaijia Chen, Haoxuan Wang, and Ziyang Ren. Stcocc: Sparse spatial-temporal cascade renovation for 3d occupancy and scene flow prediction. In *CVPR*, 2025.

Haisong Liu, Yang Chen, Haiguang Wang, Zetong Yang, Tianyu Li, Jia Zeng, Li Chen, Hongyang Li, and Limin Wang. Fully sparse 3d occupancy prediction. In *ECCV*, pp. 54–71, 2024a.

Lizhe Liu, Bohua Wang, Hongwei Xie, Daqi Liu, Li Liu, Zhiqiang Tian, Kuiyuan Yang, and Bing Wang. Surroundsdf: Implicit 3d scene understanding based on signed distance field. In *CVPR*, 2024b.

Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, 2022.

Chuandong Lyu, Wenkai Li, Iman Yi Liao, Feng Ding, Hao Liu, and Hongchao Zhou. Hbevocc: Height-aware bird's-eye-view representation for 3d occupancy prediction from multi-camera images. *Sensors*, 26(3):934, 2026. doi: 10.3390/s26030934.

Qihang Ma, Xin Tan, Yanyun Qu, Lizhuang Ma, Zhizhong Zhang, and Yuan Xie. Cotr: Compact occupancy transformer for vision-based 3d occupancy prediction. In *CVPR*, pp. 19936–19945, 2024.

Junhong Min, Youngpil Jeon, Jimin Kim, and Minyong Choi. S^2M^2: Scalable stereo matching model for reliable depth estimation. In *ICCV*, 2025.

Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Hongwei Xie, Bing Wang, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12404–12411. IEEE, 2024.

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

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, 2020.

Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. In *NeurIPS*, 2024.

Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *ICCV*, 2023.

Jiabao Wang, Zhaojiang Liu, Qiang Meng, Liujiang Yan, Ke Wang, Jie Yang, Wei Liu, Qibin Hou, and Mingming Cheng. Opus: Occupancy prediction using a sparse set. In *NeurIPS*, 2024.

Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In *ICCV*, 2023.

Yuping Wang, Xiangyu Huang, Xiaokang Sun, Mingxuan Yan, Shuo Xing, Zhengzhong Tu, and Jiachen Li. Uniocc: A unified benchmark for occupancy forecasting and prediction in autonomous driving. In *ICCV*, 2025.

Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, 2023.

Zhangchen Ye, Tao Jiang, Chenfeng Xu, Yiming Li, and Hang Zhao. Cvt-occ: Cost volume temporal fusion for 3d occupancy prediction. In *ECCV*, 2024.

Zichen Yu, Changyong Shu, Jiajun Deng, Kangjie Lu, Zongdai Liu, Jiangyong Yu, Dawei Yang, Hui Li, and Yan Chen. Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv preprint arXiv:2311.12058*, 2023.

Zichen Yu, Changyong Shu, Qianpu Sun, Yifan Bian, Xiaobao Wei, Jiangyong Yu, Zongdai Liu, Dawei Yang, Hui Li, and Yan Chen. Panoptic-flashocc: An efficient baseline to marry semantic occupancy with panoptic via instance center. *arXiv preprint arXiv:2406.10527*, 2024.

Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In *ICCV*, 2023.

Linqing Zhao, Xiuwei Xu, Ziwei Wang, Yunpeng Zhang, Borui Zhang, Wenzhao Zheng, Dalong Du, Jie Zhou, and Jiwen Lu. Lowrankocc: tensor decomposition and low-rank recovery for vision-based 3d semantic occupancy prediction. In *CVPR*, 2024.

Xubo Zhu, Haoyang Zhang, Fei He, Rui Wu, Yanhu Shan, Wen Yang, and Huai Yu. Dr. occ: Depth- and region-guided 3d occupancy from surround-view cameras for autonomous driving. In *CVPR*, 2026.

A USE OF LARGE LANGUAGE MODELS (LLMS)

In accordance with the ICLR 2027 policy on the use of large language models, we disclose that generative AI tools were used during the preparation of this work for language editing and manuscript organization, code review and debugging, and feedback on the presentation and validation of experiments.

The core research problem, theoretical formulation, proposed method, and scientific contributions of this work were developed by the authors. All mathematical claims, implementations, experimental configurations, reported numerical results, and interpretations were independently reviewed and verified by the authors. The authors take full responsibility for the content of this manuscript.

B ADDITIONAL THEORETICAL DETAILS

This appendix provides the formal details supporting the rank-collapse analysis in Sec. 3. We first state the rank-one structure and on-ray collinearity induced by scalar lifting, then provide the assumptions and proofs underlying Sec. 3.1 and Sec. 3.2.

B.1 RANK-ONE STRUCTURE OF SCALAR LIFTING

From Eq. 3.1, the per-ray slice $\mathbf{L}_{u,v,:} \in \mathbb{R}^{D\times C}$ can be expressed as

$$\mathbf{L}_{u,v,:} = \mathbf{w}_{u,v}\, \mathbf{f}_{u,v}^{\top}, \qquad \operatorname{rank}(\mathbf{L}_{u,v,:}) \le 1. \tag{B.1}$$

Every row is a scalar multiple of the same image feature, so the per-ray slice has rank at most one.

Corollary 2 (On-ray collinearity)**.** *For any depth d with $w_{u,v,d} > 0$ and $\mathbf{f}_{u,v} \neq \mathbf{0}$,*

$$\frac{\mathbf{L}_{u,v,d}}{\|\mathbf{L}_{u,v,d}\|} = \frac{\mathbf{f}_{u,v}}{\|\mathbf{f}_{u,v}\|}. \tag{B.2}$$

Hence, the normalized lifted features are identical at all positive-mass depths along the same ray.

B.2 SCOPE OF THE BACKBONE AND PATCH ASSUMPTIONS

Assumption 1 (Shared local Lipschitz backbone)**.** *For each depth d, let $P_d \in \mathbb{R}^{M\times C}$ denote a bounded, center-aligned local patch around the ego-grid voxel intersected by ray (u, v) at depth d. The downstream 3D backbone applies the same location-independent map*

$$\phi\colon \mathbb{R}^{M\times C} \to \mathbb{R}^{C}, \tag{B.3}$$

to each patch P_d. We assume that ϕ is L_g-Lipschitz for some $L_g > 0$:

$$\|\phi(P) - \phi(P')\| \le L_g \, \|P - P'\|_F. \tag{B.4}$$

For notational simplicity, we use the same feature dimension C for the lifted and downstream representations. This assumption captures convolutional occupancy backbones with bounded receptive fields and shared weights. A finite composition of local convolutions with bounded operator norms, pointwise Lipschitz nonlinearities, and residual blocks defines such a map with a finite Lipschitz constant. Fixed normalization statistics at inference time can be absorbed into ϕ, while operations whose statistics depend globally on the spatial domain fall outside the scope of this local assumption.

Although the patches are indexed by the frustum ray and depth (u, v, d), P_d denotes the corresponding neighborhood after splatting into the ego-centric voxel grid. The analysis requires only that the patches at different depths be represented in the same relative coordinate frame and processed by the same map ϕ.

