OneOcc: Semantic Occupancy Prediction for Legged Robots with a Single Panoramic Camera

(Supplementary Material)

Hao Shi[1,2,*] Ze Wang[1,3,*] Shangwei Guo[1,*] Mengfei Duan[4] Song Wang[1] Teng Chen[5]
Kailun Yang[4,†] Lin Wang[2,†] Kaiwei Wang[1,†]
[1]ZJU [2]NTU [3]MirrorMe Technology [4]HNU [5]Xiaomi Corporation

Contents

*Equal contribution.
†Corresponding authors (e-mail: kailun.yang@hnu.edu.cn, linwang@ntu.edu.sg, wangkaiwei@zju.edu.cn).

1. More Ablations

Setup and notation. Unless stated otherwise, we evaluate on H3O-Heter with the native equirectangular view and report P/R/IoU/mIoU on non-empty voxels. We additionally include QuadOcc only where the raw annulus is required (DP-ER). Grid bounds and taxonomy follow the main paper. For H3O we study both native $64\times64\times8$ and $128\times128\times16$ resolutions. Timing is measured with the batch size 1 on a single NVIDIA RTX-4090 GPU (FP32). We report Params (M), FPS, and Mem (GB).

1.1. Hierarchical AMoE-3D: Number of Experts K

> *Question.* How does the number of 3D experts affect accuracy?

Table 1. A1: AMoE-3D experts on H3O-Heter. $K=1$ reduces to a fused bottleneck. Gate is 3D Gradient-Energy by default unless noted. We report precision (P), recall (R), IoU, and mIoU on non-empty voxels under the official H3O-Heter protocol.

K	Gate	P	R	IoU	mIoU
1	–	65.38	61.02	46.12	29.68
2	GradEnergy3D	66.79	63.69	48.37	31.25
4	GradEnergy3D	67.89	**64.77**	**49.58**	**32.23**
8	GradEnergy3D	**68.08**	64.39	49.46	32.03
4	Uniform	66.21	63.04	47.70	31.13
4	Top-k	67.18	64.06	48.79	31.84

Findings. Increasing K from 1 to 4 consistently improves robustness on H3O-Heter. Moving to $K=8$ slightly increases precision but reduces recall, leading to marginal declines in IoU/mIoU due to expert fragmentation and noisier routing. 3D Gradient-Energy gating (Sec. 3.6) prioritizes high-gradient regions (class boundaries and thin structures), yielding better voxel-wise expert assignment than uniform or naive Top-k routing [1]. $K=4$ achieves the best accuracy–efficiency trade-off and is our default. We also note related progress in point-cloud domain adaptation: Point-MoDE [2] employs a mixture-of-domain-experts to enhance cross-domain generalization, reaching conclusions consistent with ours.

1.2. Horizontal Field-of-View: from 90° to 360°

> *Question.* How much does full 360° surround help, and how do methods degrade when evaluated under narrower FoVs?

Protocol (train-once, test-with-crops). Unless otherwise stated, all methods are trained with full 360° ER panoramas. At evaluation time, we reduce the horizontal FoV by cropping the panorama *without finetuning* the models. We use forward-centered cropping so the input spatial extent truly shrinks as the FoV narrows (thereby reflecting realistic compute/memory savings).

Table 2. A2: FoV ablation on H3O-Heter (ER panorama; horizontal crop at eval only; no retraining). We report overall metrics on non-empty voxels. Both OneOcc and MonoScene [3] are *trained once at* 360°, and *evaluated* at narrower FoVs via forward-centered cropping.

FoV	Visible	Method	#Params	FPS ↑	Mem ↓	P ↑	R ↑	IoU ↑	mIoU ↑
90°	25%	MonoScene [3]	146.13M	13.07	1.72GB	56.14	13.59	12.28	4.65
		OneOcc (ours)	101.76M	24.19	1.65GB	58.65	15.43	13.92	7.35
180°	50%	MonoScene [3]	146.13M	12.63	1.78GB	63.40	29.64	25.31	13.74
		OneOcc (ours)	101.76M	19.24	1.70GB	62.68	31.62	26.61	17.27
270°	75%	MonoScene [3]	146.13M	10.08	2.09GB	65.46	41.19	33.84	18.20
		OneOcc (ours)	101.76M	16.13	1.76GB	65.79	47.53	38.11	24.28
360°	100%	MonoScene [3]	146.13M	8.29	2.60GB	67.39	55.00	43.44	24.15
		OneOcc (ours)	101.76M	14.30	1.82GB	67.89	64.77	49.58	32.23

Findings.

- **Train-once, test-with-crops.** For OneOcc, mIoU scales near-monotonically with FoV: $+9.92$ ($90\to180$), $+7.01$ ($180\to270$), and $+7.95$ ($270\to360$), totaling $+24.88$ from 90° to 360° ($+77.2\%$ rel.). MonoScene also improves but less ($+19.50$ total). This indicates that *surround cues translate into long-range context*, with stronger returns when the azimuthal ring is closed.
- **Closing-the-ring effect.** The last increment ($270\to360$) is sizable for both methods, and larger for OneOcc ($+7.95$ vs. $+5.95$ mIoU), evidencing a *nonlinear benefit* when the 360° ring continuity becomes complete.
- **Precision parity, Recall advantage.** OneOcc and MonoScene have similar precision at a given FoV (*e.g.*, 67.89 vs. 67.39 at 360°), while the *recall gap* dominates OneOcc's advantage: $+9.77$ recall points at 360° (and $+6.34$ at 270°). This matches our design goal: cylindrical alignment and dual-projection preserve azimuthal continuity, yielding *more complete* occupancy recovery as FoV expands.
- **Absolute margins grow with FoV.** OneOcc surpasses MonoScene by $+2.70$ / $+3.53$ / $+6.08$ / $+8.08$ mIoU at $\{90^\circ, 180^\circ, 270^\circ, 360^\circ\}$, respectively. Relative gains are $+58.1\%$, $+25.7\%$, $+33.4\%$, $+33.5\%$.
- **Accuracy–throughput Pareto (AET).** We define AET $=$ mIoU $\times$ FPS to summarize accuracy at a given runtime budget. OneOcc vs. MonoScene AET is:

$$\begin{aligned} 90^\circ &: \ 177.8 \text{ vs } 60.8 \ (\mathbf{2.93}\times) \\ 180^\circ &: \ 332.3 \text{ vs } 173.5 \ (\mathbf{1.91}\times) \\ 270^\circ &: \ 391.6 \text{ vs } 183.5 \ (\mathbf{2.13}\times) \\ 360^\circ &: \ 460.9 \text{ vs } 200.2 \ (\mathbf{2.30}\times) \end{aligned}$$

Across all FoVs, OneOcc is *Pareto-superior* in accuracy–throughput.

- **Memory/parameter efficiency.** OneOcc uses 30.3% fewer parameters (101.76M vs. 146.13M). Its mIoU per GB is consistently higher: 4.45 vs. 2.70 (90°), 10.16 vs. 7.72 (180°), 13.80 vs. 8.71 (270°), 17.71 vs. 9.29 (360°), *i.e.*, 1.31–1.91× memory-normalized gains.
- **Practical iso-accuracy choices.** If targeting ∼24 mIoU, *OneOcc@270°* achieves 24.28 mIoU at 16.13 FPS and 1.76 GB, matching/exceeding *MonoScene@360°* (24.15 mIoU at 8.29 FPS, 2.60 GB). Thus, for similar accuracy, OneOcc needs ∼**2**× throughput and −0.84 GB memory. Likewise, *OneOcc@180°* (17.27 mIoU, 19.24 FPS) rivals *MonoScene@270°* (18.20 mIoU, 10.08 FPS) at roughly ∼**1.9**× throughput.

Takeaways. Full-surround cues are critical; methods that explicitly encode ring continuity not only achieve higher absolute accuracy, but also deliver better *accuracy-per-compute* and *accuracy-per-memory*. In embodied settings, these advantages persist even when FoV must shrink to meet runtime constraints.

Why an ego-centric 360° symmetric occupancy grid? Unlike automotive datasets (*e.g.*, front-view grids such as SemanticKITTI [4]) that emphasize a forward driving cone, embodied agents must reason about objects and free space *all around* the ego: turning-in-place, backtracking, and manipulations behind/aside the agent are common. A *symmetrically* centered 360° grid (front/back/left/right balanced) aligns the spatial prior with such behaviors, reduces coordinate bias, and enables globally consistent occlusion reasoning and memory beyond the forward frustum. Empirically, this grid pairs naturally with ring-continuous features and improves cross-directional consistency in long-range context.

1.3. Bi-Grid Voxelization: Cartesian vs. Cylinder

> *Question.* Does the camera-aligned cylindrical grid help beyond a standard Cartesian grid?

Table 3. A3: Single-grid vs. Bi-Grid on H3O-Heter. We stratify *near* using an XY-centered crop (center ratio 0.5, *i.e.*, front/back/left/right ±6.4 m) and *far* using the full XY footprint (±12.8 m); the vertical range is fixed to $z \in [-2, 1.2]$ m.

Voxelization	#Params	FPS ↑	Mem ↓	P ↑	R ↑	IoU ↑	Near mIoU ↑	Far mIoU ↑
Cartesian-only	101.73M	14.32	1.70GB	62.54	63.18	45.84	36.42	30.56
Cylindrical-only	101.73M	14.33	1.78GB	63.65	**65.30**	47.57	34.15	31.00
Bi-Grid	101.76M	14.30	1.82GB	**67.89**	64.77	**49.58**	**37.35**	**32.23**

Setup. Unless otherwise stated, we predict occupancy on a Cartesian grid and evaluate within a fixed vertical range $z \in [-2, 1.2]$ m. For range-wise analysis, we follow a panoramic-camera protocol: *near* uses an XY-centered crop with a center ratio of 0.5 (*i.e.*, front/back/left/right ±6.4 m around the ego) and excludes far voxels; *far* uses the full XY footprint (front/back/left/right ±12.8 m). The far split is intrinsically harder due to larger geometric errors, heavier occlusions, sparser voxels, and higher sampling noise, so many classes naturally obtain higher Recall/IoU in the near window, which in turn boosts the near mIoU.

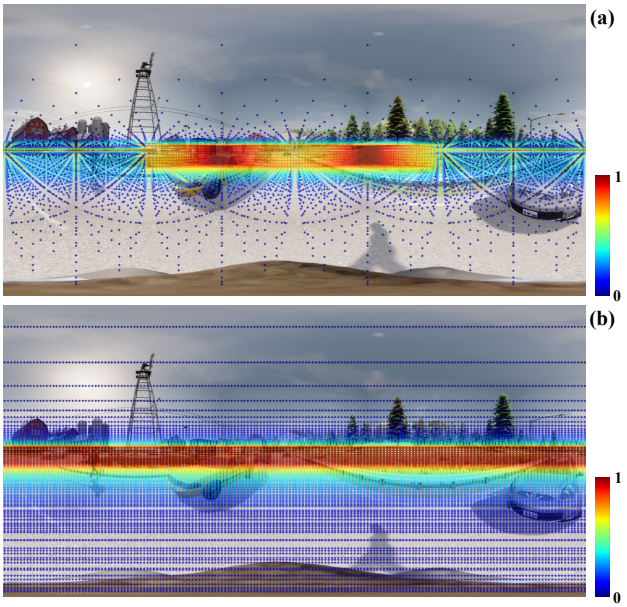

Figure 1. **Cartesian vs. cylindrical voxel projections on a panoramic view.** We visualize how different voxel parameterizations project onto an equirectangular panorama in Human360Occ. Dots denote voxel centroids projected to the image; the colored band (0–1) encodes the normalized depth distribution of occupied voxels along each azimuthal ray. (a) A conventional axis-aligned *Cartesian* grid in world coordinates produces fan-shaped footprints when projected to the panorama: voxels at different heights but similar depth map to tilted rays, and far-range structures such as roads, snow banks, and building façades are squeezed into a narrow band around the equator. This favors metrically uniform sampling in 3D but breaks the angular regularity of the panoramic image. (b) In contrast, an omnidirectional camera-centric *cylindrical* grid uses equal steps in azimuth and height, so the projected voxel centroids form nearly horizontal sampling rows that follow the equirectangular parameterization and maintain ring continuity for distant structures. This complementarity motivates combining both grids in our Bi-Grid voxelization to balance near-field metric fidelity and far-field azimuthal coherence.

Findings.

- **Complementary inductive biases.** A cylindrical grid (aligned with panoramic camera rays and azimuth) preserves ring continuity and equal-angle sampling, favoring *far-field* layout regularities (roads/sidewalks bands, building facades). Compared with Cartesian-only, Cylindrical-only raises Recall by +2.12 (65.30 vs. 63.18) and IoU

by +1.73, but degrades *near* mIoU by -2.27 (34.15 vs. 36.42), while slightly improving *far* mIoU by +0.44 (31.00 vs. 30.56), echoing Cylinder3D-style observations on cylindrical parameterizations for long-range structure [5].

- **Bi-Grid synergy with negligible overhead.** Our Bi-Grid fuses both discretizations and delivers the best of both worlds: over Cartesian-only it gains +5.35 Precision (67.89 vs. 62.54), +1.59 Recall, and +3.74 IoU, while consistently lifting *both* ranges (near: +0.93, far: +1.67). Against Cylindrical-only, Bi-Grid trades a small Recall drop (-0.53) for a much larger Precision gain (+4.24), resulting in the highest IoU/mIoU overall. The parameter and runtime overhead are minimal (+0.03 M params; $\sim$14.3 FPS maintained; +0.12 GB peak memory).
- **Why the overhead is tiny (implementation).** The voxel centroids for both cartesian and cylindrical grids are precomputed and projected to the images in the `dataloader`; at inference we only bilinearly sample 2D features at these fixed locations. We remove the offset-MLP that predicts per-voxel 2D sampling displacements, so the parameter budget barely changes (observed $\Delta \approx +0.03$ M) and throughput remains $\sim$14.3 FPS. The main extra cost is memory: Bi-Grid performs two sets of feature samplings and maintains slightly larger activation buffers, causing a modest peak-memory increase (about +0.12 GB).
- **Why Cartesian still matters near the ego.** Evaluation and downstream planners operate on a Euclidean (Cartesian) grid; close-range contact geometry (ground, curbs, thin structures, small objects) benefits from uniform metric spacing and axis-aligned neighborhoods. This explains why Cartesian-only is stronger than Cylindrical-only on *near* mIoU (36.42 vs. 34.15).
- **Panoramic geometry alignment.** With equirectangular/-panoramic imaging, cylindrical bins match the camera's spherical parameterization and mitigate azimuthal aliasing, improving *far*-field coherence; Cartesian bins better preserve metric fidelity and local topology critical for *near*-field planning. Bi-Grid inherits both advantages.
- **Context from SSC/OCC.** Prior occupancy works (*e.g.*, VoxFormer [6], OccFiner [7]) typically report higher short-range fidelity due to denser observations and stronger priors, while accuracy decays with distance. Our *near/far* split makes this explicit: the near window naturally boosts Recall/IoU for many classes, and cylindrical alignment helps counteract the far-range drop; Bi-Grid couples the two.

Takeaways. (i) Cylindrical improves far-field ring coherence but may underfit metric-accurate contact geometry nearby; (ii) Cartesian preserves near-field fidelity but underexploits azimuthal continuity; (iii) For panoramic input with Cartesian output, Bi-Grid is a near-free, robust default that simultaneously improves both *near* and *far* regimes.

1.4. Dual-Projection: ER-only vs. Raw-only

> *Question.* How much does processing both the equirectangular (ER) panorama *and* the raw annulus help?

Setup. All variants share the same backbone/decoder and training schedule; only the input projection path differs. For *ER-only* we unwrap the PAL annulus to equirectangular and feed a single ER encoder. For *Raw-only* we process the native annulus with a single raw-path encoder. *Dual (ER+Raw)* instantiates both encoders and fuses features before 3D lifting. Unless otherwise stated, inference is measured with batch size 1, FP32; ER-based paths use 352×1216, and Raw-only uses 512×512. We report wall-clock throughput (FPS) and peak CUDA memory alongside accuracy (Precision P, Recall R, IoU, mIoU). On H3O (native ER panoramas) we adopt ER-only by default; on QuadOcc (true PAL optics) we compare all three paths.

Table 4. A4: Projection path ablation. On H3O we use ER-only by default (native ER); on QuadOcc we compare ER-only, Raw-only, and Dual (ER+Raw).

Projection Path	#Params	FPS ↑	Mem ↓	P ↑	R ↑	IoU ↑	mIoU ↑
ER-only	101.76M	18.15	1.71GB	65.77	**64.81**	48.47	20.03
Raw-only	101.97M	25.18	1.55GB	66.42	62.86	47.70	19.76
Dual (ER+Raw)	189.83M	15.46	2.14GB	**66.69**	64.74	**48.92**	**20.56**

Findings.

- **Accuracy on QuadOcc.** Dual achieves the best overall occupancy quality, improving over ER-only by +0.45 IoU and +0.53 mIoU (48.92/20.56 vs. 48.47/20.03), with higher precision (66.69 vs. 65.77) at essentially unchanged recall (64.74 vs. 64.81). Raw-only is weaker on IoU/mIoU (47.70/19.76), indicating that using the raw annulus *alone* is insufficient for best voxel quality.
- **Cost/efficiency.** Relative to ER-only, Dual increases parameters by +86.5% (189.83M vs. 101.76M) and peak memory by +25.1% (2.14 GB vs. 1.71 GB), while throughput drops by 14.8% (15.46 vs. 18.15 FPS). Raw-only is the fastest (25.18 FPS; +38.7% vs. ER-only) and most memory-frugal (1.55 GB), but sacrifices recall (-1.95) and IoU (-0.77).
- **H3O (native ER).** When inputs are already native ER panoramas, the *incremental* benefit of adding a raw path is marginal relative to its compute/memory overhead; ER-only is the practical setting.