Feature-dependent class scores. The main text considers the feature-dependent class score

$$s_{u,v,d,k} = \langle \phi(P_d), \mathbf{c}_k \rangle, \qquad \mathbf{c}_k \in \mathbb{R}^C. \tag{B.5}$$

If the classifier additionally includes a class-dependent bias b_k, the full class score is

$$\tilde{s}_{u,v,d,k} = \langle \phi(P_d), \mathbf{c}_k \rangle + b_k. \tag{B.6}$$

The lifting-stage class-ordering result in Proposition 1 concerns only feature-dependent scores and excludes class-dependent biases. However, these biases cancel when comparing class margins across depths and therefore do not affect Proposition 2.

Target-ray and off-ray decomposition. For the contextual analysis, we consider pre-backbone patches formed by additive splatting of lifted ray features and decompose each patch as

$$P_d = P_d^{\text{ray}} + P_d^{\text{off}}, \tag{B.7}$$

where $P_d^{\text{ray}} \in \mathbb{R}^{M\times C}$ collects contributions originating from the target ray (u, v) and $P_d^{\text{off}} \in \mathbb{R}^{M\times C}$ collects contributions from all remaining rays.

Under scalar lifting, every contribution from the target ray is proportional to $\mathbf{f}_{u,v}$. Hence there exists a vector $\boldsymbol{\alpha}_d \in \mathbb{R}^M$, determined by the depth weights and splatting locations, such that

$$P_d^{\text{ray}} = \boldsymbol{\alpha}_d \, \mathbf{f}_{u,v}^\top. \tag{B.8}$$

Even when multiple samples from the target ray contribute to the local patch, their contributions remain in $\text{span}\{\mathbf{f}_{u,v}\}$.

B.3 Proof of Proposition 1

Before downstream contextual reasoning, scalar lifting gives

$$\mathbf{L}_{u,v,d} = w_{u,v,d} \, \mathbf{f}_{u,v}. \tag{B.9}$$

Therefore, for classes a and b,

$$s^{\text{lift}}_{u,v,d,a} - s^{\text{lift}}_{u,v,d,b} = w_{u,v,d} \left(\langle \mathbf{f}_{u,v}, \mathbf{c}_a \rangle - \langle \mathbf{f}_{u,v}, \mathbf{c}_b \rangle \right) \tag{B.10}$$

$$= w_{u,v,d} \left(\langle \mathbf{f}_{u,v}, \mathbf{c}_a - \mathbf{c}_b \rangle \right). \tag{B.11}$$

For any positive-mass depth $w_{u,v,d} > 0$, multiplication by $w_{u,v,d}$ cannot change the sign. Thus,

$$\text{sign}(s^{\text{lift}}_{u,v,d,a} - s^{\text{lift}}_{u,v,d,b}) = \text{sign}(\langle \mathbf{f}_{u,v}, \mathbf{c}_a - \mathbf{c}_b \rangle), \tag{B.12}$$

which proves Proposition 1.

B.4 Proof of Proposition 2

Let

$$m_{a,b}(d) := s_{u,v,d,a} - s_{u,v,d,b} = \langle \phi(P_d), \mathbf{c}_a - \mathbf{c}_b \rangle. \tag{B.13}$$

Suppose

$$m_{a,b}(d_1) \geq \gamma, \qquad m_{a,b}(d_2) \leq -\gamma. \tag{B.14}$$

Then

$$m_{a,b}(d_1) - m_{a,b}(d_2) \geq 2\gamma. \tag{B.15}$$

Using Eq. B.13,

$$2\gamma \leq \langle \phi(P_{d_1}) - \phi(P_{d_2}), \mathbf{c}_a - \mathbf{c}_b \rangle \tag{B.16}$$

$$\leq \|\phi(P_{d_1}) - \phi(P_{d_2})\| \, \|\mathbf{c}_a - \mathbf{c}_b\|, \tag{B.17}$$

where the second inequality follows from Cauchy–Schwarz. Therefore,

$$\|\phi(P_{d_1}) - \phi(P_{d_2})\| \geq \frac{2\gamma}{\|\mathbf{c}_a - \mathbf{c}_b\|}, \tag{B.18}$$

which proves the first inequality in Proposition 2.

By Assumption 1,

$$\|\phi(P_{d_1}) - \phi(P_{d_2})\| \leq L_g \, \|P_{d_1} - P_{d_2}\|_F. \tag{B.19}$$

Combining the two inequalities gives

$$\|P_{d_1} - P_{d_2}\|_F \geq \frac{2\gamma}{L_g \, \|\mathbf{c}_a - \mathbf{c}_b\|} = \Delta_{\text{req}}, \tag{B.20}$$

which proves Eq. 3.6.

B.5 Proof of Corollary 1

We now connect the patch-separation requirement of Proposition 2 to the directional restriction imposed by rank-one lifting.

For $\mathbf{f}_{u,v} \neq \mathbf{0}$, define the orthogonal projections

$$\Pi_{u,v} := \frac{\mathbf{f}_{u,v}\,\mathbf{f}_{u,v}^\top}{\|\mathbf{f}_{u,v}\|^2}, \qquad \Pi_{u,v}^\perp := I - \Pi_{u,v}. \tag{B.21}$$

They satisfy

$$\Pi_{u,v}^\top = \Pi_{u,v}, \qquad \Pi_{u,v}^2 = \Pi_{u,v}, \qquad \Pi_{u,v}\,\Pi_{u,v}^\perp = 0. \tag{B.22}$$

Let

$$\Delta P := P_{d_1} - P_{d_2}. \tag{B.23}$$

Since

$$\Delta P = \Delta P\,\Pi_{u,v} + \Delta P\,\Pi_{u,v}^\perp, \tag{B.24}$$

and the two terms are orthogonal under the Frobenius inner product,

$$\|\Delta P\|_F^2 = \|\Delta P\,\Pi_{u,v}\|_F^2 + \|\Delta P\,\Pi_{u,v}^\perp\|_F^2. \tag{B.25}$$

Define

$$\Delta_\parallel := \|\Delta P\,\Pi_{u,v}\|_F. \tag{B.26}$$

Proposition 2 gives

$$\|\Delta P\|_F \geq \Delta_{\mathrm{req}}. \tag{B.27}$$

Substituting into Eq. B.25 therefore yields

$$\|\Delta P\,\Pi_{u,v}^\perp\|_F^2 = \|\Delta P\|_F^2 - \Delta_\parallel^2 \tag{B.28}$$

$$\geq \Delta_{\mathrm{req}}^2 - \Delta_\parallel^2. \tag{B.29}$$

Since the left-hand side is nonnegative,

$$\|\Delta P\,\Pi_{u,v}^\perp\|_F \geq \sqrt{\left[\Delta_{\mathrm{req}}^2 - \Delta_\parallel^2\right]_+}. \tag{B.30}$$

It remains to identify the source of this orthogonal variation. From Eq. B.8,

$$P_d^{\mathrm{ray}}\,\Pi_{u,v}^\perp = \boldsymbol{\alpha}_d\,\mathbf{f}_{u,v}^\top\,\Pi_{u,v}^\perp = 0, \tag{B.31}$$

because $\mathbf{f}_{u,v}^\top\,\Pi_{u,v}^\perp = 0$. Using the decomposition in Eq. B.7,

$$\Delta P\,\Pi_{u,v}^\perp = \left[(P_{d_1}^{\mathrm{ray}} - P_{d_2}^{\mathrm{ray}}) + (P_{d_1}^{\mathrm{off}} - P_{d_2}^{\mathrm{off}})\right]\,\Pi_{u,v}^\perp \tag{B.32}$$

$$= (P_{d_1}^{\mathrm{off}} - P_{d_2}^{\mathrm{off}})\,\Pi_{u,v}^\perp. \tag{B.33}$$

Combining Eq. B.33 with Eq. B.30 gives

$$\|(P_{d_1}^{\mathrm{off}} - P_{d_2}^{\mathrm{off}})\,\Pi_{u,v}^\perp\|_F \geq \sqrt{\left[\Delta_{\mathrm{req}}^2 - \Delta_\parallel^2\right]_+}, \tag{B.34}$$

which proves Corollary 1.

Interpretation and scope. The corollary depends on how much of the difference between the two input patches is aligned with $\mathbf{f}_{u,v}$. This includes contributions from both the target ray and surrounding rays. If $\Delta_\parallel \geq \Delta_{\mathrm{req}}$, the lower bound on orthogonal variation becomes zero. However, satisfying the required patch separation is only a necessary condition for a class-order reversal and does not contradict the lifting-stage invariance established in Proposition 1.

When $\Delta_\parallel < \Delta_{\mathrm{req}}$, additional orthogonal variation is necessary. Since scalar lifting cannot supply this variation along the target ray, it must originate from off-ray contributions.

C RayLift Module Details

This appendix provides the complete architectural details of the three RayLift components introduced in Sec. 4. We first describe the feature-head construction and geometry-conditioned optimal-transport routing used by multi-feature lifting, then detail the causal ray-radial branch and visibility-conditioned temporal fusion.

C.1 Optimal-Transport-Routed Multi-Feature Lifting

This subsection details the feature-head construction, geometry-conditioned routing, and numerical solution of the optimal-transport problem introduced in Sec. 4.1.

Feature heads and routing queries. Given an image feature $\mathbf{f}_{u,v} \in \mathbb{R}^C$, MFL generates $G = 8$ feature heads:

$$\mathbf{F}^{(g)}_{u,v} = \Phi_g(\mathbf{f}_{u,v}) \in \mathbb{R}^C, \qquad g = 1, \dots, G, \tag{C.1}$$

where Φ_g denotes the learned 1×1 projection for head g. The heads are implemented through a single joint projection whose output is reshaped along the head dimension.

Each feature head is then mapped to an image-conditioned routing query:

$$\mathbf{q}_{u,v,g} = Q(\mathbf{F}^{(g)}_{u,v}) + \mathbf{e}_g, \tag{C.2}$$

where $Q : \mathbb{R}^C \to \mathbb{R}^{C_q}$ is a shared two-layer 1×1 projection with a GELU activation, $C_q = 64$, and $\mathbf{e}_g \in \mathbb{R}^{C_q}$ is a learned embedding for head g.

Geometry-conditioned routing keys. The stereo geometry branch predicts depth logits $\ell_{u,v,d}$ from the image features and stereo cost volume. Applying a softmax over the D depth bins gives

$$\rho_{u,v,d} = \mathrm{softmax}_d(\ell_{u,v,d}), \tag{C.3}$$

where $\boldsymbol{\rho}_{u,v} \in \Delta^{D-1}$ is the predicted depth distribution. The branch also predicts occlusion-transfer weights $\mathbf{o}_{u,v}$, which convert the depth distribution $\boldsymbol{\rho}_{u,v}$ into the non-negative lifting masses $\boldsymbol{\sigma}_{u,v} \in \mathbb{R}^D_+$ through the base model's depth-to-occupancy transformation.

We normalize these lifting masses to obtain the depth marginal

$$\nu_{u,v,d} = \frac{\sigma_{u,v,d}}{\sum_{d'=1}^{D} \sigma_{u,v,d'}}. \tag{C.4}$$

Let z_d denote the metric center of depth bin d. We compute the expected depth under this marginal as

$$\bar{z}_{u,v} = \sum_{d=1}^{D} \nu_{u,v,d}\, z_d. \tag{C.5}$$

The relative metric depth $z_d - \bar{z}_{u,v}$ is encoded using Fourier features to obtain $\mathbf{r}_{u,v,d} \in \mathbb{R}^{16}$. We additionally project the occlusion-transfer weights $\mathbf{o}_{u,v}$ into a ray-level geometry feature $\mathbf{c}_{u,v} \in \mathbb{R}^{31}$.

Each depth key combines its predicted depth logit, the ray-level geometry feature, and its relative metric depth:

$$\mathbf{k}_{u,v,d} = E_{\mathrm{geo}}\left([\ell_{u,v,d}; \mathbf{c}_{u,v}; \mathbf{r}_{u,v,d}]\right) \in \mathbb{R}^{C_q}, \tag{C.6}$$

where $[\cdot;\cdot]$ denotes concatenation and E_{geo} is a shared two-layer 1×1 projection with a GELU activation. The geometry inputs to the router are detached from the geometry branch.

Affinity construction and transport routing. Given the image-conditioned queries $\mathbf{q}_{u,v,g}$ and geometry-conditioned depth keys $\mathbf{k}_{u,v,d}$, we compute their compatibility as

$$A_{u,v,g,d} = \frac{s_A}{\sqrt{C_q}} \langle \mathbf{q}_{u,v,g}, \mathbf{k}_{u,v,d} \rangle, \tag{C.7}$$

where $s_A > 0$ is a learnable affinity scale. These affinities form $\mathbf{A}_{u,v} \in \mathbb{R}^{G \times D}$.

We solve the optimal-transport problem in Eq. 4.3 using the depth marginal $\boldsymbol{\nu}_{u,v}$ defined in Eq. C.4 and the uniform head marginal $\mu_g = 1/G$. The transport entropy is defined as

$$H(\boldsymbol{\Gamma}) = -\sum_{g=1}^{G}\sum_{d=1}^{D}\Gamma_{u,v,g,d}\log\Gamma_{u,v,g,d}. \tag{C.8}$$

We use a transport temperature of $\tau = 1.0$. We obtain the conditional routing weights by normalizing each depth column of the transport plan:

$$p_{u,v}(g \mid d) = \frac{\Gamma_{u,v,g,d}}{\sum_{g'=1}^{G}\Gamma_{u,v,g',d} + \epsilon}, \tag{C.9}$$

where $\epsilon = 10^{-6}$ provides numerical stability.