Why Dual helps on QuadOcc (true PAL). A PAL camera forms a circular annulus on the sensor. Unwrapping to ER is convolution-friendly and keeps azimuthal continuity, but

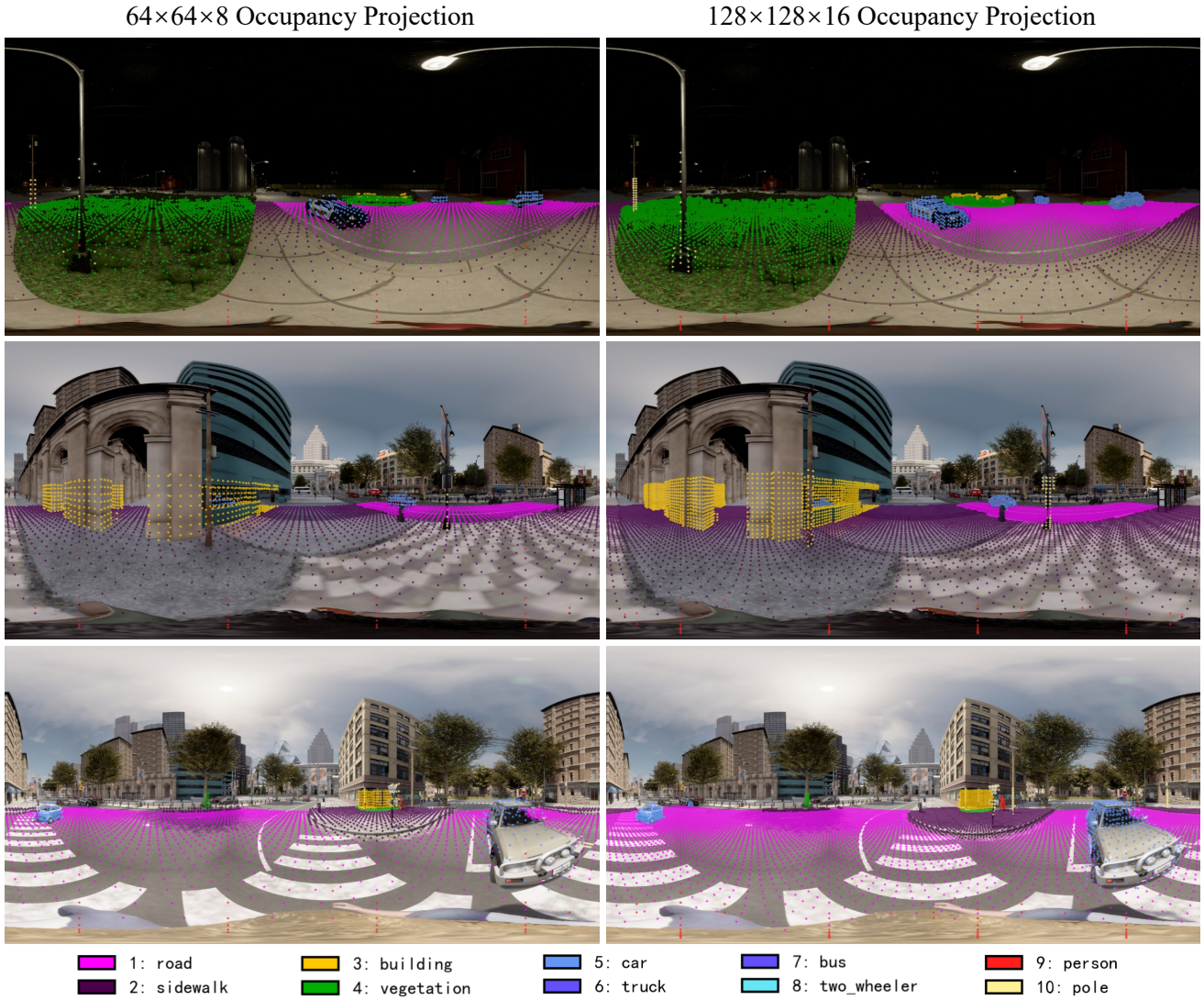

Figure 2. **Semantic ground-truth occupancy projections at different voxel resolutions.** We visualize the projection of *ground-truth* semantic voxels from grids of size $64 \times 64 \times 8$ (left) and $128 \times 128 \times 16$ (right) onto the panoramic image. Colored dots follow the legend at the bottom and indicate voxel centers of different semantic classes relevant to legged locomotion, such as road, sidewalk, buildings, vegetation, vehicles, pedestrians, and poles. The higher-resolution $128 \times 128 \times 16$ grid produces visually denser and more detailed occupancy patterns, with sharper boundaries and more finely sampled thin structures. However, the coarser $64 \times 64 \times 8$ grid still provides sufficient spatial coverage and granularity for a legged robot to perceive traversable surfaces and nearby obstacles in 360° around the body. Considering the computation and payload constraints of embodied agents, we adopt $64 \times 64 \times 8$ as the default resolution for main results on both datasets.

induces latitude-dependent distortion and can smooth out high-frequency structures near the inner/outer rings. The *raw annulus* stream preserves native PAL geometry and local texture statistics; fusing it with the ER stream lets the network resolve ER-induced distortions and seam/edge artifacts while retaining the global continuity provided by ER. This complementary coverage improves precision (fewer false positives around thin/elongated objects) without hurting recall, thereby nudging IoU/mIoU upward.

Takeaway. For true PAL deployments (QuadOcc-like), Dual (ER+Raw) is a robust choice when accuracy is prioritized over a modest efficiency loss. For native ER inputs (H3O-like), ER-only offers the best accuracy–efficiency trade-off; enable Dual only when the last bit of precision is critical.

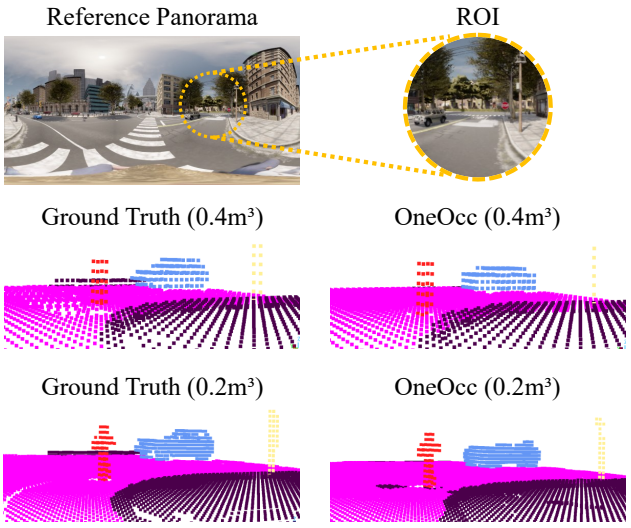

Figure 3. **Voxel grid resolution vs. throughput/accuracy on *H3O-Heter***. We keep identical 3D bounds and voxel size; only the grid resolution changes: *BEV* increases from 64×64 to 128×128 while the number of vertical bins increases from 8 to 16. Timings are measured on a single RTX 4090 with batch size 1 (FP32). Raising resolution adds +3.41 M parameters (101.76→105.17 M), reduces FPS (14.30→5.01), and inflates peak memory (1.82→10.71 GB), with a concomitant drop in IoU/mIoU (49.58/32.23→29.98/21.16).

1.5. Voxel Resolution Scaling

> *Question.* How does performance scale when increasing the voxel grid from $64\times64\times8$ to $128\times128\times16$ under identical spatial bounds and training/inference settings?

Setup. We isolate the effect of resolution by fixing the spatial range R, evaluation protocol (*H3O-Heter*), and all model/hyperparameters; only the voxel grid resolution is varied. We report precision (P), recall (R), IoU, and mIoU on non-empty voxels, together with #Params/FPS/Memory measured on a single RTX 4090 (FP32, batch size 1).

Table 5. A5: Resolution scaling on H3O-Heter. Same spatial bounds; only the grid resolution changes.

Resolution	#Params	FPS ↑	Mem ↓	P ↑	R ↑	IoU ↑	mIoU ↑
64×64×8	101.76M	14.30	1.82GB	67.89	64.77	49.58	32.23
128×128×16	105.17M	5.01	10.71GB	44.69	47.66	29.98	21.16

Findings.

- **Throughput & footprint.** Increasing BEV resolution by $2\times$ in X/Y and doubling vertical bins ($8\rightarrow16$) leaves the backbone almost unchanged in parameter count ($+3.41M$) but cuts throughput by $\sim3\times$ and raises memory by $\sim5.9\times$, as denser 3D feature maps amplify intermediate activations.
- **Accuracy at higher resolution.** Finer-grained occupancy is intrinsically harder: evaluating and *learning* at a denser discretization exacerbates class imbalance and boundary sensitivity, and demands longer-range completion for thin, rare structures. Empirically, our $128\times128\times16$ setting yields lower aggregate IoU/mIoU than $64\times64\times8$ despite the finer grid. This trend aligns with multiscale SSC literature (*e.g.*, LMSCNet [8]), where coarse-scale heads are designed for efficiency and are consistently easier to optimize/infer than their fine-scale counterparts[1].
- **Embodied perspective.** Our target use case is *embodied intelligence* with tight on-board compute and memory budgets. Given the $\sim3\times$ FPS gain and $\sim6\times$ lower memory at $64\times64\times8$, we adopt this resolution as the *main* setting in the paper: it strikes the best accuracy–efficiency trade-off for real-time, resource-constrained platforms, while $128\times128\times16$ is reserved for analyses prioritizing geometric detail with a substantial compute penalty.

Therefore, under fixed bounds, $64\times64\times8$ is the preferred default for accuracy–efficiency in embodied settings; $128\times128\times16$ offers crisper geometry on slender structures but at markedly worse throughput/memory and lower overall IoU/mIoU.

1.6. Calibration Robustness

> *Question.* How robust is OneOcc to test-time intrinsic/extrinsic calibration perturbations, and can simple calibration-noise augmentation flatten the degradation curve?

Setup. We evaluate robustness on *H3O-Heter* by injecting controlled calibration noise at *data loading time* in the equirectangular projection. Intrinsic perturbation is implemented as a global pixel scaling, while extrinsic perturbation is simulated as per-point pixel offsets. For reproducibility, the perturbation is deterministic for each frame via a fixed seed and frame id, with clipping and FoV-mask update applied accordingly. We report mIoU under three settings: joint intrinsic+extrinsic noise, intrinsic-only noise, and extrinsic-only noise, each with perturbation ratios $p \in \{1\%, 2\%, 5\%\}$. We additionally include robustness-oriented variants retrained with 5% joint calibration-noise augmentation.

Findings.

- **Robustness under calibration perturbation.** Under all corruption levels, OneOcc consistently remains above MonoScene. With joint intrinsic+extrinsic noise, OneOcc achieves 27.26/23.67/16.75 mIoU at 1%/2%/5%, outperforming MonoScene by +5.27/+4.09/+2.79, respectively.

[1] See LMSCNet's multiscale completion design and coarser-head efficiency.

Table 6. A6: Robustness to intrinsic/extrinsic calibration noise on H3O-Heter. We report mIoU under clean and noisy projections.

Method	Clean	Both (Intr.+Extr.)			Intr. only			Extr. only		
	0%	1%	2%	5%	1%	2%	5%	1%	2%	5%
MonoScene	24.15	21.99	19.58	13.96	22.78	21.00	16.30	22.75	20.69	15.90
OneOcc	32.23	27.26	23.67	16.75	27.88	24.80	19.15	29.42	25.28	18.95
MonoScene†	18.97	18.86	18.64	16.48	18.71	18.28	15.74	19.01	19.17	18.92
OneOcc†	23.00	22.97	23.03	22.84	22.86	22.84	22.48	22.94	23.15	23.16

† *Retrained with 5% joint intrinsic+extrinsic calibration-noise augmentation.*

This indicates that the proposed dual-projection lifting and 3D reasoning pipeline retains stronger geometric consistency under imperfect calibration.

- **Intrinsic vs. extrinsic noise.** Both perturbation types degrade performance, while their combination is the most harmful as expected. Intrinsic-only noise is slightly more disruptive at low-to-moderate levels, likely because global projection-scale distortion affects all lifted features simultaneously, whereas extrinsic perturbation mainly introduces spatial misalignment. Nevertheless, OneOcc preserves a clear margin over the baseline in all settings.
- **Effect of calibration-noise augmentation.** Retraining with simple 5% calibration-noise augmentation substantially flattens the degradation curve. For example, the robustness-oriented OneOcc† variant stays nearly constant at 22.97/23.03/22.84 mIoU under 1%/2%/5% joint noise. This comes at the cost of lower clean performance (23.00 vs. 32.23), suggesting a standard robustness–accuracy trade-off: the default model is preferable when calibration is reliable, whereas noise augmentation is attractive for long-term deployment under persistent drift.

Therefore, OneOcc is reasonably robust to bounded calibration errors and consistently more resilient than MonoScene; when stronger calibration drift is expected, simple calibration-noise augmentation can further improve robustness at the expense of clean-set accuracy.

1.7. Temporal Aggregation

> *Question.* Does lightweight temporal aggregation materially improve occupancy prediction, and how does it trade accuracy against latency and memory compared with the default single-frame design?

Setup. We compare the default *single-frame* OneOcc with two simple 3-frame temporal variants under identical image resolution (608×1216), voxel bounds, backbone, and evaluation protocol. The first variant performs temporal feature averaging using *ground-truth pose* for alignment, representing a minimal upper-bound baseline with reliable multi-frame registration. The second variant adopts a lightweight BEVFormer-style temporal attention over three frames. We report mIoU on *QuadOcc-val* and *H3O-Heter*, together with latency and memory measured on a single RTX 4090 (FP32, batch size 1).

Table 7. A7: Temporal aggregation vs. the default single-frame design. Temporal fusion improves accuracy, but also increases deployment cost.

Setting	QuadOcc-val mIoU ↑	H3O-Heter mIoU ↑	Latency (ms) ↓	Memory (GB) ↓
OneOcc (single-frame)	20.56	32.23	69.93	1.82
+ Temporal average by GT pose (3 frames)	20.92	33.74	69.93	1.82
+ Temporal BEVFormer-like attention (3 frames)	21.18	34.25	78.60	2.35

Findings.

- **Temporal aggregation is beneficial with reliable alignment.** Both 3-frame variants improve over the default single-frame setting. Simple temporal averaging already raises mIoU from 20.56 to 20.92 on QuadOcc-val and from 32.23 to 33.74 on H3O-Heter, confirming that multi-frame context helps suppress transient gait-induced jitter and recover partially occluded structures when alignment is accurate.
- **Stronger temporal fusion yields larger gains but higher cost.** The BEVFormer-style temporal attention further improves mIoU to 21.18 on QuadOcc-val and 34.25 on H3O-Heter, but increases latency from 69.93 ms to 78.60 ms and memory from 1.82 GB to 2.35 GB. This shows that temporal fusion is effective, yet its benefit is not free: better multi-frame reasoning comes with additional state handling, feature fusion overhead, and higher deployment cost.
- **Why single-frame remains the default.** Our goal is an *easy-to-deploy*, *low-latency*, and *odometry-free* embodied perception system. The temporal-average variant relies on ground-truth pose for reliable frame alignment and therefore should be interpreted as an upper-bound reference rather than a plug-and-play deployment mode. In contrast, the default single-frame OneOcc avoids temporal drift accumulation, requires no motion history, and maintains a cleaner inference path for resource-constrained onboard platforms.

Therefore, temporal aggregation can further improve panoramic occupancy prediction when accurate multi-frame alignment is available, but the default single-frame OneOcc remains the preferred setting for real-time embodied deployment due to its simplicity, robustness to drift, and lower system complexity.

2. Qualitative Analysis of GDC

Figure 4 provides a qualitative view of GDC on real-world quadruped runs. The three rows correspond to panoramas captured under increasing gait intensity, ranging from mild to pronounced body oscillation. Because the panoramic camera is mounted on a moving body, the gait induces vertical shake during the exposure time and creates a characteristic motion-blur pattern: distant objects and tree lines are

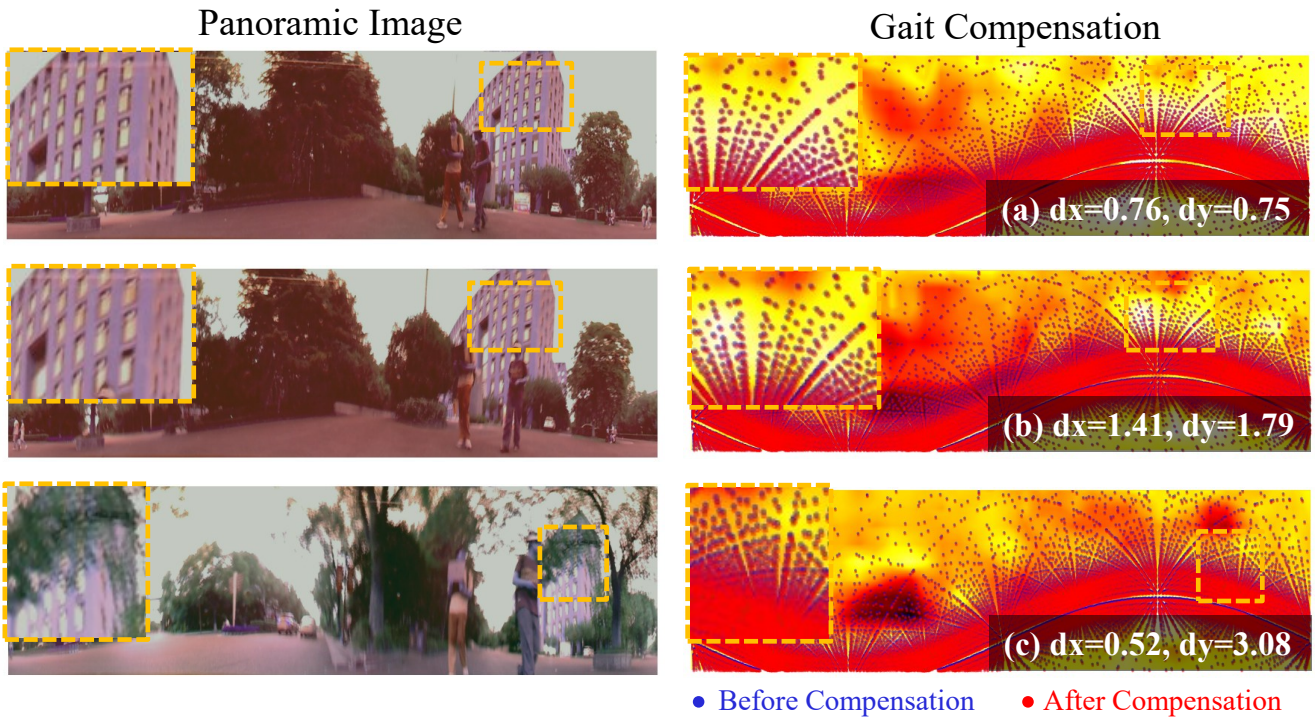

Figure 4. **Panoramic gait displacement compensation on real-world quadruped data.** We visualize the effect of Gait Displacement Compensation (GDC) on our QuadOcc dataset collected by a quadruped robot equipped with a panoramic camera. Each row shows one panorama captured at a different gait intensity: the left column is the input image with a zoom-in of a distant structure, and the right column overlays FLoSP [3] sampling locations on the equirectangular feature map before (blue) and after (red) GDC. From top to bottom, the robot motion changes from mild to pronounced body oscillation, and the estimated vertical offsets dy increase accordingly (see the (dx, dy) values in (a)–(c)), while the horizontal offsets dx remain comparatively small. This indicates that gait-induced jitter produces a characteristic *vertical* motion-blur pattern during exposure, and GDC automatically adapts the strength of its correction mainly along the vertical axis. Although the predicted displacements are real-valued and sometimes fractional, we augmented FLoSP with bilinear interpolation, so sub-pixel shifts of the sampling grid are still meaningful and help recover sharper, less blur-contaminated features for downstream occupancy prediction in this single-frame model.

smeared mainly along the vertical direction. Without compensation, the fixed FLoSP [3] sampling grid (blue) may query different parts of this blurred streak depending on the current gait phase, effectively mixing background and foreground evidence within a single frame. GDC predicts per-frame offsets that correct this effect: from (a) to (c), the estimated dy increases while dx stays small, showing that the module automatically scales its response with the blur magnitude and focuses almost exclusively on the dominant vertical direction. Note that OneOcc operates purely on single panoramas without temporal aggregation, so GDC is not used for temporal alignment; instead, it refines the instantaneous feature sampling locations to better match the underlying static geometry under gait-induced motion blur. Since we implemented FLoSP with bilinear interpolation rather than the classic nearest-neighbor lookup [3], the real-valued (often fractional) (dx, dy) still produce meaningful sub-pixel shifts of the sampling grid, which leads to cleaner voxel features and the consistent mIoU gains observed in our ablations.