Log-domain Sinkhorn solver. We approximate the transport plan in Eq. 4.3 using five log-domain Sinkhorn iterations, following S^{2}M^2 Min et al. (2025) with our head and depth marginals. Let

$$\mathbf{Z}_{u,v} = \frac{\mathbf{A}_{u,v}}{\tau}. \tag{C.10}$$

For each ray, we initialize the log-domain scaling vectors $\mathbf{a} \in \mathbb{R}^G$ and $\mathbf{b} \in \mathbb{R}^D$ to zero. We then iteratively update

$$b_d \leftarrow \log(\nu_{u,v,d} + \epsilon) - \mathrm{LSE}_g(Z_{u,v,g,d} + a_g), \tag{C.11}$$

$$a_g \leftarrow \log(\mu_g + \epsilon) - \mathrm{LSE}_d(Z_{u,v,g,d} + b_d), \tag{C.12}$$

where LSE denotes log-sum-exp over the indicated dimension. The two updates alternately move the transport plan toward the prescribed depth and head marginals. After the final iteration, we obtain the approximate transport plan

$$\Gamma_{u,v,g,d} = \exp(Z_{u,v,g,d} + a_g + b_d). \tag{C.13}$$

C.2 Causal Ray-Radial Convolution

This subsection details the polar-grid construction, causal radial convolution, and residual integration of CRRC introduced in Sec. 4.2.

Polar grid construction. Let $\mathbf{V} \in \mathbb{R}^{C\times X\times Y\times Z}$ denote the ego-centric Cartesian voxel feature volume, with C feature channels and Cartesian grid dimensions X, Y, and Z. We construct a polar grid centered at the ego vehicle with $N_\theta = 180$ uniformly spaced azimuth bins and N_R radial bins covering the horizontal spatial extent. The radial grid is uniformly sampled from zero to $r_{\max} = \sqrt{(X/2)^2 + (Y/2)^2}$ in voxel-grid units, with $N_R = \mathrm{round}(r_{\max}) + 1$. For radial bin n and azimuth bin m, we obtain the corresponding Cartesian as $x_{n,m} = (X-1)/2 + r_n\cos\theta_m$ and $y_{n,m} = (Y-1)/2 + r_n\sin\theta_m$. Each height plane is then sampled independently:

$$\mathbf{P}_{:,n,m,z} = \mathrm{Bilinear}\left(\mathbf{V}_{:,:,:,z}, x_{n,m}, y_{n,m}\right). \tag{C.14}$$

After causal radial convolution, the processed polar features $\mathbf{P}'$ are mapped back to the Cartesian grid:

$$[\mathcal{S}_{\mathrm{cart}}(\mathbf{P}')]_{:,i,j,z} = \mathrm{Bilinear}\left(\mathbf{P}'_{:,:,:,z}, r_{i,j}, \theta_{i,j}\right), \tag{C.15}$$

where (i,j) indexes a Cartesian grid cell and $(r_{i,j}, \theta_{i,j})$ denotes its corresponding polar coordinates. Both resampling operations use fixed sampling grids and are differentiable with respect to the input features. The resampled features are added residually to $\mathbf{V}$ to obtain $\mathbf{V}_{\mathrm{out}}$ as in Eq. 4.8.

Causal radial operator. For every azimuth–height pair, we apply a one-dimensional causal convolution along radius with kernel size $K = 5$ and dilation $\delta = 2$:

$$\mathbf{P}'_{:,n,m,z} = \psi\left(\sum_{k=0}^{K-1}\mathbf{W}_k\mathbf{P}_{:,n-k\delta,m,z}\right), \tag{C.16}$$

where $\mathbf{W}_k$ are learnable channel-mixing weights and ψ denotes normalization followed by activation. Indices with $n - k\delta < 0$ are zero padded, so the output at radial bin n depends only on features from radial bins $n' \le n$. This yields directed propagation from smaller to larger radii without introducing the reverse dependency.

C.3 Visibility-Conditioned Temporal Fusion

This subsection details the construction of voxel-level visibility cues and their integration into the temporal fusion pathway introduced in Sec. 4.3.

Voxel visibility construction. Each pixel (u, v) and depth bin d define a 3D point along a camera ray. We project these points from all cameras into the ego-centric voxel grid. Let $\mathcal{I}_t(\mathbf{x})$ denote the set of camera–pixel–depth tuples (c, u, v, d) whose projected points fall inside voxel $\mathbf{x}$ at frame t, where c indexes cameras. We assign each voxel the mean visibility of these points:

$$v_t(\mathbf{x}) = \frac{\sum_{(c,u,v,d)\in\mathcal{I}_t(\mathbf{x})} T_{c,u,v,d}}{\max\left(|\mathcal{I}_t(\mathbf{x})|, 1\right)}. \tag{C.17}$$

Voxels receiving no projected points are assigned zero visibility.

Temporal warping and fusion. We detach the visibility cue from the geometry branch and concatenate it with the corresponding voxel feature:

$$\widetilde{\mathbf{V}}_t = [\mathbf{V}_t; \mathrm{stopgrad}(v_t)]. \tag{C.18}$$

Let $\mathcal{W}_{t-k\to t}$ denote ego-motion warping from the historical frame into the current ego coordinate system. Feature and visibility channels are warped jointly:

$$\widehat{\mathbf{V}}_{t-k\to t} = \mathcal{W}_{t-k\to t}\left(\widetilde{\mathbf{V}}_{t-k}\right). \tag{C.19}$$

The temporal fusion network then receives the current voxel representation together with the aligned historical volumes:

$$\mathbf{V}_t^{\mathrm{fused}} = \mathcal{F}_{\mathrm{temp}}\left(\widetilde{\mathbf{V}}_t, \left\{\widehat{\mathbf{V}}_{t-k\to t}\right\}_{k=1}^{N_{\mathrm{hist}}}\right), \tag{C.20}$$

where N_{hist} is the number of historical frames and $\mathcal{F}_{\mathrm{temp}}$ denotes the base temporal fusion operator.

D Implementation and Training Details

This appendix provides shared implementation details and dataset-specific configurations for the experiments in Sec. 5. Additional benchmark protocols and results are provided in Appendix E. Module-specific details of multi-feature lifting, CRRC, and visibility-conditioned temporal fusion are provided in Appendix C.

D.1 Base Architecture and Benchmark Configurations

RayLift builds on a dense lift-splat occupancy architecture with a ResNet-50 image backbone and FPN neck. Image features are lifted into a frustum representation, pooled into an ego-centric voxel grid, and processed by a shared 3D occupancy backbone. The stereo geometry branch retains the depth supervision of the base occupancy model.

For temporal configurations, historical voxel features are aligned to the current ego coordinate system before fusion. When visibility-conditioned fusion is enabled, each frame's voxel features are augmented with a detached visibility cue, as detailed in Appendix C.3. Table 4 summarizes the dataset-specific input, geometry, and architectural configurations.

The main Occ3D-nuScenes RayIoU experiments do not use official camera-visible-mask supervision. A separate experiment using camera-visible-mask supervision is reported in Appendix E.1. Dataset-specific training and evaluation protocols for SurroundOcc and Occ3D-Waymo are provided in Appendix E.

D.2 Training Objectives

RayLift retains the voxel-level occupancy and stereo-depth supervision of the base model (Chen et al., 2025a). We additionally apply frustum-level semantic supervision and, for Occ3D-nuScenes and SurroundOcc, image-level semantic supervision. Both auxiliary objectives use the occupancy

Table 4: **Benchmark configurations.** Input resolution is reported as width $\times$ height. All depth grids use $0.5\,\mathrm{m}$ intervals, and all models use a ResNet-50 image backbone.

Setting	Occ3D-nuScenes	SurroundOcc	Occ3D-Waymo
Cameras	6	6	5
Input resolution	704×256	1600×896	960×640
Depth range (m)	$[1, 45)$	$[1, 60)$	$[1, 58)$
Depth bins	88	118	114
Voxel extent (m)	$[-40, 40]^2 \times [-1, 5.4]$	$[-50, 50]^2 \times [-5, 3]$	$[-40, 40]^2 \times [-1, 5.4]$
Voxel size (m)	0.4	0.5	0.4
Voxel dimensions	$200 \times 200 \times 16$	$200 \times 200 \times 16$	$200 \times 200 \times 16$
Temporal voxel fusion	16 frames	Disabled	16 frames
MFL heads	8	8	4
Visibility fusion	Yes	No	No

head's mask and dice losses, with class-frequency-aware weighting and uncertainty-based point sampling.

For frustum supervision, we project occupied ground-truth voxel centers into each camera and transform their image coordinates according to the applied image augmentation. Projected points are assigned to image feature-map cells, and the nearest point in each cell determines the semantic target and corresponding depth bin. The resulting targets supervise the routed frustum features before voxel pooling, using the mask and dice losses defined in Eq. 4.6.

The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{occ}} + \mathcal{L}_{\mathrm{depth}} + \lambda_{\mathrm{img}}\mathcal{L}_{\mathrm{img}} + \lambda_{\mathrm{frust}}\mathcal{L}_{\mathrm{frust}}, \tag{D.1}$$

where $\mathcal{L}_{\mathrm{occ}}$ and $\mathcal{L}_{\mathrm{depth}}$ denote the base occupancy and depth objectives, respectively. For Occ3D-nuScenes and SurroundOcc, we set $\lambda_{\mathrm{img}} = \lambda_{\mathrm{frust}} = 0.1$. For Occ3D-Waymo, we set $\lambda_{\mathrm{img}} = 0$ and $\lambda_{\mathrm{frust}} = 0.1$.

D.3 Optimization and Training Schedule

Across all three benchmarks, we optimize RayLift using AdamW with an initial learning rate of 2×10^{-4}, weight decay of 0.01, and a total batch size of 16 across eight GPUs. We maintain an exponential moving average (EMA) of the model parameters with decay 0.999 and use the EMA model for evaluation. The main Occ3D-nuScenes model and its ablations are trained for 12 epochs, while the SurroundOcc and Occ3D-Waymo models are trained for 24 and 12 epochs, respectively.

We apply linear learning-rate warmup for the first 200 iterations, starting at 10^{-3} of the initial learning rate, followed by a constant learning rate. Gradients are clipped to a maximum norm of 5. Unless otherwise specified, the Occ3D-nuScenes ablations retain the same optimization, augmentation, geometry supervision, and temporal configuration, changing only the component under investigation.

D.4 Runtime and Hardware

Training is performed using eight NVIDIA A100 GPUs. Inference speed is measured on a single NVIDIA A100 GPU using the PyTorch FP32 backend. Under this setting, the main Occ3D-nuScenes configuration achieves 5.2 FPS.

The additional computation introduced by RayLift comes from the multi-feature lifting heads and the residual CRRC branch, while visibility-conditioned temporal fusion adds one scalar visibility channel per voxel. The corresponding architectural settings are summarized in Table 4, with module details provided in Appendix C.

Table 5: **Semantic occupancy results on Occ3D-nuScenes with camera-visible-mask supervision**. mIoU^m denotes semantic mIoU evaluated on camera-visible voxels, while mIoU^m_D averages the eight dynamic classes following ALOcc (Chen et al., 2025a). All methods use camera-only input and a ResNet-50 backbone. FPS values are taken from the respective papers unless otherwise stated. For ALOcc, we report the result obtained from its released checkpoint and configuration.

Method	Backbone	Input Size	mIoU^m_D (%) ↑	mIoU^m (%) ↑	FPS
BEVDetOcc-SF (Huang et al., 2021)	R50	704 × 256	34.4	41.9	6.5
UniOCC (Wang et al., 2025)	R50	704 × 256	-	39.7	-
FB-Occ (Li et al., 2023d)	R50	704 × 256	34.2	39.8	10.3
SurroundSDF (Liu et al., 2024b)	R50	704 × 256	36.2	42.4	-
FlashOCC (Yu et al., 2023)	R50	704 × 256	24.7	32.0	29.6
COTR (Ma et al., 2024)	R50	704 × 256	38.6	44.5	0.5
ViewFormer (Li et al., 2024)	R50	704 × 256	35.0	41.9	-
OPUS (Wang et al., 2024)	R50	704 × 256	33.3	36.2	8.2
ALOcc (Chen et al., 2025a)	R50	704 × 256	38.9	<u>45.5</u>	6.0
STCOcc (Liao et al., 2025)	R50	704 × 256	<u>39.0</u>	44.6	-
HBEVOcc (Lyu et al., 2026)	R50	704 × 256	38.3	44.0	4.5
Dr.Occ (Zhu et al., 2026)	R50	704 × 256	38.2	44.1	-
RayLift (Ours)	R50	704 × 256	**39.7**	**46.2**	5.2

E Additional Results and Analyses

This appendix complements the main experiments in Sec. 5 with additional evaluations on Occ3D-nuScenes, SurroundOcc, and Occ3D-Waymo. We also provide an empirical analysis of the learned per-ray representations, additional qualitative comparisons on Occ3D-nuScenes, and qualitative visualizations on the other two benchmarks.

E.1 Occ3D-nuScenes: Camera-Mask Semantic Occupancy

For comparison with prior semantic occupancy methods using the official Occ3D camera-visible-mask protocol, we separately train RayLift with camera-visible-mask supervision and evaluate semantic occupancy on camera-visible voxels. This differs from the main RayIoU experiments in Sec. 5.2, which do not use camera-visible-mask supervision during training.

Table 5 reports mIoU^m over the 17 semantic classes and mIoU^m_D over the eight dynamic classes following ALOcc (Chen et al., 2025a), both evaluated on camera-visible voxels. RayLift achieves 46.2 mIoU^m and 39.7 mIoU^m_D, compared with 45.5 and 38.9 for ALOcc, respectively.

E.2 SurroundOcc

We further evaluate RayLift on SurroundOcc (Wei et al., 2023), which provides dense semantic occupancy annotations under a different occupancy construction and evaluation protocol from Occ3D-nuScenes. We use camera-only input and compare against prior methods under their reported SurroundOcc settings.

Table 6 reports occupied IoU, semantic mIoU, mIoU_D, and per-class IoU. RayLift achieves 35.5 occupied IoU and 24.3 mIoU, compared with 34.7 and 24.0 for ALOcc, respectively. This corresponds to improvements of 0.8 and 0.3 points. RayLift also achieves 22.0 mIoU_D.