3. Lightweight Design Philosophy of OneOcc

While Sec. 4.5 of the main paper reports the overall runtime and memory footprint, here we decompose the parameter counts of OneOcc and the MonoScene [3] baseline into their major components. We focus on the panoramic setting on QuadOcc as well as the Human360Occ (H3O) variant.

3.1. QuadOcc: dual-projection variant

Tab. 8 summarizes the parameter breakdown when training on QuadOcc with the full dual-projection encoders (DP–ER) and bi-grid voxelization. MonoScene [3] consists of a single 2D encoder operating on the equirectangular panorama and a standard 3D UNet decoder, resulting in 132M parameters for the 2D encoder and 16.9M for the 3D decoder (148.9M total).

In contrast, OneOcc factorizes its capacity across (i) two 2D encoders for raw and equirectangular views (DP–ER), (ii) a lightweight depthwise-separable 3D decoder with Hierarchical AMoE-3D, and (iii) a set of geometry-aware aux-

Table 8. Parameter breakdown on QuadOcc (panorama-only). The additional geometry-aware modules in OneOcc (GDC, BGV-informed volumetric fusion, and the lightweight AMoE-3D gating heads) contribute less than 0.2% of the total parameters. The AMoE-3D attention and expert weights are counted as part of the 3D decoder. Both the 3D decoder and each single 2D encoder are lighter than the MonoScene [3] counterpart.

Method	2D encoder(s)	3D decoder	GDC+fusion+gates	Total
MonoScene (QuadOcc)	1×132M	16.9M	–	148.9M
OneOcc (QuadOcc)	2×87.8M	12.3M	≈ 0.27M	189.8M

iliary modules (GDC, volumetric fusion, and tiny gating heads). Each 2D encoder in DP–ER contains 87.8M parameters, and the DWLite3D decoder with AMoE-3D attention blocks (excluding the gating heads) contains 12.3M. The remaining components—GDC heads, 3D SE/fusion layers, and the lightweight AMoE-3D gating MLPs—add only about 0.27M parameters in total (less than 0.2% of the overall capacity). For clarity, we count the AMoE-3D attention and expert weights inside the "3D decoder" bucket, while the gating networks are included in the "GDC+fusion+gates" bucket in Tab. 8 and Tab. 9.

Although the total parameter count increases by $\approx 26\%$ compared to MonoScene (148.9M $\rightarrow$ 189.8M), the design is *structurally lightweight*: (1) the 3D decoder is 27% smaller (16.9M $\rightarrow$ 12.3M) thanks to the depthwise-separable DWLite3D design; (2) each single DP–ER encoder is 33% lighter than MonoScene's 2D encoder (132M $\rightarrow$ 87.8M); and (3) the additional geometry-aware modules (GDC, BGV-based fusion, AMoE-3D gates) are extremely cheap in parameters. The modest increase in total parameters on QuadOcc therefore comes almost entirely from using two lighter encoders in parallel, which is amortized by GPU parallelism and justified by the improved robustness under gait-induced jitter.

3.2. H3O: single-projection variant

On Human360Occ (H3O), the images are already native equirectangular, so we follow the main paper and disable the raw-annulus branch in DP–ER and the associated fusion layers, while keeping the same DWLite3D decoder and GDC. Tab. 9 reports the resulting parameter statistics. MonoScene [3] again uses a single 2D encoder (132M) and the standard 3D decoder (16.9M), totaling 148.9M parameters.

The H3O configuration of OneOcc uses a single 2D encoder (87.8M), the same 12.3M DWLite3D decoder, and the small GDC and gating modules, for a total of $\sim$100M parameters. This corresponds to a reduction of approximately 33% in parameter count compared to MonoScene [3] (148.9M $\rightarrow$ 101.7M), while retaining all the geometry-aware components that are shown in the main paper ablations to be crucial for far-field context and near-field contact geometry.

Table 9. Parameter breakdown on H3O. When the raw panoramic branch is disabled, OneOcc becomes markedly lighter than MonoScene [3] in terms of learned parameters, while preserving GDC, the lightweight AMoE-3D gating heads, and the DWLite3D attention blocks (whose weights are counted inside the 3D decoder).

Method	2D encoder(s)	3D decoder	GDC+gates	Total
MonoScene (H3O)	1×132M	16.9M	–	148.9M
OneOcc (H3O)	1×87.8M	12.3M	≈ 0.27M	101.7M

Table 10. Measured runtime breakdown on NVIDIA Jetson AGX Orin 64GB (MAX power mode). Input: $1\times3\times370\times1220$ (QuadOcc panoramas), batch size 1; 5 warm-up iterations + 100 measured runs. The 2D image encoder/decoder is executed in INT8 (TensorRT), while the remaining modules are run in FP16. Despite the extra Cartesian-to-polar resampling step, OneOcc remains faster overall due to its lighter 2D/3D computation.

Method	2D Enc. + 2D Dec.	Lift2Cart Samp.	Cart2Polar Samp.	3D Dec.	Total	FPS
MonoScene	61.97	2.46	*n.a.*	33.58	98.01	10.20
OneOcc	53.86	2.48	2.40	14.57	73.32	13.64

3.3. Runtime on NVIDIA Jetson AGX Orin

To complement the parameter analysis above, we further report *measured* runtime on an NVIDIA Jetson AGX Orin 64GB (MAX power mode), which provides a more deployment-realistic efficiency reference than desktop-GPU throughput alone. In particular, we compare OneOcc against MonoScene [3] under the QuadOcc panoramic setting using the native input size $1\times3\times370\times1220$ and batch size 1. All numbers are averaged over 100 runs after 5 warm-up iterations. Following a practical edge-deployment setup, the 2D UNet-style image encoder/decoder is executed in INT8 using TensorRT, while the remaining geometry-aware modules are run in FP16. We decompose the latency into four stages: (i) *2D Enc. + 2D Dec.*, *i.e.*, the image-space feature extraction and decoding backbone; (ii) *Lift2Cart Samp.*, which lifts image features into the Cartesian voxel space; (iii) *Cart2Polar Samp.*, the additional Cartesian-to-polar resampling used only in OneOcc; and (iv) *3D Dec.*, the volumetric decoder operating on lifted features.

As shown in Tab. 10, OneOcc achieves a total latency of 73.32 ms (13.64 FPS), compared with 98.01 ms (10.20 FPS) for MonoScene [3], corresponding to an overall speedup of approximately $1.34\times$ on the embedded platform. Notably, this gain is achieved *despite* the additional *Cart2Polar Samp.* stage, which is absent in MonoScene. The main reason is that OneOcc reduces computation in the dominant stages. In the 2D branch, OneOcc requires

53.86 ms versus 61.97 ms for MonoScene, reflecting the lighter dual-projection image backbone design discussed above. The largest reduction appears in the 3D decoder: OneOcc uses only 14.57 ms, whereas MonoScene requires 33.58 ms. This is consistent with the parameter analysis in Tab. 8–9, where the DWLite3D decoder is substantially lighter than the standard dense 3D UNet decoder. By contrast, the geometry-aware sampling operations are cheap: *Lift2Cart Samp.* is nearly identical for both methods (2.48 ms vs. 2.46 ms), and the extra *Cart2Polar Samp.* in OneOcc adds only 2.40 ms. These measurements support the lightweight positioning of OneOcc from an edge-deployment perspective. Rather than relying only on desktop-GPU throughput or theoretical scaling estimates, the Jetson AGX Orin results show that the proposed design remains practically efficient on an embedded robotics platform while providing stronger panoramic semantic occupancy prediction.

3.4. Discussion

Across both benchmarks, OneOcc follows a consistent lightweight design philosophy: (i) concentrate capacity in flexible 2D encoders that can be reconfigured (dual-projection on QuadOcc vs. single-projection on H3O), (ii) employ a depthwise-separable DWLite3D decoder with Hierarchical AMoE-3D to reduce 3D convolutional cost, and (iii) ensure that geometry-specific modules (GDC, BGV-based volumetric fusion, and the AMoE-3D gating heads that select experts based on GradEnergy) remain extremely parameter-efficient. On QuadOcc, this yields a modest increase in parameters but significantly better panoramic SSC under gait-induced jitter; on H3O, the same architecture, with the redundant projection path removed, becomes *substantially* lighter than MonoScene [3] while still providing clear performance gains.

4. QuadOcc: Dataset Construction

Goal and setting **QuadOcc** targets full-surround *semantic occupancy* from a *single* omnidirectional panorama on a quadruped robot. Each frame couples an omnidirectional image captured via Panoramic Annular Lens (PAL) [10] with time-synchronized LiDAR for semi-automatic Ground Truth (GT) construction; outputs are $64\times64\times8$ semantic volumes at 0.4 m per voxel under a 6-class taxonomy. The dataset comprises 10 scenes and $24K$ frames across day/-dusk/night, reflecting realistic legged operation with moderate speeds and tight payload/power budgets.

4.1. Sensor Suite, Time Sync, and Calibration

We mount on a quadruped a Livox Mid-360 LiDAR (with built-in IMU) and an omnidirectional panoramic annular lens (PAL) camera, time-synchronized by hardware trigger and software timestamp alignment. Let $\mathcal{I} = \{I_t\}$ be panoramic RGB streams and $\mathcal{L} = \{\mathcal{P}_t\}$ be LiDAR scans.

PAL (Taylor/OCam) calibration and unwarping. We adopt the generic Taylor (OCam) model [9] for PAL calibration. For an equirectangular pixel $(u, v) \in [0, W) \times [0, H)$ with spherical angles:

$$\phi = \tfrac{2\pi}{W}u - \pi, \quad \theta = \tfrac{\pi}{2} - \tfrac{\pi}{H}v, \tag{1}$$

the corresponding polar radius on the annulus is $r(\theta) = \sum_{i=0}^{N} a_i\theta^i$, and the raw-annulus coordinates are:

$$\begin{bmatrix} u_{\text{raw}} \\ v_{\text{raw}} \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + A\, r(\theta) \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix}, \tag{2}$$

where $\{a_i\}$, (u_0, v_0), and $A \in \mathbb{R}^{2\times2}$ come from calibration. The unwarped equirectangular image I^{equi} is obtained by sampling the raw annulus I^{raw} at $(u_{\text{raw}}, v_{\text{raw}})$.

4.2. Panoramic Semantic Acquisition

We perform pixel-wise open-vocabulary segmentation on I_t^{equi} with Grounded-SAM [11] using a prompt set $\mathcal{P}$ (*e.g.*, "road", "vehicle", "vegetation"), producing a semantic map $S_t \in \{1, \dots, C\}^{H\times W}$ with confidence map Q_t.

4.3. LiDAR–Image Label Transfer

With extrinsics $\mathbf{T}_{L\to C} \in \mathrm{SE}(3)$ and the PAL intrinsics, a LiDAR point $\mathbf{p}_L \in \mathbb{R}^3$ projects to pixel $\mathbf{u} = \pi(\mathbf{T}_{L\to C}\,\tilde{\mathbf{p}}_L)$, where $\tilde{\mathbf{p}}_L$ is in homogeneous coordinates and $\pi(\cdot)$ denotes the PAL projection composed with the equirectangular sampling above. We assign the point label:

$$\ell(\mathbf{p}_L) = S_t(\mathbf{u}), \qquad w(\mathbf{p}_L) = Q_t(\mathbf{u}), \tag{3}$$

thus converting costly 3D annotation into robust 2D segmentation.

4.4. Semantic-Guided Dynamic Mapping

We split points into *static* vs. *dynamic* by semantics $\ell(\cdot)$ (*e.g.*, "vehicle", "pedestrian" as dynamic). A LiDAR-SLAM [12] backend yields a dense global map $\mathcal{M}$ in a world frame. Dynamic instances form *tubes* $\{\Gamma_k\}$ in space-time; each tube provides a motion axis and per-frame bounding boxes $\mathcal{B}_{k,t}$. Background is integrated as static surfels/voxels; dynamic objects are re-inserted in temporal order to preserve their trajectories with minimal manual interaction.

4.5. Voxelization and Majority Voting

We use a fixed occupancy grid $\mathbf{V} \in \{0, \dots, C\}^{X\times Y\times Z}$ (default $64 \times 64 \times 8$ with 0.4 m voxels). Given all points

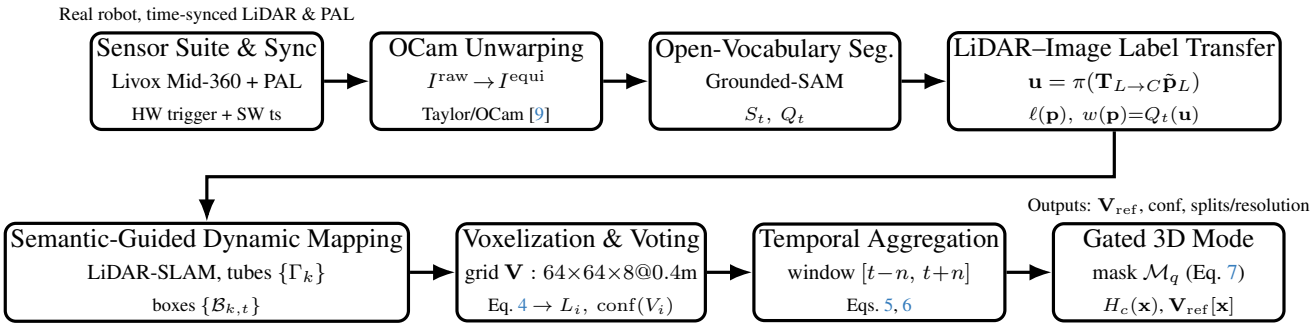

Figure 5. **QuadOcc ground-truth construction pipeline.** From synchronized LiDAR & PAL capture, OCam unwarping and open-vocabulary segmentation provide S_t and Q_t. LiDAR–image label transfer assigns (ℓ, w) to points, followed by semantic-guided dynamic mapping. Voxelization and majority voting (Eq. 4) produce $\mathbf{V}$; temporal aggregation (Eqs. 5, 6) and a gated 3D mode with $\mathcal{M}_q$ (Eq. 7) yield the refined volume $\mathbf{V}_{\text{ref}}$.

$\{P_{i,j}\}$ in voxel V_i, we take a weighted majority vote:

$$L_i = \arg\max_c \sum_j w(P_{i,j})\,\delta\big(\ell(P_{i,j}),\,c\big),$$
$$\text{conf}(V_i) = \frac{\max_c \sum_j \delta\big(\ell(P_{i,j}),c\big)}{\sum_j 1}. \tag{4}$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta and $\text{conf}(\cdot)$ is a per-voxel confidence (later used for quality control).

4.6. Temporal Aggregation and Refinement

To suppress flicker and fill occlusions, we aggregate a temporal window $[t{-}n,\ t{+}n]$. All points inside the window are first transformed to the current frame by $\mathbf{T}_\tau^{\text{cur}} \in \text{SE}(3)$:

$$\mathcal{P}_\tau^{\text{cur}} \;=\; \mathbf{T}_\tau^{\text{cur}}\,\mathcal{P}_\tau, \qquad \forall\,\tau \in [t{-}n,\ t{+}n]. \tag{5}$$

Here, $\mathcal{P}_\tau$ is the LiDAR point set at time τ and $\mathcal{P}_\tau^{\text{cur}}$ its transformed set. After devoxelization and re-voxelization, we accumulate per-class votes in each voxel and take a temporal majority:

$$\mathbf{V}_{\text{agg}}[x,y,z] \;=\; \arg\max_c \sum_{\tau=t-n}^{t+n} \mathbb{I}\big(\mathcal{P}_\tau^{\text{cur}}(x,y,z) = c\big). \tag{6}$$

Here, $\mathbf{V}_{\text{agg}}$ denote the aggregated label volumes. $\mathbb{I}(\cdot)$ denotes the *indicator function*: $\mathbb{I}(\text{predicate}) = 1$ if the predicate is true and 0 otherwise.

Quantization-aware 3D mode (our implementation). Let U denote the *ignore/unlabeled* index, and define a quantization mask that flags "empty-vote" voxels (no support from any frame):

$$\mathcal{M}_q[x,y,z] \;=\; \mathbb{I}\Big(\sum\nolimits_c \sum\nolimits_\tau \mathbb{I}\big(\mathcal{P}_\tau^{\text{cur}}(x,y,z) = c\big) \;=\; 0\Big). \tag{7}$$

We then apply a *gated* 3D mode filter only where $\mathcal{M}_q{=}1$. Using a cubic neighborhood of size $k{\times}k{\times}k$ (default $k{=}5$), and let the local class histogram $H_c(\mathbf{x})$ be:

$$\mathbf{x} = (x,y,z)^\top, \qquad \mathcal{N}_k = \big\{\boldsymbol{\delta} \in \mathbb{Z}^3 :\ \|\boldsymbol{\delta}\|_\infty \le \lfloor k/2 \rfloor\big\}.$$

$$H_c(\mathbf{x}) \;=\; \sum_{\boldsymbol{\delta}\in\mathcal{N}_k} \mathbb{I}\big(\mathbf{V}_{\text{agg}}[\mathbf{x}{+}\boldsymbol{\delta}] = c\big). \tag{8}$$

The refined label $\mathbf{V}_{\text{ref}}$ is then:

$$\mathbf{V}_{\text{ref}}[\mathbf{x}] \;=\; \begin{cases} \arg\max\limits_{c\neq U} H_c(\mathbf{x}), & \mathcal{M}_q[\mathbf{x}] = 1, \\ \mathbf{V}_{\text{agg}}[\mathbf{x}], & \text{otherwise.} \end{cases} \tag{9}$$

Ties and safety: in case of a tie, we keep $\mathbf{V}_{\text{agg}}[\mathbf{x}]$ (no change). If the neighborhood histogram is empty (all U), we fall back to a nearest-neighbor fill in Euclidean space with a small radius (default $r_{\max}{=}5$ voxels) and still ignore U; if no valid neighbor exists within $r_{\max}$ the voxel remains U. Border handling uses a constant pad (out-of-bounds treated as U). This quantization-aware gating prevents the mode operation from overwriting confident voxels while reliably filling holes introduced by multi-frame alignment and re-voxelization.

4.7. Data Splits, Resolution, and Taxonomy

Our legged/humanoid setting prioritizes reliable near-field awareness under tight onboard compute. We therefore bound the grid to $R = [-X_{\max}, X_{\max}] \times [-Y_{\max}, Y_{\max}] \times [Z_{\min}, Z_{\max}]$ with $X_{\max} = Y_{\max} = 12.8\,\text{m}$, $Z_{\min} = -2.0\,\text{m}$, $Z_{\max} = 1.2\,\text{m}$, and a native cubic voxel size $\Delta = 0.4\,\text{m}$, yielding a $64 \times 64 \times 8$ grid that balances (i) adequate radius for low-speed legged motion and foothold safety, (ii) memory/latency budgets for real-time inference on embedded GPUs, and (iii) azimuthal continuity for panoramic cues. We also standardize a six-class schema {vehicle, pedestrian, road, building, vegetation, terrain} to stabilize training under long-tailed class frequencies while keeping labels deployable for onboard planning. The corpus contains ten

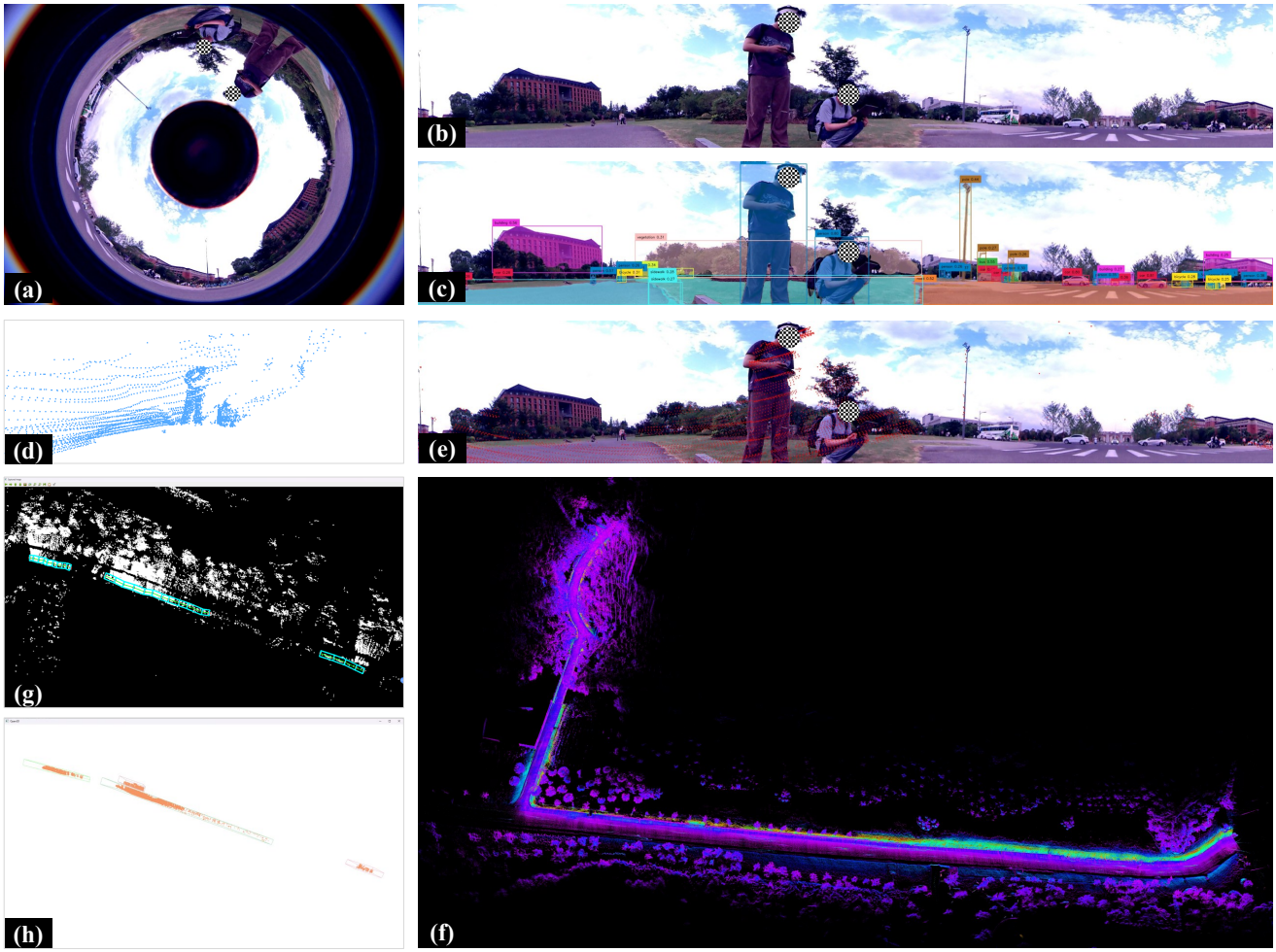

Figure 6. **QuadOcc ground-truth construction pipeline.** (a) Raw panoramic annulus captured by the PAL camera. (b) Equirectangular (ER) panorama obtained by *calibrated* OCam/Taylor [9] unwarping of (a). (c) Open-vocabulary segmentation on the ER image to obtain per-pixel class scores (*e.g.*, Grounded-SAM [11]), yielding $\mathcal{S}_t$ and score maps $\mathbf{Q}_t$. (d) A single time-synchronized LiDAR scan $\mathbf{P}_t$. (e) LiDAR→camera projection followed by *image→point* label transfer: pixels at $u = \pi(T_{L\to C}\tilde{\mathbf{p}}_L)$ assign $(\ell, w) = \mathbf{Q}_t(u)$ to 3D points, producing per-point semantics and confidence. (f) LiDAR SLAM [12] mapping for long-range geometry and temporal consistency. (g) Human-in-the-loop BEV annotation of moving-object trajectories to form tubes/boxes $\{\Gamma_k\}, \{B_{k,t}\}$ for dynamic/static separation. (h) Points are associated with trajectory boxes and motion attributes (direction, instance), then voxelized with majority voting and temporal aggregation to yield the refined semantic occupancy volume $\mathbf{V}_{\text{ref}}$. This pipeline produces $64\times64\times8$ semantic volumes at $0.4\,\text{m}$ resolution under a 6-class taxonomy and is used as GT for training/evaluation on *QuadOcc*.

scenes and $24K$ frames captured under day/dusk/night (the latter also used in stress tests).

4.8. Quality Control and Semi-Automatic Labeling

Our labels originate from multi-frame LiDAR accumulation, Grounded-SAM initialization, and targeted manual fixes in ambiguous regions and around thin structures. During quality control, we flag voxels whose $\text{conf}(V_i)$ or temporal agreement rates fall below thresholds, and prioritize these for human proofreading.

4.9. Analysis: Distribution, Difficulty, and Lighting

Class imbalance. On non-empty voxels, road dominates ($\approx 52.3\%$) followed by vegetation ($\approx 22.8\%$) and building ($\approx 14.0\%$), while vehicle/pedestrian are rare ($\approx 0.9\%/2.1\%$). This long-tail calls for per-class reweighting or sampling-aware training.

Dynamic vs. static. Dynamic tubes reduce motion ghosts and improve temporal coherence near moving objects; they also ease targeted manual verification.

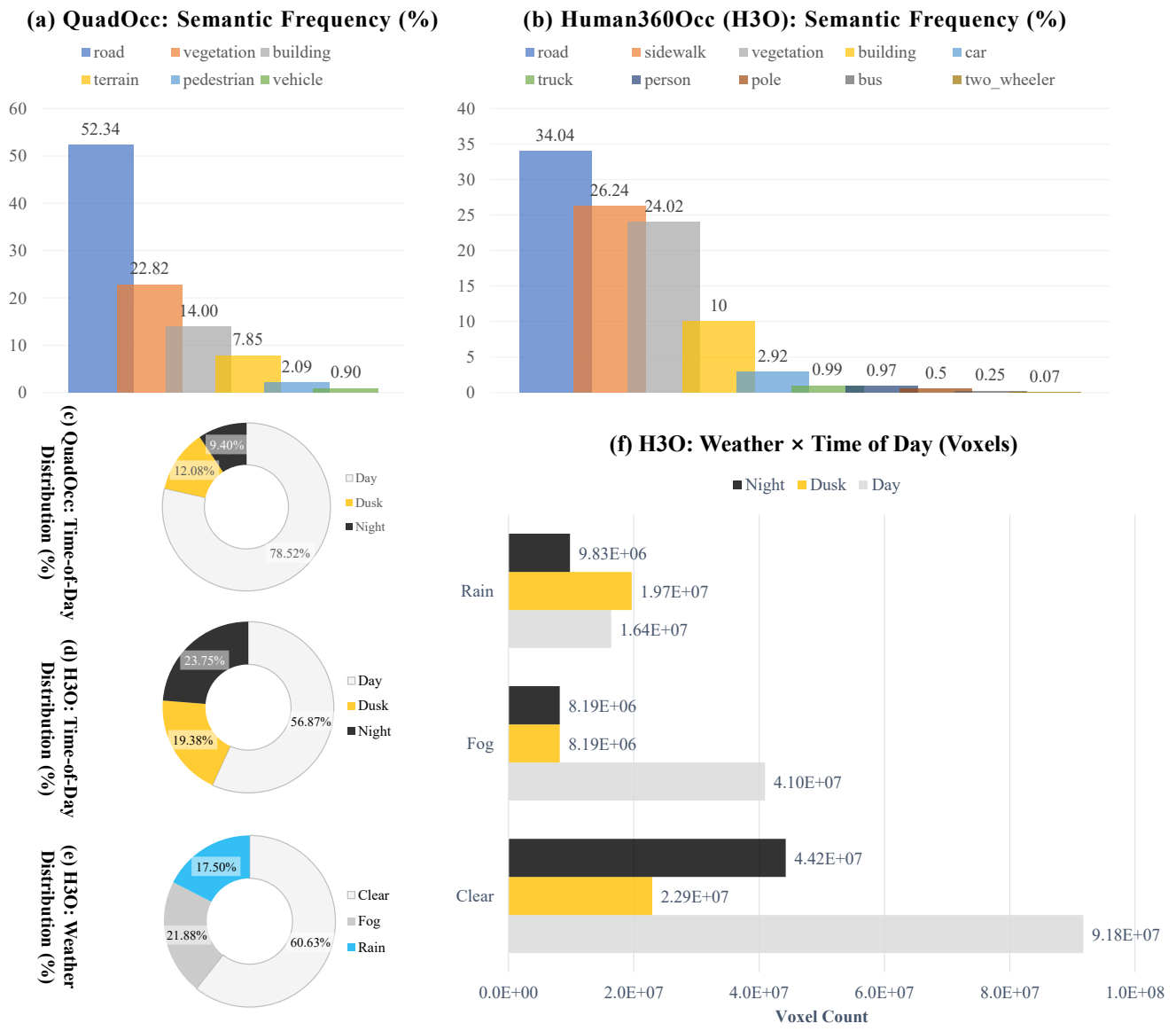

Figure 7. **Dataset statistics for QuadOcc and Human360Occ (H3O).** (a) QuadOcc semantic frequency (non-empty voxels): strong head classes (*e.g.*, *road* $\approx 52.34\%$) with a long tail (*pedestrian*, *vehicle*). (b) H3O semantic frequency (non-empty voxels): most voxels lie in *road/sidewalk/vegetation/building*, with smaller shares for traffic-related rare classes. (c) QuadOcc time-of-day split by frames: Day 78.5%, Dusk 12.1%, Night 9.4%. (d) H3O time-of-day split by voxels: Day 56.9%, Dusk 19.4%, Night 23.8%. (e) H3O weather split by voxels: Clear 60.6%, Fog 21.9%, Rain 17.5%. (f) H3O weather $\times$ time of day (voxels): **clear/day** $91.75M$, **clear/night** $44.24M$, **fog/day** $40.96M$ dominate. *Notes:* (a,b) use non-empty voxel frequencies; (c) is frame-level stats for QuadOcc; (d–f) are voxel-level stats for H3O.

Lighting. Day/dusk/night coverage intentionally stresses robustness. We observe that crepuscular lighting often improves geometric precision (reduced glare), whereas nighttime increases sparsity and aliasing in the far range, recommending stronger temporal priors or test-time refinement.

Confidence-driven curation. The per-voxel confidence and windowed agreement are reliable indicators for spotting residual errors at class boundaries (*e.g.*, vehicle–road).

5. Human360Occ: Dataset Construction

Goal and setting Human360Occ (H3O) targets full-surround semantic occupancy for legged/humanoid platforms from a *single* panoramic stream. Each frame provides RGB, metric depth, and semantic panoramas, and the ground-truth (GT) occupancy at two resolutions ($64\times64\times8$

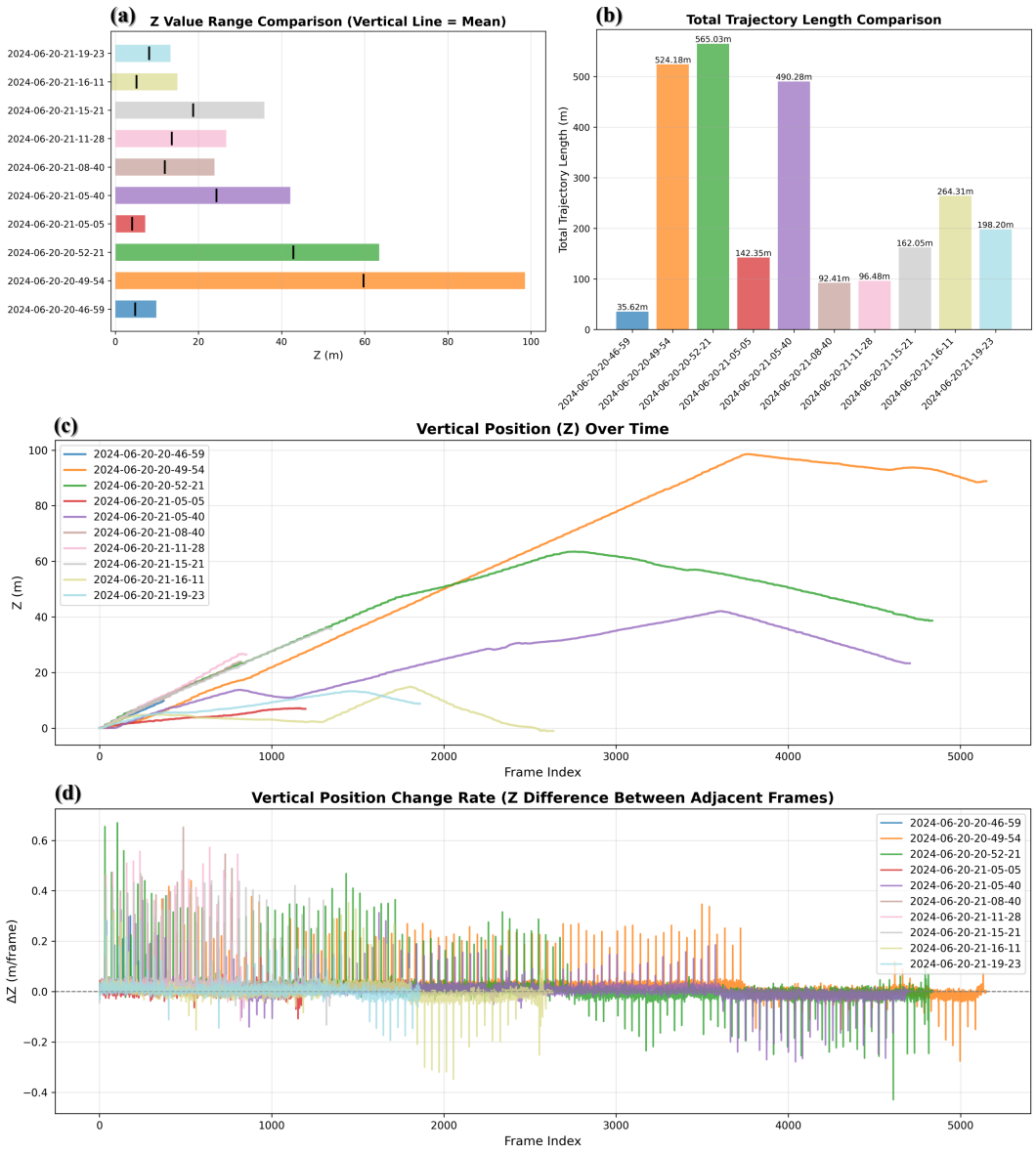

Figure 8. **QuadOcc camera pose statistics.** Visualization of the ego camera pose distribution on QuadOcc in the canonical grid frame (X forward, Y left, Z up). (a)–(d) show complementary views of the aggregated 6-DoF poses over all sequences: top-down ego trajectories, heading statistics, and height/orientation distributions. The coverage concentrates within the 12.8 m radius and $64\times64\times8$ voxel grid used for ground-truth construction, confirming that the chosen range and resolution (Sec. 4.7) are well-aligned with real quadruped operation and the gait-induced body motion that motivates our GDC and bi-grid design.

and $128\times128\times16$). We standardize within-/cross-city splits and record rich metadata for reproducibility.

5.1. Panoramic capture

Cubemap rig and stitching. We mount six synchronized virtual cameras in CARLA [13] (front/right/back/left/up/-down) with unified exposure to avoid face-wise brightness drift. Images from the six faces are stitched into an equirectangular panorama using a calibrated cubemap-to-ER pipeline.

Gait-induced motion (bob). To emulate first-person legged motion, the camera center undergoes a vertical bob whose amplitude and frequency adapt to walking speed. Let t be time (s), $v(t)$ the ego speed, $A(v)$ the amplitude (m), and $f(v)$ the frequency (Hz). The camera z position is:

$$z(t) = z_0 + A\big(v(t)\big) \cdot \sin\big(2\pi \cdot f\big(v(t)\big) \cdot t + \phi\big), \quad (10)$$

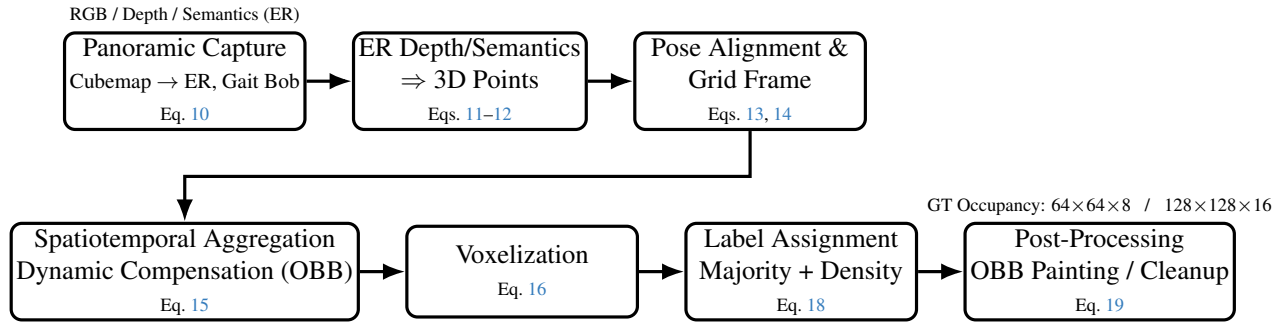

Figure 9. **H3O ground-truth construction pipeline.** Starting from synchronized cubemap panoramas, we stitch to an equirectangular (ER) image and emulate legged motion with a speed-conditioned vertical bob (Eq. 10). ER depth and semantics are back-projected to the camera frame by ER→sphere→3D conversion (Eqs. 11–12). We then align points to a canonical grid frame (forward X, up Z, left Y via Y-flip) using a reference pose (Eqs. 13, 14). A spatiotemporal aggregation window stabilizes moving actors through OBB-based local registration and transport to the reference time τ (Eq. 15), mitigating ghost trails. The aligned points are voxelized within a fixed 3D range to indices (i, j, k) (Eq. 16) and labeled by majority with density safeguards (Eq. 18). Finally, we apply post-processing to ensure complete occupancy of dynamic instances (*OBB painting*) and clean small/noisy components (Eq. 19). Outputs include semantic occupancy at $64\times64\times8$ and $128\times128\times16$ over the same spatial bounds, together with per-sequence metadata for exact reproducibility.