E.3 Occ3D-Waymo

We further evaluate RayLift on Occ3D-Waymo using camera-only input. Following the benchmark protocol, semantic occupancy is evaluated within the intersection of the camera-visibility and field-of-view masks. The Waymo configuration uses four MFL heads and disables visibility-conditioned temporal fusion, as summarized in Table 4.

Table 6: **Semantic occupancy results on SurroundOcc** (Wei et al., 2023). IoU denotes occupied-voxel IoU, while mIoU denotes the mean IoU over the 16 semantic classes. mIoU_D averages the eight dynamic classes: bicycle, bus, car, construction vehicle, motorcycle, pedestrian, trailer, and truck.

Method	Backbone	IoU(%)	mIoU(%)	mIoU_D(%)	barrier	bicycle	bus	car	cons. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	driv. surf.	other flat	sidewalk	terrain	manmade	vegetation
BEVFormer	R101	30.5	16.8	14.2	14.2	6.6	23.5	28.3	8.7	10.8	6.6	4.1	11.2	17.8	37.3	18.0	22.9	22.2	13.8	22.2
TPVFormer	R101	30.9	17.1	14.0	16.0	5.3	23.9	27.3	9.8	8.7	7.1	5.2	11.0	19.2	38.9	21.3	24.3	23.2	11.7	20.8
SurroundOcc	R101	31.5	20.3	18.4	20.6	11.7	28.1	30.9	10.7	15.1	14.1	12.1	14.4	22.3	37.3	23.7	24.5	22.8	14.9	21.9
GaussianFormer	R101	29.8	17.3	17.3	19.5	11.3	26.1	29.8	10.5	13.8	12.6	8.7	12.7	21.6	39.6	23.3	24.5	23.0	9.6	19.1
GaussianFormer2	R101	31.7	20.8	18.7	21.4	13.4	**28.5**	30.8	10.9	15.8	13.6	10.5	14.0	22.9	40.6	24.4	26.1	24.3	13.8	22.0
BEVDetOcc	R50	29.2	17.5	14.1	18.1	2.1	25.5	29.5	11.6	9.5	7.0	4.4	7.3	20.1	40.4	21.3	26.3	23.8	11.5	21.6
ALOcc	R50	34.7	24.0	21.7	**23.8**	17.4	28.0	32.9	17.0	20.2	17.2	**16.9**	**15.4**	**25.5**	41.8	26.7	28.3	27.0	18.6	27.0
RayLift (Ours)	R50	**35.5**	**24.3**	**22.0**	**23.8**	**17.9**	28.3	**33.1**	**17.4**	**21.1**	**17.3**	16.5	**15.4**	25.4	**42.7**	**26.9**	**29.1**	**27.6**	**19.1**	**27.6**

Table 7: **Semantic occupancy results on Occ3D-Waymo.** Following CVT-Occ (Ye et al., 2024) and ALOcc (Chen et al., 2025a), mIoU is computed over the 14 reported semantic classes, excluding motorcycle. RayLift uses camera-only input, a ResNet-50 backbone, and 960×640 input resolution (width $\times$ height).

Method	mIoU(%)	Gen. Obj.	Vehicle	Pedestrian	Sign	Cyclist	T. Light	Pole	Cons. Cone	Bicycle	Building	Vegetation	Tree Trunk	Road	Walkable
BEVFormer-w/o TSA	23.87	7.50	34.54	21.07	9.69	20.96	11.48	11.48	14.06	14.51	23.14	21.82	8.57	78.45	56.89
BEVFormer	24.58	7.18	36.06	21.00	9.76	20.23	12.61	14.52	14.70	16.06	23.98	22.50	9.39	79.11	57.04
SOLOFusion	24.73	4.97	32.45	18.28	10.33	17.14	8.07	17.83	16.23	19.30	31.49	28.98	16.93	70.95	53.28
BEVFormer-WrapConcat	25.07	6.20	36.17	20.95	9.56	20.58	12.82	16.24	14.31	16.78	25.14	23.56	12.81	79.04	56.83
CVT-Occ (Ye et al., 2024)	27.37	7.44	41.00	23.93	11.92	20.81	12.07	18.03	16.88	21.37	29.40	27.42	14.67	79.12	59.09
ALOcc (Chen et al., 2025a)	30.03	6.51	39.61	24.14	20.84	20.56	20.56	24.28	17.95	12.22	35.67	37.25	22.45	78.42	59.91
RayLift (Ours)	**36.55**	**10.00**	**47.20**	**29.73**	**24.95**	**27.02**	**21.66**	**27.72**	**28.30**	**24.94**	**43.87**	**44.57**	**26.72**	**84.31**	**70.67**

Table 7 reports semantic mIoU and per-class IoU over the 14 reported classes, excluding motorcycle. RayLift achieves 36.6 mIoU, compared with 30.0 for ALOcc (Chen et al., 2025a), an improvement of 6.5 points.

E.4 Empirical Per-Ray Rank Analysis

We examine whether the trained RayLift model uses the additional per-ray representational capacity enabled by MFL. From a fixed subset of 20 Occ3D-nuScenes validation samples, we select rays containing at least two occupied depth bins with different ground-truth semantic labels, separated by at least $1\,\mathrm{m}$. This yields 22,074 rays, for which we analyze the lifted $D \times C$ feature matrices before voxel pooling.

Given the singular values $\{s_i\}$ of each matrix, we compute its effective rank as

$$r_{\mathrm{eff}} = \exp\left(-\sum_i p_i \log p_i\right), \qquad p_i = \frac{s_i}{\sum_j s_j}. \tag{E.1}$$

We additionally measure the ratio s_2/s_1 and the cosine similarity between normalized lifted features at occupied depths with different ground-truth semantic labels. The latter measures how similar the representations remain across semantically distinct depths along the same ray.

As predicted by the rank-collapse analysis, scalar lifting produces rank-one feature matrices. In contrast, RayLift achieves an effective rank of 3.52 and a second-to-first singular-value ratio of 0.45. Its lower cross-class cosine similarity (0.68 versus 1.00) further indicates that semantically distinct depths are represented by less collinear features. These findings provide empirical evidence that the trained model uses the additional per-ray capacity enabled by MFL.

Table 8: **Empirical analysis of per-ray lifted features.** Statistics are computed before voxel pooling on GT-selected multi-class rays.

Method	$r_{\text{eff}} \uparrow$	$s_2/s_1 \uparrow$	Cross-class cosine $\downarrow$
Scalar lifting	1.00	0.00	1.00
RayLift	3.52	0.45	0.68

E.5 Qualitative Results

Occ3D-nuScenes. Figure 4 extends the selected-ray comparison in Fig. 3 with three additional examples. In the first car–truck example, ALOcc misclassifies the entire farther truck as the nearer car. In the second example, ALOcc correctly recognizes the truck in its more directly visible portions but still assigns the nearer car's semantics to farther depths along the selected ray. In both cases, RayLift correctly recovers the farther truck semantics along the selected rays. The third example illustrates a different source of ambiguity: although the motorcycle is visible in the input image, coarse voxelization causes the nearer car to obscure it in the occupancy rendering. ALOcc again predicts the car class at the farther depth, whereas RayLift correctly identifies the motorcycle. These examples illustrate semantic confusion arising from both physical occlusion and voxelization-induced overlap along the selected rays.