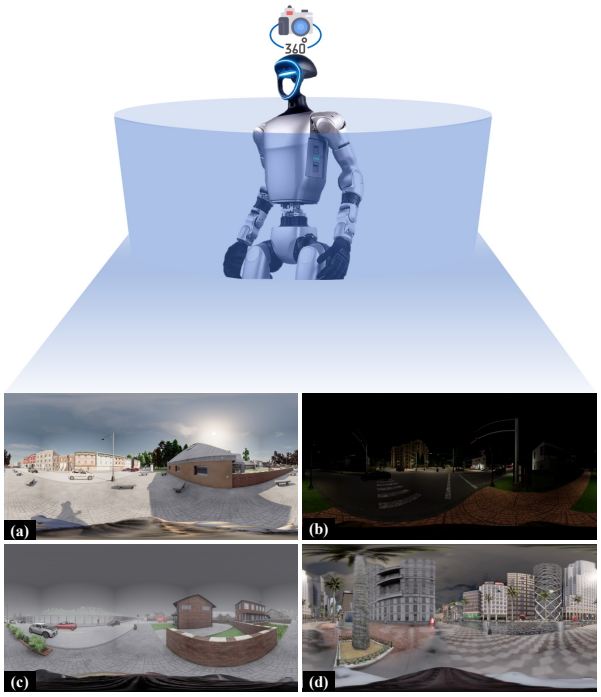

Figure 10. **H3O: Weather and lighting diversity.** Representative equirectangular (ER) panoramas illustrating the variety of conditions covered by *Human360Occ (H3O)*. (a) Clear—daytime; (b) Clear—night; (c) Fog; (d) Rain/overcast in a metropolitan scene. All frames are native ER inputs used directly for training/evaluation. Across H3O, voxel-level distributions span *Clear/Fog/Rain* and *Day/Dusk/Night*, enabling controlled stress-tests under illumination and adverse weather. Ground-truth supervision is provided at two resolutions ($64\times64\times8$ and $128\times128\times16$) over the same spatial bounds.

where $A(v)$ linearly interpolates within $[A_{\min}, A_{\max}]$ and $f(v)$ within $[f_{\min}, f_{\max}]$ as speed increases.

Environment randomization. We randomize the environment at the *sequence* level across maps, weather, and time-of-day with fixed quotas for coverage. Concretely, H3O spans **16** distinct CARLA maps (10 sequences per map) and totals **160** sequences (**8,000** frames). Weather is sampled from {`Clear, Fog, Rain`}, yielding the realized distribution: Clear 97 (≈60.6%), Fog 35 (≈21.9%), Rain 28 (≈17.5%). Time-of-day is sampled from {`day, dusk, night`}, realized as: day 91 (≈56.9%), dusk 31 (≈19.4%), night 38 (≈23.8%). For exact reproducibility, each sequence provides a `sequence_meta.json` that records the loaded map ID, time-of-day label, weather type and parameters, and simulation tick rate.

5.2. From panoramas to a registered 3D point cloud

Depth-to-3D in camera coordinates. Given an equirectangular depth map $r(u, v)$ (meters) of size $W\times H$ with pixel coordinates $(u, v) \in [0, W) \times [0, H)$, we first map pixels to spherical angles:

$$\theta(u) = \tfrac{2\pi}{W}u - \pi, \qquad \phi(v) = \tfrac{\pi}{H}v, \tag{11}$$

and then project to the local camera frame (left-handed, X forward, Y right, Z up):

$$\mathbf{p}_c(u, v) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = r(u, v) \begin{bmatrix} \sin\phi\,\cos\theta \\ \sin\phi\,\sin\theta \\ \cos\phi \end{bmatrix}. \tag{12}$$

We discard invalid measurements (too near/far) and sky pixels and keep $(\mathbf{p}_c, \ell)$ pairs where ℓ is the Cityscapes-style [14] semantic ID.

Pose alignment and grid frame. Let $\mathbf{T}_{w\leftarrow c(t)} \in SE(3)$ be the camera pose at time t and define the grid frame so that

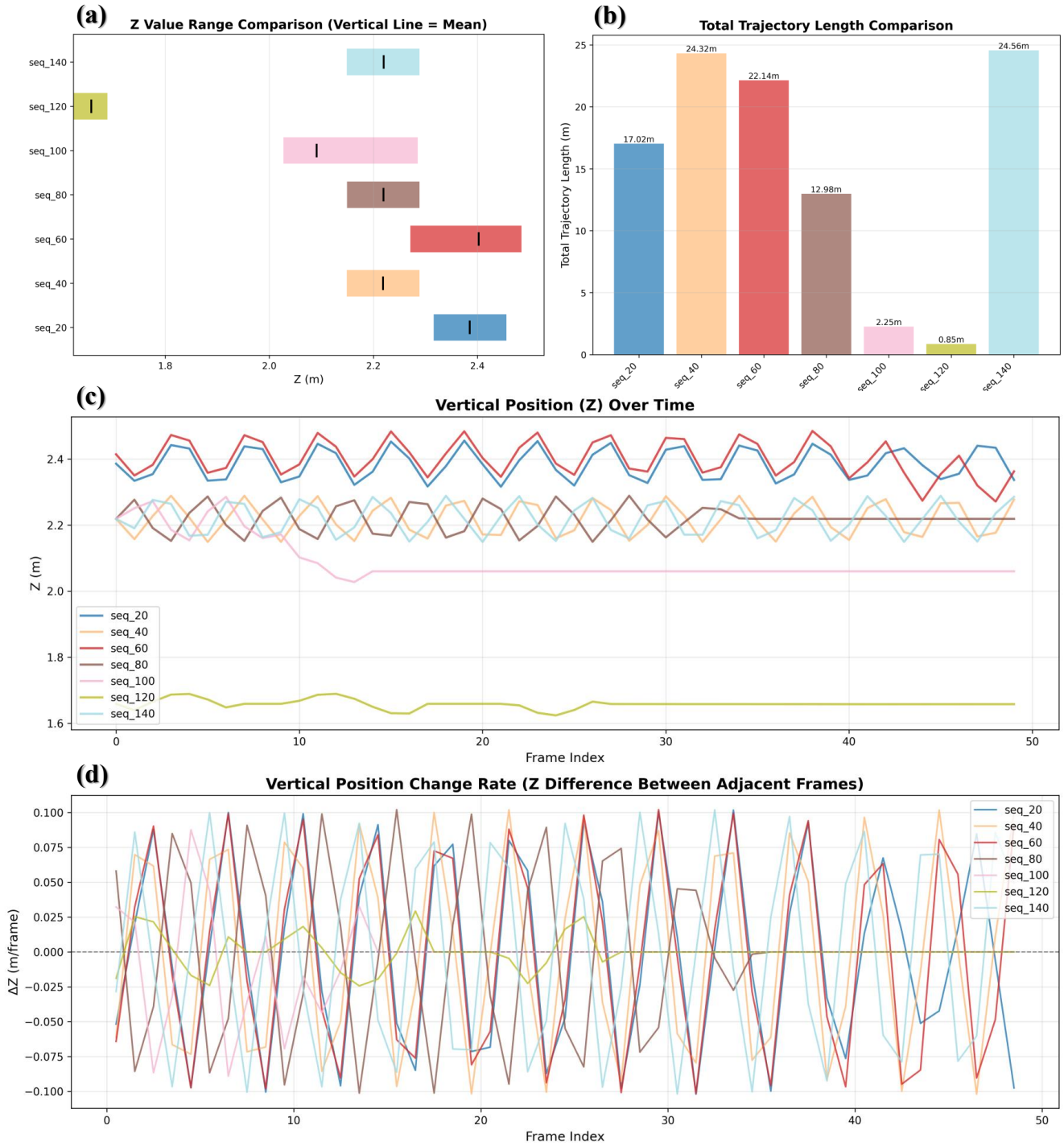

Figure 11. **Human360Occ (H3O) ego pose distribution.** Camera pose statistics for Human360Occ in the same grid frame as used for QuadOcc, aggregated over 160 sequences across 16 CARLA maps and diverse weather/lighting conditions. (a)–(d) provide complementary views of the ego trajectories and 6-DoF poses, including top-down coverage and orientation/height distributions. The plots highlight that H3O exhibits wide heading coverage, non-trivial vertical bobbing, and varied viewpoints, which together stress-test panoramic SSC under gait-like motion and support the standardized within-/cross-city splits and dual-resolution occupancy volumes used in our benchmarks.

X is forward, Z is up, and Y points *left* (a Y-flip w.r.t. the camera). If $\mathbf{F}_y = \mathrm{diag}(1, -1, 1, 1)$, then the world-to-grid transform at a *reference* frame τ is

$$\mathbf{T}_{g \leftarrow w} = \left(\mathbf{T}_{w \leftarrow c(\tau)}\, \mathbf{F}_y\right)^{-1}. \tag{13}$$

A point captured at time t transforms to the grid frame as

$$\tilde{\mathbf{p}}_g \;=\; \mathbf{T}_{g\leftarrow w}\,\mathbf{T}_{w\leftarrow c(t)}\begin{bmatrix}\mathbf{p}_c\\1\end{bmatrix}, \qquad \mathbf{p}_g=\Pi(\tilde{\mathbf{p}}_g), \quad (14)$$

where Π drops the homogeneous coordinate. We aggregate a temporal window of frames around τ (Sec. 5.3).

5.3. Spatiotemporal aggregation with dynamic compensation

We stabilize moving actors (cars, walkers) to the reference time τ to mitigate "ghost tubes" in time-accumulated clouds.

OBB-based local registration. For every dynamic actor, we obtain an oriented bounding box (OBB) at time t via its rigid transform $\mathbf{T}_{\text{box}}^{(t)}$ and extents $\mathbf{e}=(e_x,e_y,e_z)$. A world point $\mathbf{p}_w$ is inside a *scaled* OBB if

$$\left|\left(\mathbf{T}_{\text{box}}^{(t)}\right)^{-1}[\mathbf{p}_w;1]\right| \;\preceq\; \mathbf{e}\odot s \;+\; \mathbf{m}, \quad (15)$$

where $s{>}1$ is a scale factor, $\mathbf{m}$ is a per-axis margin (m), and $\preceq$ is element-wise. For points inside, we transport them from time t to τ by composing actor poses; otherwise we keep the standard camera-pose alignment of (14). As a fall-back, we optionally *remove* dynamic points from non-reference frames or *paint* current-frame OBBs into the grid (Sec. 5.5).

5.4. Voxelization and semantic labeling

Grid bounds and resolution. We set an asymmetric vertical range anchored at the ego camera: $Z_{\min}=-2.0\,\text{m}$ ensures sufficient ground coverage and small depressions below the head-height camera, while $Z_{\max}=1.2\,\text{m}$ captures overhanging obstacles without wasting vertical resolution. We therefore define a fixed 3D range

$$\mathcal{R}=[-X_{\max},X_{\max}]\times[-Y_{\max},Y_{\max}]\times[Z_{\min},Z_{\max}],$$

with defaults $X_{\max}=Y_{\max}=12.8\,\text{m}$, $Z_{\min}=-2.0\,\text{m}$, $Z_{\max}=1.2\,\text{m}$, and a native cubic voxel size $\Delta=0.4\,\text{m}$. The native grid size is $\mathbf{N}=(N_x,N_y,N_z)=\lfloor(\max-\min)/\Delta\rceil$, and we also export higher resolutions (*e.g.*, 128×128×16) over the *same* $\mathcal{R}$.

Indexing. A grid index for a point $\mathbf{p}_g=(x,y,z)$ is

$$(i,j,k)\;=\;\left\lfloor\frac{x-x_{\min}}{\Delta}\right\rfloor,\left\lfloor\frac{y-y_{\min}}{\Delta}\right\rfloor,\left\lfloor\frac{z-z_{\min}}{\Delta}\right\rfloor. \quad (16)$$

Label assignment with density safeguards. Let $\mathcal{P}_{i,j,k}$ be the set of aggregated points falling into voxel (i,j,k) and $\ell(\cdot)$ their semantic IDs. We adopt majority voting with a minimum support:

$$\hat{c}_{i,j,k}\;=\arg\max_c\;\#\{\,\mathbf{p}\in\mathcal{P}_{i,j,k}\;:\;\ell(\mathbf{p})=c\,\}, \quad (17)$$

$$\text{keep if }\;\#\mathcal{P}_{i,j,k}\;\geq\;\tau, \qquad \tau\;=\;\max\big(N_{\min},\,\lceil\rho_{\min}V_{\text{vox}}\rceil\big), \quad (18)$$

where $\#S$ denotes the cardinality (number of elements) of a set S. $N_{\min}$ is a point-count threshold, $\rho_{\min}$ is an optional density floor (points/m^3), and V_{vox} is the voxel volume. Unless stated otherwise, we use majority voting with $N_{\min}=4$ and optionally enable the density floor.

5.5. Post-processing

OBB painting. To ensure complete occupancy for current-frame dynamic instances, we fill their OBBs in the grid using a scaled+margin ROI $(s,\mathbf{m})$:

$$\text{for voxels } v\in\text{ROI}:\quad \text{occ}[v]\leftarrow\begin{cases}c_{\text{actor}}, & \texttt{override},\\ c_{\text{actor}}\text{ if occ}[v]=0, & \texttt{keep}.\end{cases} \quad (19)$$

We analogously paint known static vehicles (*environment objects*) with a slightly tighter ROI.

Noise removal. We remove isolated singletons (no same-label neighbors) and delete small connected components below a voxel-count threshold using spatial-connectivity (we favor 6-connectivity to preserve thin structures).

Semantic schema and storage. Occupancy volumes are stored as `uint8` NumPy arrays of shape $[N_x,N_y,N_z]$, where 0 denotes empty and 1∼28 follow a Cityscapes-style palette (*e.g.*, road/sidewalk/building/vegetation/person/car/truck/bus/twowheeler, *etc.*). We export both native and 128×128×16 grids, plus optional colored PCDs of voxel centers for visual auditing.

5.6. Train/val splits and stats

H3O comprises 160 sequences / $8K$ frames across 16 maps under diverse weather and lighting. Each frame includes RGB, metric depth, ER semantics, poses, and GT occupancy at two resolutions, supporting within-city (per-map 8:2) and cross-city (12/4 maps) protocols.

5.7. Default configuration

Unless specified otherwise, we use: time window $T=50$ frames centered at τ; native voxel size $\Delta=0.4\,\text{m}$ over $\mathcal{R}=[\pm12.8\,\text{m}]$ (XY) and $[-2.0,1.2]\,\text{m}$ (Z); label assignment `majority` with $N_{\min}{=}4$ (optionally density-gated); OBB fill for dynamic and static vehicles enabled; and singleton/small-component removal.

Deliverables. For each sequence, we provide ER panoramas (RGB/Depth/Semantics), per-frame `ground_truth.json` (poses, actor OBBs, ego info), `sequence_meta.json`, native and 128×128×16 occupancy volumes, and optional point-cloud visualizations for quality control.

6. Implementation Details of Baselines

General protocol. All camera-only baselines are *retrained from scratch* on our panoramic legged-robot benchmarks with *minimal modifications*: we preserve original backbones/heads and only adapt dataset hooks, projection reshaping, voxel grid sizes, and loss toggles required by the panoramic small-volume regime. Unless otherwise noted, we fix seeds, follow original training schedules, and report results with the default settings stated under each method.

6.1. Overview

We organize per-method details in a uniform structure to facilitate verification and extension:

- **[Method Name]**: a short protocol summary (minimal changes, training schedule).
- **QuadOcc**: dataset- and grid-specific settings for retraining on our QuadOcc split.
- **H3O**: corresponding settings for retraining on H3O.
- **Design rationale**: why these choices are necessary under panoramic, small-volume settings.

6.2. MonoScene

Scope. We keep MONOSCENE's [3] original design and adapt only the dataset branch, projection reshape, and losses required by panoramic inputs and smaller grids; all models are trained from scratch.

QuadOcc. Data and taxonomy. QuadOcc provides per-frame panoramic RGB and voxelized semantic occupancy within the 3D range $X{=}Y{=}\pm\,12.8\,\text{m}$, $Z \in [-2.0,\ 1.2]\,\text{m}$, discretized into a $64{\times}64{\times}8$ grid at $0.4\,\text{m}$ resolution. We train on 7 labels with `empty` at index 0 and six non-empty classes: {`vehicle`, `pedestrian`, `road`, `building`, `vegetation`, `terrain`}. Images are equirectangular panoramas of size $370{\times}1220$ produced by calibrated OCam/Taylor [9] unwarping of a PAL camera. We use the official train/val sequences and sample every 5 frames.

Architecture adaptations. We adopt the small-grid CP and standard lifting without extra magnification:

- *Context Prior (CP) for small grids:* the CP module (`CPMegaVoxels`) dynamically adjusts kernel sizes/strides as a function of the minimum side-length; if any side is < 3, it falls back to 1^3 kernels with unit stride to avoid over-downsampling.
- *Logit–GT alignment:* the decoder head ensures output logits match the $64{\times}64{\times}8$ GT grid exactly after lifting.
- *FLoSP lifting (QuadOcc):* we use the *standard* FLoSP reshape consistent with the chosen `project_scale`.

Training hyper-parameters. We mostly keep the KITTI [4] recipe and only adapt what is necessary:

- *Scene/voxel:* `full_scene_size`$=(64, 64, 8)$, voxel size $0.4\,\text{m}$, origin $[-12.8, -12.8, -1.2]$.
- *Backbone/decoder:* feature dimension 64.
- *Classes/losses:* $n\_classes{=}7$ (including `empty`); enable `CE_ssc_loss`, `sem_scal_loss`, `geo_scal_loss`, and `fp_loss` with `frustum_size`$=8$; disable relation loss (`relation_loss=false`) due to small grids and panoramic distortions.
- *Optimization:* AdamW [15] with LR$=10^{-4}$, WD$=10^{-4}$; MultiStepLR (`milestones`$=[20]$, $\gamma = 0.1$); `max_epochs`$=30$, `batch_size`$=1$, `n_gpus`$=1$.
- *Data loading:* `num_workers_per_gpu`$=16$.
- *Augmentation:* color jitter $(0.4, 0.4, 0.4)$ at train-time; no augmentation at val/test.

Camera and normalization. We use OCam/-Taylor [9] intrinsics and the precomputed annulus→equirectangular maps. RGB normalization: $\mu = \left[\frac{123.971}{255}, \frac{123.564}{255}, \frac{164.785}{255}\right]$, $\sigma = \left[\frac{84.271}{255}, \frac{88.170}{255}, \frac{72.966}{255}\right]$.

Loss weighting. We compute frequency-based class weights on the QuadOcc training split (non-empty voxels) for the CE term to stabilize rare dynamic categories and the head class (`road`).

Design rationale. (i) *Small-grid & dynamic CP:* with $64{\times}64{\times}8$, aggressively strided 3^3 stacks can collapse features; dynamic kernels retain receptive fields without over-downsampling. (iii) *Disable relation loss:* co-occurrence priors that help on KITTI [4] tend to over-smooth azimuthal seams and suppress rare classes under panoramas. (iv) *jitter on:* moderate color jitter improves robustness.

H3O. Projection (FLoSP). We use the *standard* FLoSP reshape consistent with `project_scale`$=1$.

Voxel grid and scene bounds. Two modes are supported: *native* $64{\times}64{\times}8$ at $0.4\,\text{m}$/voxel (default) and *fixed 128* $128{\times}128{\times}16$ at $0.2\,\text{m}$/voxel; both share $25.6{\times}25.6{\times}3.2\,\text{m}$ with origin $[-12.8, -12.8, -2.4]\,\text{m}$.

Remapping and losses. We adopt the default 11-class remap (`car`/`truck`/`bus` separated; traffic furniture merged). We enable `CE_ssc_loss`, `sem_scal_loss`, `geo_scal_loss`, `fp_loss` (`frustum_size`$=8$), and disable relation loss.

Image pipeline and optimization. Panoramas are resized to $352{\times}1216$ and normalized with statistics ($\mu{=}[0.485, 0.456, 0.406]$, $\sigma{=}[0.229, 0.224, 0.225]$). We follow the KITTI [4] schedule (AdamW [15], LR$=10^{-4}$, WD$=10^{-4}$; MultiStepLR at epoch 20 with $\gamma{=}0.1$; 30 epochs). The smaller grid permits `batch_size`$=2$ and `num_workers_per_gpu`$=12$ on a single GPU.

Design rationale. (i) *Equirect + standard reshape:* no extra magnification is required for uniform azimuthal sampling; this reduces aliasing. (ii) *Two grid modes:* `native` trades speed for scale; `fixed128` recovers thin structures. (iii) *Relation loss off:* improves stability under panoramic distortions and class imbalance.

Table 11. **QuadOcc vs. KITTI [4] (MonoScene [3]).**

Component	KITTI [4]	QuadOcc	Rationale
Grid size	256×256×32	64×64×8	Embedded compute / near-field focus
Voxel size	0.2 m	0.4 m	Balanced radius vs. memory
Classes	20	6	legged-relevant labels
Relation loss	`true`	`false`	Unstable / unhelpful on small grids
3D decoder	KITTI default	small-grid CP preserved	Prevent over-downsampling

Table 12. **H3O vs. KITTI [4] (MonoScene [3]).**

Component	KITTI [4]	H3O	Rationale
Grid size	256×256×32	64×64×8 / 128×128×16	Near-field focus vs. detail
Voxel size	0.2 m	0.4/0.2 m	Match grid mode and memory
Projection scale	2	1 (standard reshape)	Lifting without over-enlargement
Classes	20	11	Stable yet granular labels
Relation loss	`true`	`false`	Avoid over-smoothing rare classes
Batch size	1	2	Smaller grid ⇒ higher throughput

6.3. SGN

Scope. We keep SGN's [16] transformer-based image-to-voxel lifting and the original detector–head layout, and make only minimal changes required by panoramic inputs and small voxel grids. All models are trained from scratch. Single-GPU batches ≥ 1 are fully supported by us in both training and testing.

QuadOcc. Data and taxonomy. QuadOcc provides single-frame omnidirectional RGB and voxelized semantic occupancy within $X{=}Y{=}\pm 12.8\,\text{m}$ and $Z \in [-2.0,\ 1.2]\,\text{m}$, discretized at $0.4\,\text{m}$ into a $64{\times}64{\times}8$ grid. We use 7 labels with `empty` at index 0 and six non-empty classes: {`vehicle`, `pedestrian`, `road`, `building`, `vegetation`, `terrain`}. Images are equirectangular panoramas of size $370{\times}1220$ obtained via calibrated OCam/Taylor [9] unwarping.

Architecture adaptations (transformer lifting).

- *3D queries.* We instantiate one 3D query per voxel center within the native bounds; each query carries a 3D sinusoidal positional encoding.
- *2D tokens for panoramas.* ER images are encoded by a 2D CNN and flattened into tokens with 2D positional encodings augmented by spherical (longitude/latitude) terms computed from OCam/Taylor [9] intrinsics.
- *Cross-attention lifting.* Multi-head cross-attention lets 3D queries sample informative 2D tokens to produce fused voxel features; The decoder head maps fused features to per-voxel logits aligned with the $64{\times}64{\times}8$ GT grid.

Losses, schedules, and IO. We enable `CE_ssc_loss`, `sem_scal_loss`, and `geo_scal_loss`. Optimization uses AdamW [15] (LR $= 2{\times}10^{-4}$, WD $= 10^{-2}$) with CosineAnnealing and a linear warmup of 500 iterations; `total_epochs`$=$ 48, `batch_size`$=$ 1, `workers_per_gpu`$=$ 4. RGB normalization uses dataset statistics $\mu = \left[\frac{123.971}{255}, \frac{123.564}{255}, \frac{164.785}{255}\right]$, $\sigma = \left[\frac{84.271}{255}, \frac{88.170}{255}, \frac{72.966}{255}\right]$.

H3O. Grid modes. We support voxel grid mode with the spatial bounds $25.6{\times}25.6{\times}3.2\,\text{m}$ and origin $[-12.8, -12.8, -2.4]\,\text{m}$: *native* ($64{\times}64{\times}8$ at $0.4\,\text{m}$/voxel; default). The transformer lifting is unchanged: 3D queries at voxel centers cross-attend to ER image tokens.

Remap, losses, and schedule. We adopt the default 11-class remap (`car`/`truck`/`bus` separated; traffic furniture merged), and enable the same `CE`/`sem_scal`/`geo_scal` losses. Panoramas are resized to $608{\times}1216$ and normalized with statistics ($\mu{=}[0.485, 0.456, 0.406]$, $\sigma{=}[0.229, 0.224, 0.225]$). We follow a KITTI-style schedule [4]: AdamW (LR $= 10^{-4}$, WD $= 10^{-4}$), MultiStepLR at epoch 20 with $\gamma = 0.1$, for 30 epochs; `batch_size`$=$ 2, `num_workers_per_gpu`$=$ 12.

6.4. VoxFormer

Scope. We follow VoxFormer [6]'s two-stage design and keep the model (VoxFormer-S) with minimal adaptations for panoramic inputs and small voxel grids. We re-implement the head/encoder hooks to support batch ≥ 1 and retrain all models from scratch.

QuadOcc. Data and taxonomy. QuadOcc provides single-frame omnidirectional RGB and voxelized semantic occupancy within $X{=}Y{=}\pm 12.8\,\text{m}$ and $Z \in [-2.0,\ 1.2]\,\text{m}$, discretized at $0.4\,\text{m}$ into a $64{\times}64{\times}8$ grid.

Table 13. **KITTI [4] vs. QuadOcc (SGN [16]).**

Component	KITTI [4] baseline	QuadOcc (ours)	Rationale
Grid size (GT)	$256\times256\times32$	$64\times64\times8$	Near-field grid for embedded budget
Voxel size	0.2 m	0.4 m	Trade detail for memory/speed
Bounds (x/y/z)	$[0, -25.6, -2, 51.2, 25.6, 4.4]$	$[-12.8, -12.8, -1.2, 12.8, 12.8, 2.0]$	Symmetric, small-volume scenes
2D input	Pinhole RGB	ER panorama 370×1220	Panoramic coverage
2D tokens/PE	Planar PE	ER tokens + spherical (lon/lat) terms	Longitude/latitude-aware tokenization
Lifting	X-attn: 3D←2D	Same (X-attn)	Transformer lifting unchanged
3D queries	Per voxel (dense)	Per voxel (dense)	Same mechanism; fewer voxels
Classes	20	7 (`empty`+6)	Dataset taxonomy
Losses	CE/sem_scal/geo_scal	Same	Stable on small grids

Table 14. **KITTI [4] vs. H3O (SGN [16]).**

Component	KITTI [4] baseline	H3O (ours)	Rationale
Grid size (GT)	$256\times256\times32$	$64\times64\times8$ (*native*)	Matched to human-scale scenes
Voxel size	0.2 m	0.4 m	Memory/speed under panorama
Bounds (x/y/z)	$[0, -25.6, -2, 51.2, 25.6, 4.4]$	$[-12.8, -12.8, -2.4, 12.8, 12.8, 0.8]$	Symmetric bounds
2D input	Pinhole RGB	ER panorama 608×1216	Wider FoV under H3O
2D tokens/PE	Planar PE	ER tokens + spherical (lon/lat) terms	Azimuth/elevation-aware tokens
Lifting	X-attn: 3D←2D	Same (X-attn)	Transformer lifting unchanged
3D queries	Per voxel (dense)	Per voxel (dense)	Same mechanism; fewer voxels
Classes	20	11 (default remap)	default mapping on H3O
Losses	CE/sem_scal/geo_scal	Same	Consistent objectives
Batching	B=1 common	B=2 (single GPU)	Smaller grid ⇒ higher throughput

We train with 7 labels (index 0 is `empty`; non-empty: {`vehicle`, `person`, `road`, `building`, `vegetation`, `terrain`}). Images are equirectangular panoramas of size 370×1220 produced by calibrated OCam/Taylor [9] unwarping of a PAL camera.

Architecture adaptations.

- *Head.* We replace the baseline head with `VoxFormerHeadQuad` (class count $20\to7$; frequency-based class weights). BEV positional encoding (PE): We extensively explored two granularities that are compatible with the repository's square-PE constraint: *(a)* 64×64 PE for a compact $32\times32\times4$ BEV query grid (our default for fair comparisons); and *(b)* 512×512 PE for $128\times128\times16$ queries (the largest setting we tested). Despite the substantially larger query sets in (b)—which increase memory and latency considerably—we observed no material accuracy gains on QuadOcc/H3O. In particular, (b) frequently leads to out-of-memory on a 24 GB RTX 4090 even at batch size 1 under panoramic inputs; enabling activation checkpointing/mixed precision improves stability but more than $10\times$ wall-clock time with negligible accuracy change. Given the fairness to other transformer-based baselines (*e.g.*, SGN [16]) and the unfavorable compute–benefit trade-off, we *report* the 64×64 PE results and regard (b) as ablations without qualitative advantage.
- *Encoder/Layers.* We adapt reference-point generation to QuadOcc bounds and voxel sizes; cross/self attention settings follow the small model.
- *Proposals.* Instead of reading proposals from `img_metas`, we dynamically voxelize 3D points to obtain sparse 3D proposals (Sec. Stage-1 below), enabling both pseudo-LiDAR and GT-LiDAR routes.

Stage-1 (depth⇒3D proposals). We use two interchangeable routes to generate 3D points for proposal voxelization:

- *Pseudo-LiDAR.* We run MONODEPTH2 [17] on ER panoramas and back-project with OCam/Taylor [9] intrinsics to obtain per-pixel 3D points, then voxelize them into a coarse $64\times64\times8$ prior.
- *GT-LiDAR.* We alternatively voxelize the provided LiDAR points to form the same coarse prior.

Both routes train the identical Stage-1 network (query update via multi-head deformable cross-attention from 3D queries to ER image features). In all cases, Stage-1 outputs $64\times64\times8$ logits.

Stage-2 (refinement). Stage-2 takes Stage-1 logits as priors and refines them with a shallow stack of query updates and 3D convolutions, keeping the same class set and bounds. Unless specified, reported results include the *GT-LiDAR Stage-1* variant followed by Stage-2.

Training hyper-parameters. We keep the VoxFormer-S recipe and adapt only what is necessary:

- *Scene/voxel.* `full_scene_size`= $(64, 64, 8)$, voxel size 0.4 m, origin $[-12.8, -12.8, -1.2]$, eval range 25.6 m.
- *Transformer (S).* embed_dims= 128, cross/self layers= 3/2, sampled points per head= 8, FFN= 1024.
- *Optimization.* AdamW (LR = 2×10^{-4}, WD = 10^{-2}), CosineAnnealing with 500 warmup iters, `total_epochs`= 20, `batch_size`= 4,

workers_per_gpu= 4, gradient clip (max_norm= 35).

- *Losses.* We enable CE_ssc_loss, sem_scal_loss, and geo_scal_loss.

Image pipeline and normalization. Panoramas are 370×1220; standard color jitter $(0.4, 0.4, 0.4)$ at train-time; no val/test augmentation. RGB normalization uses QuadOcc statistics $\mu = \left[\frac{123.971}{255}, \frac{123.564}{255}, \frac{164.785}{255}\right]$, $\sigma = \left[\frac{84.271}{255}, \frac{88.170}{255}, \frac{72.966}{255}\right]$.

Class weighting. We compute frequency-based weights on the QuadOcc training split (non-empty voxels) for the CE term to stabilize rare dynamic classes and the head class (road).

Design rationale. *(i) BEV positional encoding.* We made a best-effort push to larger PE (up to 512×512) to increase the number of BEV queries and potential receptive-field coverage. However, under omnidirectional inputs and small near-field grids, the resulting memory/latency costs are disproportionately high and do not translate into consistent accuracy gains; thus we favor the 64×64 PE for fair and efficient comparison. *(ii) Proposal voxelization (pseudo vs. GT).* Pseudo-LiDAR from MONODEPTH2 [17] preserves the monocular setting but introduces depth noise that weakens sparse-query coverage; training Stage-1 with GT-LiDAR proposals improves stability yet still does not close the gap to stronger transformer baselines [16] on QuadOcc/H3O. *(iii) Minimal deltas from the baseline.* We keep transformer depths/widths unchanged and only modify class hooks, reference-point generation, and proposal construction; this avoids capacity confounds and keeps results reproducible under panoramic small-volume scenes.

H3O. Data and taxonomy. H3O provides single-frame omnidirectional RGB and voxelized semantic occupancy within $X{=}Y{=}\pm 12.8$ m and $Z \in [-2.4,\ 0.8]$ m, discretized at 0.4 m into a native 64×64×8 grid. We adopt the default 11-class remap with empty at index 0 and ten non-empty classes: {road, sidewalk, building, vegetation, car, truck, bus, two_wheeler, person, pole}. Panoramas are equirectangular (ER).

Architecture adaptations (omni variants).

- *Head.* We use VoxFormerHeadOmni (class count 20→11; frequency-based class weights).
- *Encoder/Layers.* We operate with ER cameras and *consume precomputed* per-voxel, per-view image correspondences from img_metas: projected_pix stores the pixel coordinates (u, v) of voxel centers on each ER image, and fov_mask flags whether a voxel lies inside the valid field-of-view. These caches: (i) avoid recomputation of spherical projection in the forward pass, (ii) faithfully adapt to the equirectangular model, and (iii) naturally extend to multi-camera ER inputs by concatenating per-view tokens and masks. Reference-point generation is adapted to the symmetric bounds $[-12.8, 12.8]^2 \times [-2.4, 0.8]$.
- *BEV positional encoding (PE) → query grid.* The repository enforces a square PE whose side equals the BEV query side. Given H3O's native 64×64×8 GT, we evaluate two compatible query configurations: (a) 64×64 PE ⇒ 32×32×4 queries (default for fairness/efficiency), and (b) 512×512 PE ⇒ 128×128×16 queries (largest tested; ablation).

Stage-1 (depth⇒3D proposals). We consider two proposal routes within the H3O bounds:

- *Pseudo-depth (ER).* Back-project ER depth from MONODEPTH2 [17] and voxelize the 3D points.
- *Ground-truth depth (ER).* To rule out reproduction artifacts, we also voxelize proposals from *ground-truth* depth maps and train VOXFORMER-S [6] accordingly; we *report* its accuracy alongside the pseudo-depth variant.

Stage-1 performs multi-head deformable cross-attention from 3D queries to ER image features and outputs 64×64×8 logits.

Stage-2 (refinement). Stage-2 refines Stage-1 logits using a shallow stack of query updates and 3D convolutions under the same bounds/classes. Unless otherwise specified, H3O results correspond to Stage-1→Stage-2.

Training hyper-parameters.

- *Scene/voxel.* full_scene_size= $(64, 64, 8)$, voxel size 0.4 m, origin $[-12.8, -12.8, -2.4]$, eval range 25.6 m.
- *Transformer (S).* embed_dims= 128, cross/self layers= 3/2, sampled points per head= 8, FFN= 1024.
- *Optimization.* AdamW (LR = $2{\times}10^{-4}$, WD = 10^{-2}), CosineAnnealing with 500 warmup iters, total_epochs= 24, batch_size= 1, workers_per_gpu= 4, gradient clip (max_norm= 35).
- *Losses.* CE_ssc_loss, sem_scal_loss, geo_scal_loss are enabled.

Image pipeline and normalization. Panoramas are resized to 608×1216; we use statistics for normalization (μ=[0.485, 0.456, 0.406], σ=[0.229, 0.224, 0.225]). No test-time augmentation is used.

Design rationale. *(i) Cached ER correspondences.* Precomputing projected_pix and fov_mask yields deterministic cross-attention sampling, eliminates redundant spherical projection during the forward pass, and scales to multi-camera ER inputs—this is the key omni feature we introduce to strengthen VoxFormer [6] on H3O. *(ii) BEV PE under square-constraint.* We pushed the BEV PE to 512×512 (*i.e.*, 128×128×16 queries) to maximize coverage; however, the substantial memory/latency increase did not translate into consistent accuracy gains on H3O, mirroring our QuadOcc findings. *(iii) Depth-driven proposals (pseudo and GT).* Using *ground-truth* depth removes the confound from depth estimation noise and confirms that the performance gap is *not* due to reproduction ar-

Table 15. **KITTI [4] vs. QuadOcc (VoxFormer-S [6]).**

Component	KITTI baseline	QuadOcc (ours)	Rationale
Grid size (GT)	256×256×32	64×64×8	Embedded compute / near-field focus
Voxel size	0.2 m	0.4 m	Balanced radius vs. memory
BEV PE → queries	128×128×16	32×32×4 (default); 128×128×16 (ablations)	Larger PE yields heavy memory/latency with no material gains
Classes	20	7	Panoramic, legged-relevant labels
Proposal source	meta/file	voxelized (pseudo/GT) Depth	Rule out reproduction artifacts
Batch size	1	4	Smaller grid ⇒ higher throughput

Table 16. **KITTI [4] vs. H3O (VoxFormer-S [6]).**

Component	KITTI baseline	H3O (ours)	Rationale
Grid size (GT)	256×256×32	64×64×8	Native near-field
Voxel size	0.2 m	0.4	Match grid mode and memory
BEV PE → queries	128×128×16	32×32×4 (default); 128×128×16 (ablation)	Square-PE constraint; larger PE = heavy cost
Omni projection	pinhole	ER	Deterministic sampling
Proposals	meta/file	voxelized (pseudo/GT)	Rule out reproduction artifacts

tifacts: even with GT-depth proposals and Stage-2 refinement, VoxFormer-S [6] remains markedly weaker than stronger transformer baselines (*e.g.*, OccFormer [18], SGN [16]) in our panoramic setting.