SurroundOcc. Figure 5 presents four scenes with front-facing RGB images and paired ground-truth and RayLift occupancy renderings. The examples illustrate reconstruction of road geometry, vehicles, vegetation, and surrounding structures under different scene layouts. RayLift captures the dominant scene structure, although discrepancies remain in some distant and fragmented regions. Both occupancy panels use the same front-aligned elevated ego viewpoint, which facilitates comparison with the RGB images without implying exact pixel-level alignment.

Occ3D-Waymo. Figure 6 shows additional results across roadwork scenes, urban intersections, and nighttime driving. RayLift reconstructs the principal road layout and surrounding semantic structure under these varied conditions. Ground truth and predictions are restricted to the same intersection of the official camera-visibility and field-of-view masks and rendered from an identical elevated ego viewpoint. These examples complement the quantitative cross-benchmark evaluation in Table 7.

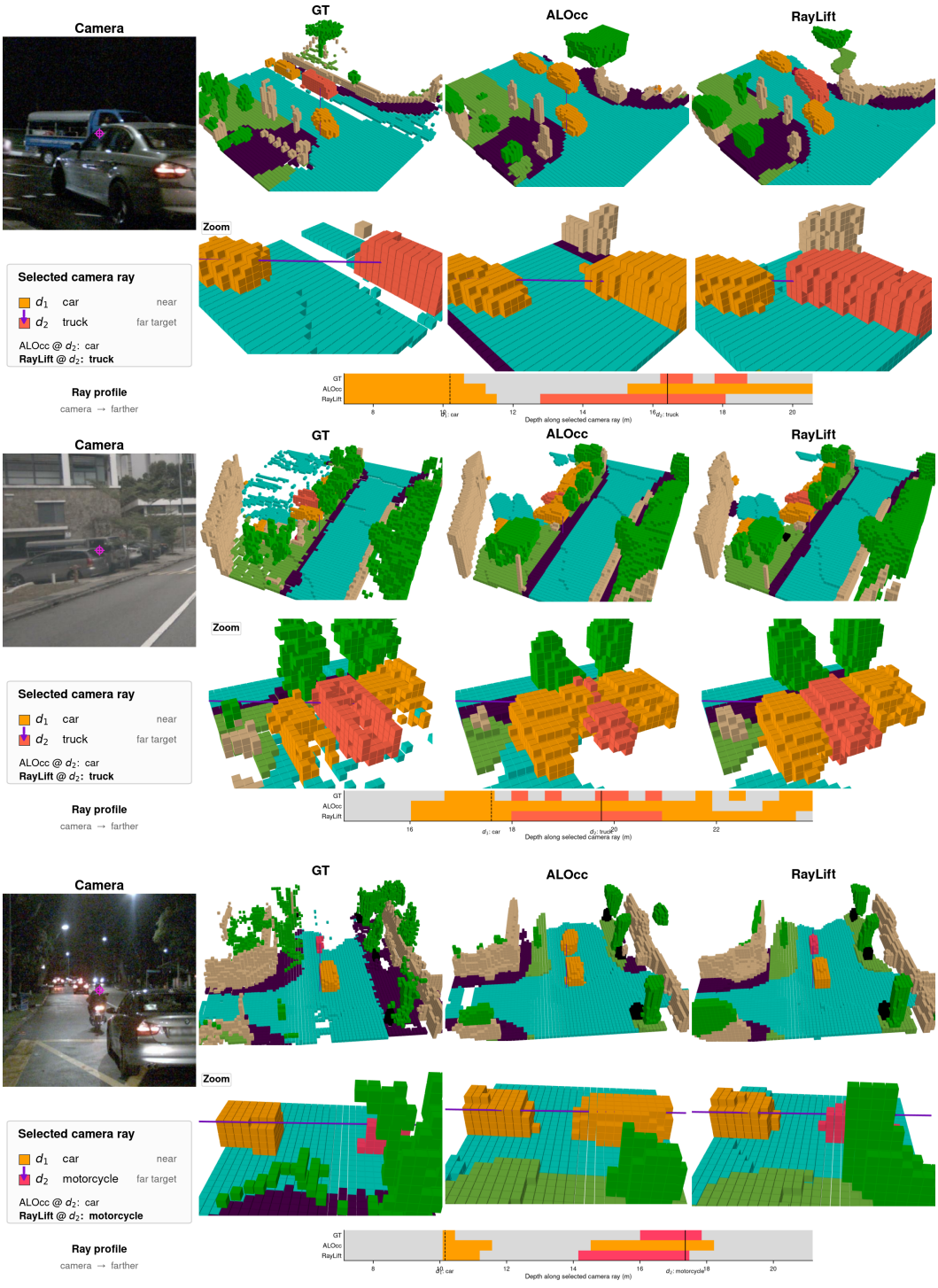

Figure 4: **Additional qualitative comparisons on Occ3D-nuScenes.** Each row shows one selected multi-surface ray example. From left to right, we show the input camera image with the selected ray, ground-truth occupancy, ALOcc prediction, and RayLift prediction. The zoomed views and ray profiles highlight the semantic labels along the selected ray. Across these examples, ALOcc tends to repeat the nearer semantics at farther depths, whereas RayLift better recovers the farther semantic target.