6.5. OccFormer

Scope. We follow OccFormer [18] and preserve the original architecture (2D backbone + LSS-style view transformer [19] + 3D encoder/decoder + Mask2Former-style head [20] with a DETR decoder [21]). To adapt to panoramic inputs and small voxel grids in QuadOcc, we only modify the dataset branch, spatial bounds, class taxonomy, and sampling/normalization. All models are trained from scratch; the core network depths/widths remain unchanged.

QuadOcc. Data and taxonomy. QuadOcc provides single-frame omnidirectional RGB and voxelized semantic occupancy within $X{=}Y{=}\pm 12.8$ m and $Z \in [-1.2, 2.0]$ m, discretized at 0.4 m into a 64×64×8 grid. We train on 7 labels (index 0 is `unlabeled`; non-empty classes: {`vehicle`, `person`, `road`, `building`, `vegetation`, `terrain`}). Images are unfolded equirectangular panoramas; Input resolution is $(384, 1280)$; We use dataset-specific normalization (mean $[123.971, 123.564, 164.785]$, std $[84.271, 88.170, 72.966]$).

Architecture/data adaptations.

- *Camera/input.* We switch `camera_used` from `['left']` (KITTI [4]) to `['images_unfold']` to consume unfolded panoramic RGB.
- *View transformer bounds.* We set `point_cloud_range`= $[-12.8, -12.8, -1.2, 12.8, 12.8, 2.0]$ and `occ_size`= $[64, 64, 8]$ (voxel size 0.4 m). This symmetrizes the near-field volume relative to KITTI's [4] 51.2×51.2×6.4 m.
- *Classes.* We replace the 20-class SemanticKITTI taxonomy with the 7-class QuadOcc taxonomy in the head and evaluator.

Model components (unchanged). We keep the KITTI [4] baseline components: `CustomEfficientNet-B7` backbone, `SECONDFPN` image neck, `ViewTransformerLiftSplatShootVoxel` for image→voxel lifting, `OccupancyEncoder` (3D), `MSDeformAttnPixelDecoder3D` BEV neck, `Mask2FormerOccHead`, and a 9-layer `DetrTransformerDecoder`.

Training hyper-parameters.

- *Optimization.* AdamW with LR = 10^{-4}, weight decay = 10^{-2}; gradient clipping with `max_norm`= 20.
- *Schedule.* StepLR with milestones at epochs $[20, 25]$; total 30 epochs.
- *Batching.* `samples_per_gpu`= 4 (vs. 1 on KITTI [4]) and `workers_per_gpu`= 8, enabled by the smaller grid.
- *Losses.* Loss weights follow the baseline: `loss_cls`= 2.0, `loss_mask`= 5.0, `loss_dice`= 5.0. We keep class weighting consistent with the 7-class setup (background down-weighted).

I/O and evaluation. Input resolution is $(384, 1280)$; we evaluate on the official QuadOcc validation split (KITTI [4] baseline uses `test`). Unless noted, we report single-model, single-scale results without test-time augmentation.

Design rationale. *(i) Symmetric near-field bounds.* QuadOcc emphasizes a 25.6×25.6×3.2 m volume; centering and halving KITTI's [4] bounds removes pinhole-centric bias and avoids wasting tokens outside the useful radius. *(ii) Dataset-specific normalization.* Quad statis-

Table 17. **KITTI [4] vs. QuadOcc (OccFormer [18]).**

Component	KITTI [4] baseline	QuadOcc (ours)	Rationale
Grid size (GT)	$256\times256\times32$	$64\times64\times8$	Near-field focus; compute budget
Voxel size	0.2 m	0.4 m	Match smaller radius, reduce tokens
Point-cloud range	$[0, -25.6, -2, 51.2, 25.6, 4.4]$	$[-12.8, -12.8, -1.2, 12.8, 12.8, 2.0]$	Symmetric bounds for panoramas
Classes	20	7	Quad taxonomy (coarser labels)
Camera	left (pinhole)	images_unfold (panorama)	Fit ER/unfolded panoramas
Batch size	1	4	Lower res ⇒ higher throughput

tics align the unfolded ER appearance with the backbone's expectations and improve stability. *(iii) Minimal deltas.* We keep the architecture intact (EffNet-B7 [22] → Mask2FormerOccHead [20]) to avoid capacity confounds; changes are isolated to data, bounds, taxonomy, and sampling so that comparisons remain fair and reproducible.

H3O. Data and taxonomy. H3O provides omnidirectional RGB and voxelized semantic occupancy within $X{=}Y{=}\pm\,12.8\,\text{m}$ and $Z \in [-2.4, 0.8]\,\text{m}$, discretized at $0.4\,\text{m}$ into a native $64\times64\times8$ grid. We adopt the 11-class remap with `empty` at index 0 and non-empty: {`road`, `sidewalk`, `building`, `vegetation`, `car`, `truck`, `bus`, `two_wheeler`, `person`, `pole`}. Images are panoramic ER (equidistant cylindrical) frames; normalization uses H3O statistics (mean $[123.971, 123.564, 164.785]$, std $[84.271, 88.170, 72.966]$).

Architecture/data adaptations.

- *Camera/input.* `camera_used` is switched from `['left']` (KITTI [4]) to `['panorama_rgb']` (ER panorama).
- *View transformer.* We replace the pinhole version with `ViewTransformerLiftSplatShootVoxelH3O` and its ER-aware geometry routine `get_geometry_h3o`, which maps pixel (u, v) to azimuth–elevation (θ, ϕ) (*e.g.*, $\theta = 2\pi(u/W - 0.5)$, $\phi = \pi\,(v/H)$) and then to 3D rays before lifting and splatting.
- *Spatial bounds and resolution.* `point_cloud_range`$=[-12.8, -12.8, -2.4, 12.8, 12.8, 0.8]$, `occ_size`$=[64, 64, 8]$ (voxel size $0.4\,\text{m}$). Symmetric near-field bounds replace KITTI's [4] forward-biased setup.
- *Pipelines.* Training uses `CreateDepthFromLiDAR(dataset='h3o')` to provide LSS [19] depth supervision and `LoadH3OAnnotation`; evaluation uses `LoadH3OAnnotation`.

Components (kept as baseline). `CustomEfficientNet-B7` backbone, `SECONDFPN` image neck, `OccupancyEncoder` (3D), `MSDeformAttnPixelDecoder3D` BEV neck, `Mask2FormerOccHead`, and a 9-layer `DetrTransformerDecoder`.

Training hyper-parameters.

- *Optimization.* AdamW (LR $= 10^{-4}$, weight decay $= 10^{-2}$); gradient clipping with `max_norm`$= 20$.
- *Schedule.* StepLR with milestones at epochs $[20, 25]$; total 30 epochs.
- *Batching and sampling.* `samples_per_gpu`$= 4$, `workers_per_gpu`$= 8$.

I/O and evaluation. Input resolution is $(640, 1280)$; we evaluate on the official `val` split of H3O and select `save_best='h3o_SSC_mIoU'`. We report single-model, single-scale results without test-time augmentation.

Design rationale. *(i) ER-aware lifting.* The equidistant cylindrical model avoids systematic azimuthal bias and preserves uniform horizontal sampling; `get_geometry_h3o` converts (u, v) to (θ, ϕ) and then to 3D rays for accurate lift–splat. *(ii) Symmetric bounds at small radius.* Centered $25.6\times25.6\times3.2\,\text{m}$ is better matched to 360° human-scale scenes than KITTI's [4] forward-biased frustum. *(iii) Minimal deltas.* By keeping backbone, necks, encoder, and head identical to the baseline, improvements (or failures) can be attributed to geometry/data adaptations rather than capacity changes.

6.6. LMSCNet

Scope. We follow LMSCNET [8]'s original 3D CNN design and keep the architecture intact (multi-scale encoder–decoder for semantic scene completion). To adapt to panoramic inputs and the native small voxel grid in QuadOcc, we only modify channel widths and the loss, and plug in the Quad dataset/config. All models are trained from scratch with identical depths and kernels as the baseline.

QuadOcc. Data and taxonomy. QuadOcc provides single-frame semantic occupancy within $X{=}Y{=}\pm\,12.8\,\text{m}$ and $Z \in [-2.0, 1.2]\,\text{m}$, discretized at $0.4\,\text{m}$ into a native $64\times64\times8$ grid (*implementation layout*: $(W, H, D){=}(64, 8, 64)$). We train on 7 labels (index 0 is `empty`; non-empty: {`vehicle`, `person`, `road`, `building`, `vegetation`, `terrain`}).

LiDAR input and alignment. Unlike monocular settings, our LMSCNET [8] input is the *real*, time-synchronized Livox Mid360 scan captured together with the PAL panoramic camera. Each scan is transformed to the benchmark frame using the calibrated rigid extrinsics $T_{\text{cam}\leftarrow\text{lidar}} \in SE(3)$: $\mathbf{X}_{\text{cam}} = T_{\text{cam}\leftarrow\text{lidar}}\mathbf{X}_{\text{lidar}}$. We then *crop* to the

Table 18. **KITTI [4] vs. H3O (OccFormer [18]).**

Component	KITTI [4] baseline	H3O (ours)	Rationale
Grid size (GT)	$256\times256\times32$	$64\times64\times8$	Small near-field; computation budget
Voxel size	0.2 m	0.4 m	Match smaller radius; reduce tokens
Point-cloud range	$[0, -25.6, -2, 51.2, 25.6, 4.4]$	$[-12.8, -12.8, -2.4, 12.8, 12.8, 0.8]$	Symmetric bounds for 360° panoramas
Camera model	pinhole (left)	ER panorama	Correct geometry for 360° input
View transformer	LSS-voxel (pinhole)	LSS-voxel (ER)	Azimuth/elevation → 3D rays
cam_channels	33	27	Match H3O camera metadata
Batch size	1	4	Lower res ⇒ higher throughput

QuadOcc volume $X{=}Y{=}\pm\,12.8\,\text{m}$, $Z\in[-2.0,\,1.2]\,\text{m}$ and *voxelize* at $0.4\,\text{m}$ to produce a native $64\times64\times8$ occupancy tensor. The origin and axes match the ground-truth convention (origin $[-12.8, -12.8, -1.2]\,\text{m}$), so that voxel indices align *exactly* with the GT occupancy region. Out-of-bounds points are discarded; extrinsics are applied consistently across all scans in a sequence. This setup eliminates projection drift, ensures geometric consistency with the PAL camera, and provides a strong LiDAR-aligned prior for semantic scene completion.

Architecture adaptations.

- *Channel expansion (core).* Let C_{in} denote the input feature channels at the encoder entrance. We expand the base width to $f=8\,C_{\text{in}}$ (baseline uses $f=C_{\text{in}}$). The first encoder block is updated to `Conv2d`$(C_{\text{in}}\to f)\to$`ReLU`$\to$`Conv2d`$(f\to f)\to$`ReLU`. At the decoder end, the fusion layer `conv1_1` maps the concatenated multi-scale features back to C_{in} instead of f to keep the output head unchanged. (We tried $2\times$ widening; it underperformed, hence the $8\times$ choice.)
- *Loss.* We use `CrossEntropyLoss` for the 1:1 scale.
- *Logit alignment.* The decoder produces logits that are voxel-for-voxel aligned with the native $64\times64\times8$ GT grid (our tensors follow (W, H, D) with $H{=}8$).

Training hyper-parameters. Unless otherwise stated, we keep the official recipe and change only what is necessary for QuadOcc:

- *Dataset/config.* `QuadSSC`, grid $(W, H, D) = (64, 8, 64)$, `num_classes`$=7$, `FLIPS`$=$`true`.
- *Optimization.* Adam [23] (LR $=10^{-3}$, $\beta{=}(0.9, 0.999)$), no weight decay; power-iteration LR schedule per epoch with `LR_POWER`$=0.98$.
- *Batching.* `train_bs`$=8$, `val_bs`$=8$, `num_workers`$=8$.
- *Schedule.* `epochs`$=80$.

Design rationale. *(i) Width-for-depth trade-off.* QuadOcc collapses the vertical axis to $H{=}8$ bins; naive downscaling severely reduces 3D capacity. Widening the base to $8\times$ restores representational power without changing depth or kernel shapes. *(ii) Input-aware widening.* Starting from $C_{\text{in}}\to f$ lets the network capture richer low-level cues that are otherwise lost on the shallow vertical dimension, while mapping back to C_{in} preserves the original head. *(iii) Minimal deltas, consistent I/O.* We keep depths/kernels unchanged, align logits exactly to the native grid, and modify only widths, thus ensuring fairness and reproducibility.

6.7. SSCNet

Scope. We follow SSCNET [24]'s fully 3D CNN design (encoder–decoder with multi-scale skip connections) and keep depths, kernel sizes, and heads unchanged. To fit QuadOcc's panoramic near-field grid, we minimally modify the upsampling head and the loss/balancing, and we plug in the Quad dataset branch. All models are trained from scratch.

QuadOcc. Data and taxonomy. QuadOcc provides single-frame voxelized semantic occupancy within $X{=}Y{=}\pm\,12.8\,\text{m}$ and $Z\in[-2.0,\,1.2]\,\text{m}$, discretized at $0.4\,\text{m}$ into a native $64\times64\times8$ grid (*implementation layout*: $(W, H, D) = (64, 8, 64)$). We adopt the Quad taxonomy with `num_classes`$=7$ for occupied voxels; *free* voxels are encoded as label `255` and *ignored* by the CE loss (thus not counted as a learnable class).

LiDAR input and alignment. Our SSCNET [24] and SSCNET-FULL [8] variant consumes the *real*, time-synchronized Livox Mid360 scan acquired together with the PAL panoramic camera. Each scan is rigidly transformed to the benchmark/camera frame using calibrated extrinsics $T_{\text{cam}\leftarrow\text{lidar}}\in SE(3)$: $\mathbf{X}_{\text{cam}} = T_{\text{cam}\leftarrow\text{lidar}}\mathbf{X}_{\text{lidar}}$. We then *crop* to the QuadOcc volume $X{=}Y{=}\pm\,12.8\,\text{m}$, $Z\in[-2.0,\,1.2]\,\text{m}$ and *voxelize* at $0.4\,\text{m}$ to form a native $64\times64\times8$ occupancy tensor (implementation layout $(W, H, D) = (64, 8, 64)$). The origin and axes follow the ground-truth convention (origin $[-12.8, -12.8, -1.2]\,\text{m}$), so that voxel indices align *exactly* with the GT occupancy region. Out-of-bounds points are discarded, and the same extrinsics are applied consistently across a sequence. The resulting single-channel free/occupied grid serves as the network input, while the CE loss supervises the 7 semantic classes with free voxels ignored (`label`$=255$). This setup removes projection drift, enforces geometric consistency with the PAL rig, and provides a strong LiDAR-aligned prior for semantic scene completion.

Architecture adaptations.

- *Deconvolution head → Upsample + Conv*

Table 19. **KITTI** [4] **baseline vs. QuadOcc (LMSCNet** [8]**).**

Component	KITTI [4] baseline	QuadOcc (ours)	Rationale
Grid size (GT)	$256\times256\times32$	$64\times64\times8$	Small near-field grid
Voxel size	0.2 m	0.4 m	Memory/compute budget
Base width f	C_{in}	$8\,C_{\text{in}}$	Recover capacity lost by small $H{=}8$
Enc. first conv	$f\to f$	$C_{\text{in}}\to f$	Widen from the input stage
Dec. last conv	$\cdot\to f$	$\cdot\to C_{\text{in}}$	Keep head I/O unchanged
Batch size	2–4	8	Smaller grid $\Rightarrow$ higher throughput
Epochs	60–80	80	Converge under widened channels

(stability fix). The baseline uses a transposed convolution to recover the native grid (`ConvTranspose3d(128→C, k=4, s=4)`). We replace it with `Upsample(scale_factor=4, mode='nearest')` followed by `Conv3d(128→C, k=3, s=1, p=1)`. This avoids loss-time shape/NaN issues while preserving the target stride and receptive field.

- *Logit alignment.* The decoder outputs voxel-aligned logits at the native $64\times64\times8$ resolution (our tensors follow (W, H, D) with $H{=}8$).

Training hyper-parameters.

- *Scene/grid.* $(W, H, D) = (64, 8, 64)$, voxel size 0.4 m, `num_classes`= 7, free= 255 (ignored).
- *Optimization.* Adam [23] (LR $= 10^{-3}$, $\beta = (0.9, 0.999)$), no weight decay; constant LR schedule.
- *Batching.* `train_bs`= 8, `val_bs`= 8, `num_workers`= 8.
- *Schedule.* `epochs`= 80.

Design rationale. *(i) Numerically stable upsampling.* Replacing deconvolution with Upsample + Conv preserves the $\times4$ stride yet eliminates checkerboard artifacts and loss-time instability observed with transposed convolutions on small $H{=}8$ grids. *(ii) Minimal deltas, aligned outputs.* Keeping the encoder/decoder unchanged and aligning logits exactly to the native grid ensures that performance differences come from data/label semantics and small-grid geometry rather than capacity shifts.

6.8. OccRWKV

Scope. We follow the original OccRWKV [25] pipeline (point-cloud preprocessing $\to$ BEV UNet branch $\to$ 3D completion branch, with RWKV-style recurrent blocks) and keep depths/kernels unchanged. To fit panoramic near-field grids in QuadOcc, we introduce Quad-specific modules, LiDAR-driven inputs, explicit voxel coordinate ordering, and a simplified loss. All models are trained from scratch.

QuadOcc. Data and taxonomy. QuadOcc provides single-frame voxelized semantic occupancy within the near-field bounds $X \in [0,\ 25.6]$ m, $Y \in [-12.8,\ 12.8]$ m, $Z \in [-1.2,\ 2.0]$ m, discretized at 0.4 m into a native $64\times64\times8$ grid. We adopt 7 semantic classes (non-empty) and treat free space as `255` (ignored by CE).

LiDAR input and alignment. Our OccRWKV [25] variant consumes the *real*, time-synchronized Livox Mid360 scan captured together with the PAL panoramic camera. Each scan is rigidly transformed to the benchmark/camera frame via calibrated extrinsics $T_{\text{cam}\leftarrow\text{lidar}} \in SE(3)$: $\mathbf{X}_{\text{cam}} = T_{\text{cam}\leftarrow\text{lidar}}\mathbf{X}_{\text{lidar}}$. We *crop* to the QuadOcc volume, *voxelize* at 0.4 m to obtain the native $64\times64\times8$ occupancy tensor, and align the origin/axes to the GT convention. Out-of-bounds points are discarded, and the same extrinsics are applied consistently per sequence.

Architecture/data adaptations.