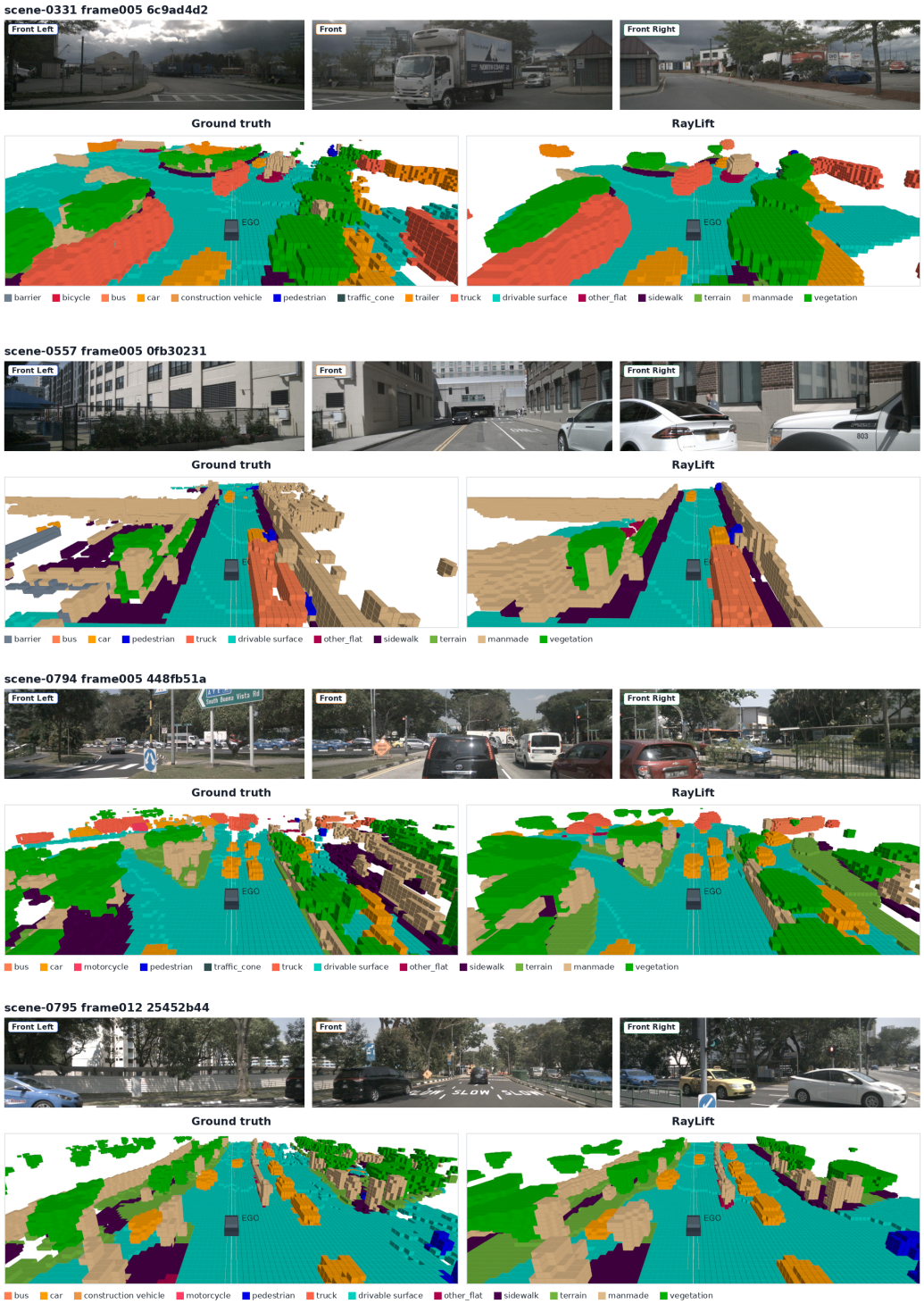

Figure 5: **Qualitative results on SurroundOcc.** Each example shows the front-left, front, and front-right camera images alongside ground-truth and RayLift semantic occupancy. Both voxel renderings use the same front-aligned elevated ego viewpoint, camera framing, and class color palette. The rendered ego vehicle is a schematic marker at the LiDAR origin; the elevated view facilitates visual comparison with the RGB images rather than exact pixel alignment.

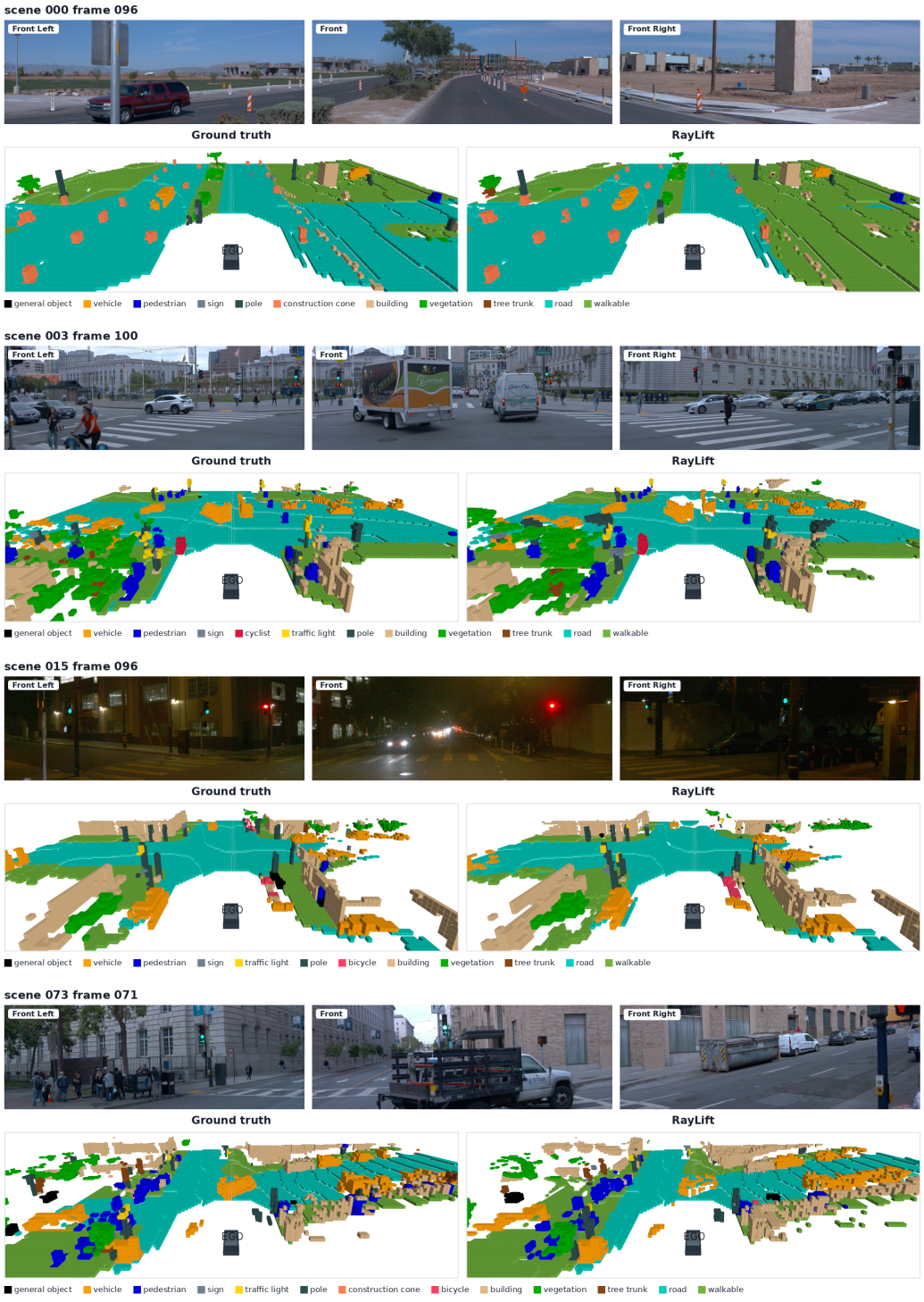

Figure 6: **Qualitative results on Occ3D-Waymo.** The front-left, front, and front-right RGB images (top) provide scene context. Ground-truth (bottom left) and RayLift-predicted (bottom right) semantic occupancy are rendered with the same front-aligned elevated ego viewpoint and semantic color palette. Both occupancy panels are restricted to the same official camera-visible region for visual comparison. The ego vehicle is a schematic orientation marker; the elevated rendering is not an exact reprojection of the RGB images.