- *LiDAR-driven inputs.* The network ingests point clouds from `PCD['1_1']` and dense occupancy from `3D_OCCUPANCY`; labels are read from `3D_LABEL['1_1']`.
- *Occupancy +BEV fusion.* We treat the $D{=}8$ occupancy slices as channels ($B\times D\times H\times W$) and concatenate them with BEV features ($B\times C_{\text{bev}}\times H\times W$) along the channel dimension, yielding a $B\times(D{+}C_{\text{bev}})\times H\times W$ tensor for the 2D BEV UNet.
- *Loss.* We remove auxiliary semantic-point loss and keep a *single* voxel-wise objective: class-weighted cross-entropy (CE) with `ignore_index=255` plus `Lovasz-Softmax`. Final logits are permuted to (B, C, W, D, H) as required by the head before loss.

Training hyper-parameters.

- *Dataset/config.* `QuadOcc`; grid $[W, H, D] = [64, 64, 8]$; voxel size 0.4 m; bounds as above; `FLIPS=true`; modalities: `PCD`, `3D_OCCUPANCY`, `3D_LABEL`.
- *Optimization.* Adam [23] (LR $= 10^{-3}$, $\beta = (0.9, 0.999)$), no weight decay; power-iteration schedule per epoch with `LR_POWER`= 0.98.
- *Batching/schedule.* `batch_size`= 2, `num_workers`= 8, `epochs`= 80.

Design rationale. *(i) Simplified loss.* Dropping the auxiliary semantic segmentation loss branch focuses capacity and stabilizes training; adding Lovasz complements CE on class-imbalanced voxels. *(v) Minimal deltas.* Network

Table 20. **KITTI [4] baseline vs. QuadOcc (SSCNet-full [24]).**

Component	KITTI [4] baseline	QuadOcc (ours)	Rationale
Grid size (GT)	$256\times256\times32$	$64\times64\times8$ (impl. $(64, 8, 64)$)	Small near-field volume
Voxel size	0.2 m	0.4 m	Memory/compute budget
Upsampling head	ConvTranspose3d ($\times4$)	Upsample $\times4$ + Conv3d	Avoid NaNs/shape issues; same stride
Labels (free)	0 (free), occupied > 0	255 (free), occupied < 255	Match Quad label semantics
Batch size	4	8	Lower res $\Rightarrow$ higher throughput
Epochs	60–80	80	Convergence at native width/depth

Table 21. **KITTI [4] baseline vs. QuadOcc (OccRWKV [25]).**

Component	KITTI [4] baseline	QuadOcc (ours)	Rationale
Grid size	$256\times256\times32$	$64\times64\times8$	Near-field, low-memory grid
Voxel size	0.2 m	0.4 m	Token reduction for panoramic scenes
Bounds (X/Y/Z)	$[0, 51.2]/-25.6{:}25.6/-2{:}4.4$	$[0, 25.6]/-12.8{:}12.8/-1.2{:}2.0$	Match Quad near-field convention
Loss	CE + aux. seg	CE (w/ class weights) + Lovasz	Stable single-head supervision
Batch size	2	2	Compute budget
Epochs	80	80	Same convergence horizon

depths/widths are unchanged; improvements (or failures) can thus be attributed to geometry/data choices rather than capacity changes.

7. Discussions

7.1. Limitations and Potential Solutions

Spatial resolution and fine-grained interaction. Our semantic occupancy is defined on a $64\times64\times8$ grid with 0.4 m voxels around the ego, which is sufficient for navigation, foothold selection, and path planning on legged/humanoid platforms with moderate speed and limited payload compared to intelligent vehicles. However, this resolution is not ideal for tasks that require fine-grained contact reasoning, such as precise grasping, object re-arrangement, or manipulation in cluttered shelves. At this scale, small objects and thin structures are often represented by only a few voxels, which amplifies label noise and makes it difficult to capture geometry with sub-voxel accuracy.

A straightforward remedy is to increase the grid resolution (*e.g.*, $128\times128\times16$ under the same metric bounds), but our resolution study shows that this leads to a significant increase in memory and latency, and can even hurt mIoU due to optimization difficulty. Instead, a more promising direction is *hierarchical* or *adaptive* occupancy. One option is to maintain a coarse global grid (as in OneOcc) for full-surround situational awareness, while allocating high-resolution local volumes around the end-effectors (hands and feet) or task-relevant regions selected by an attention or proposal mechanism. Another option is to decouple geometry and semantics: a coarse semantic grid can be refined locally by an implicit or Gaussian-based representation [26] that super-resolves surfaces where higher accuracy is required. Both strategies preserve the efficiency and robustness of the current pipeline while opening the door to fine-grained interaction.

Dataset coverage and sim-to-real gap. QuadOcc is moderate-scale and rooted in a campus-like environment, and Human360Occ (H3O) is purely simulated. While this pairing is useful for studying cross-domain robustness and robot morphology changes, it does not fully cover the diversity of real humanoid deployment scenarios (*e.g.*, dense indoor offices, homes, or highly dynamic crowds). In particular, our humanoid-style evaluation currently relies on simulated occupancy labels and thus inherits the biases of the simulator and rendering stack. This introduces an inevitable sim-to-real gap for humanoid robots with panoramas.

Mitigating this gap will require a combination of larger-scale real-world 360° occupancy datasets (*e.g.*, panoramic rigs mounted on humanoids or human operators) and sim-to-real adaptation techniques. Possible options include domain randomization and style augmentation in the panoramic image space, occupancy-level adversarial training between simulated and real volumes, and semi-supervised self-training where the model bootstraps dense occupancy from sparse or partial real sensors. Another complementary direction is to leverage weak labels, such as traversability or contact affordances, as auxiliary supervision for regions where dense voxel labels are missing.

Single-frame design and motion modeling. Our current formulation predicts occupancy from a single panoramic frame. This design choice avoids hard assumptions on the robot's motion model, simplifies deployment, and is already robust to gait-induced jitter via GDC. However, it also limits the ability to accumulate evidence over time, to explicitly reason about dynamic objects, and to exploit long-term temporal context. In particular, fast-moving agents (pedes-

trians, vehicles, other robots) are treated as static at each frame, and any temporal consistency emerges only implicitly from the training data.

A natural extension is to move from single-frame inputs to short panoramic clips, and to lift them into a 4D spatio-temporal occupancy volume. Architecturally, this could be realized by adding a recurrent [27] or transformer-style temporal module on top of OneOcc, or by introducing causal 3D/4D convolutions over occupancy sequences. To keep the embodied footprint manageable, one promising strategy is to maintain a low-rate, wide-range occupancy memory complemented by higher-rate local updates near the robot, rather than processing dense video at full resolution.

7.2. Failure Case Analysis

Figure 12 illustrates a representative failure case from the challenging H3O-Heter split, where the robot traverses a corridor that is almost entirely covered by vegetation and spans multiple maps under the cross-city configuration. This scene is further compounded by adverse conditions: it is captured at dusk, in rain, with low illumination and strong specular highlights on wet surfaces. While our training data already covers diverse weather and lighting conditions, sequences that are simultaneously *vegetation-dominated*, *cross-map*, and *rainy at dusk* are extremely rare, and most trajectories follow road- or sidewalk-centric layouts in clearer visibility. As a result, all methods face a substantial distribution shift in both geometry and appearance.

In this example, the ground-truth occupancy assigns the majority of near-field voxels to *vegetation* (class 4, green), and virtually no large dynamic objects such as *car* (class 5), *truck* (class 6), or *bus* (class 7). In contrast, vision-based baselines (*e.g.*, MonoScene [3], SGN-S [16]) hallucinate extensive road and sidewalk regions in front of the agent and along the sides. These errors reflect a strong reliance on co-occurrence priors learned from more typical urban scenes: given a forward-looking corridor with lane-like edges and headlight-like highlights, the models default to "road + sidewalk" even when the geometry and appearance actually correspond to wet vegetation. OneOcc substantially reduces these artifacts: the free/occupied structure is better aligned with the ground truth, and most spurious roads are removed, yet it still misclassifies sizable vegetation regions as sidewalk, leading to severe semantic errors despite reasonably accurate geometry.

This failure pattern highlights two important insights. First, even with improved architectural bias and stronger cross-city robustness, semantic occupancy prediction remains fundamentally data-driven: models inherit biases from the training distribution and struggle when encountering rare combinations of layout, weather, and time-of-day. Closing this gap will likely require larger-scale and more diverse 360° occupancy datasets that explicitly target under-represented environments (*e.g.*, forests, parks, off-road trails) under diverse conditions (night, adverse weather, seasonal changes), as well as more balanced class statistics. Second, the dominant error here is *semantic* rather than geometric: the model correctly infers that space is occupied, but assigns the wrong category. This suggests that future *open-vocabulary occupancy* [28] formulations, where volumetric geometry is coupled with flexible semantic embeddings instead of a fixed closed set of labels, could be beneficial. By leveraging open-set vision-language representations, such models may better adapt their semantic partition of the occupancy volume to novel environments (for example, distinguishing different types of vegetation or terrain) while reusing the same geometric backbone as OneOcc. Combined with stronger data diversity, such open-vocabulary occupancy could mitigate, though likely not completely eliminate, the kind of semantic failure observed in this case.

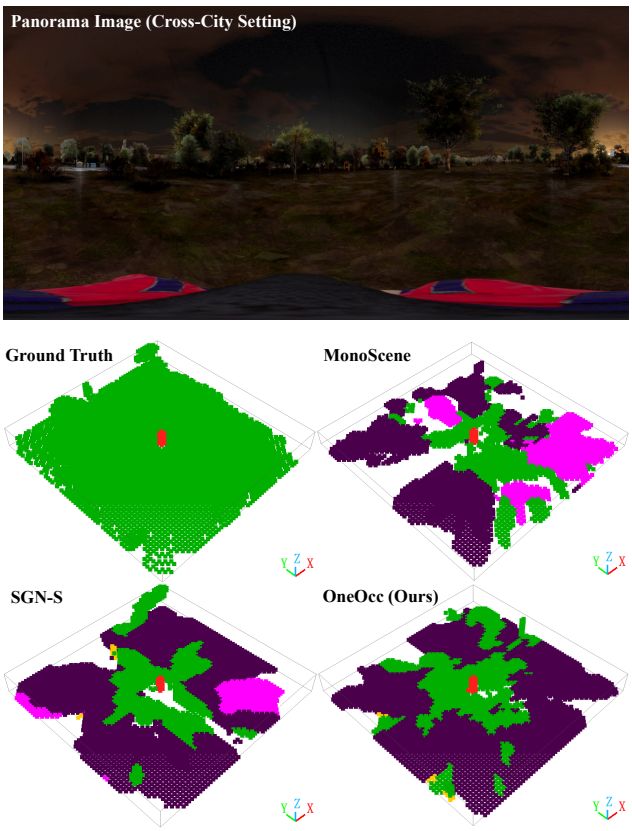

Figure 12. **Failure case on H3O-Heter (cross-city) in a rainy dusk scene dominated by vegetation.** From left to right: ground-truth occupancy, predictions from MonoScene [3], SGN-S [16], and OneOcc (Ours). OneOcc alleviates but does not eliminate severe semantic misclassification (vegetation predicted as sidewalk) under this rare combination of layout, weather, and time-of-day.

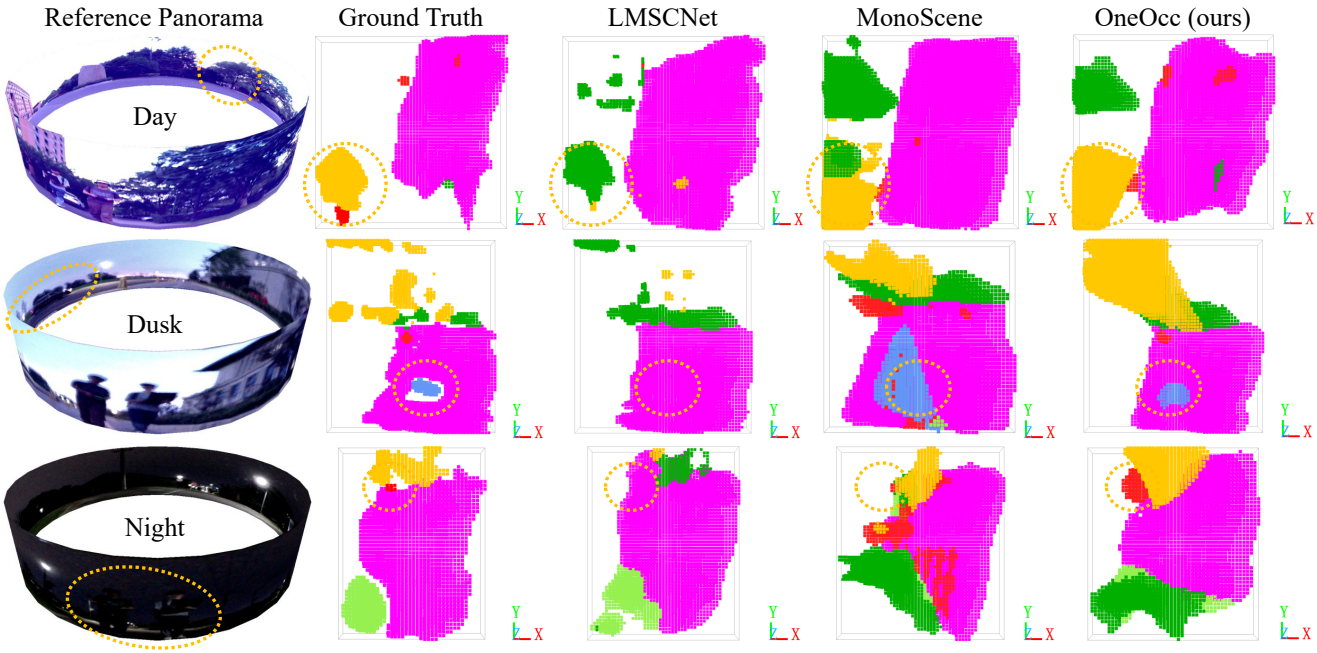

Figure 13. **Qualitative comparison on QuadOcc under different times of day.** From left to right: reference panorama, ground-truth occupancy, LMSCNet [8], MonoScene [3], and OneOcc (ours). From top to bottom: day, dusk, and night. All methods gradually degrade from day to dusk and further to night, where low illumination and glare introduce severe ambiguities. Compared to baselines, OneOcc reduces perspective sampling artifacts and yields more coherent occupied structures and semantics, especially for dynamic classes such as vehicles and pedestrians.

7.3. Range-wise Safety Analysis

To further examine whether the gains of OneOcc are confined to a particular depth interval or remain valid in safety-critical local regions, we report distance-binned mIoU comparisons against MonoScene [3] on both QuadOcc and H3O-Heter. Following the same evaluation protocol as the main paper, we restrict non-empty voxels to three nested spatial ranges centered at the agent: *Far* ($\pm$12.8m), *Mid* ($\pm$6.4m), and *Near* ($\pm$3.2m).

Figure 14 shows that OneOcc consistently outperforms MonoScene across all distance bins on both benchmarks. On QuadOcc, the mIoU improves from 19.19 to 20.56 in the Far range, from 23.32 to 24.47 in the Mid range, and from 24.29 to 24.95 in the Near range. On H3O-Heter, the corresponding gains are larger: 24.15 $\rightarrow$ 32.23 (Far), 30.17 $\rightarrow$ 37.11 (Mid), and 30.69 $\rightarrow$ 37.35 (Near). These results indicate that the benefits of OneOcc are not limited to a specific distance regime, but remain stable from long-range completion to local occupancy reasoning. From a robotics perspective, the Near range is especially important because it is most directly related to short-horizon collision avoidance, foothold selection, and local traversability. Importantly, OneOcc still maintains positive margins in this region on both datasets: +0.66 mIoU on QuadOcc and +6.66 mIoU on H3O-Heter. This suggests that the proposed

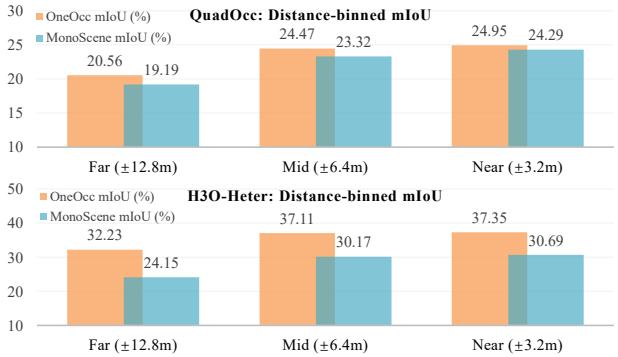

Figure 14. Distance-binned mIoU comparisons on QuadOcc and H3O-Heter. We evaluate non-empty voxels within three nested spatial ranges centered at the agent: Far ($\pm$12.8m), Mid ($\pm$6.4m), and Near ($\pm$3.2m). OneOcc consistently outperforms MonoScene [3] across all ranges on both benchmarks, including the near-field region that is most relevant to local navigation and foothold safety.

design does not merely improve far-field continuity or suppress panoramic artifacts at a global scale, but also provides more reliable local semantic occupancy in the region most relevant to embodied safety. A second trend is that both methods generally improve from Far to Near. This is expected, since smaller spatial ranges contain less severe long-

range ambiguity, fewer occlusion-induced completion errors, and denser effective evidence around the agent. However, the gap between OneOcc and MonoScene remains consistently positive throughout. This supports the claim that the gains of OneOcc arise from better geometry-aware lifting and panoramic representation, rather than from overfitting to a narrow subset of voxels or to only far-range context. Overall, the distance-binned analysis complements the main benchmark tables by showing that OneOcc improves semantic occupancy prediction not only globally, but also in the near-field regime that is most critical for safe embodied deployment.

7.4. Scaling to Other Scenarios and Modalities

From legged to diverse robot morphologies. Although QuadOcc and H3O are designed for quadruped and humanoid-like settings, the core architectural ideas in OneOcc — dual-projection fusion, bi-grid voxelization, and lightweight 3D decoding — are not specific to a particular robot morphology. For wheeled platforms with forward-biased sensing, the same framework can be used with asymmetric voxel bounds that allocate more range in the forward direction while preserving 360° continuity, or with cropped panoramas when only 180° coverage is available. For aerial robots or platforms with elevated cameras, the vertical bounds of the voxel grid can be expanded to better capture tall structures and multi-level environments.

Indoor, outdoor, and extreme conditions. Our datasets focus on outdoor campus-style and urban-like scenes. Deployments in cluttered indoor environments introduce different statistics: shorter ranges, denser occlusions, and more small-scale objects. Scaling OneOcc to such scenarios mainly requires reconfiguring the voxel bounds and semantics, and retraining on appropriate indoor 360° data. Meanwhile, operation under extreme conditions (severe rain/snow, low light, lens contamination) may benefit from explicit robustness techniques such as test-time adaptation [7], uncertainty-aware occupancy heads, or sensor health monitoring that can gracefully degrade occupancy outputs when the panoramic signal is unreliable.

Multi-modal extensions. OneOcc is intentionally vision-only, which is attractive for platforms with strict payload and power budgets. Nevertheless, in settings where additional sensors are available, the proposed architecture can serve as a backbone for multi-modal occupancy. LiDAR or depth maps can be encoded into 2D or 3D features and fused with the panoramic features before or after bi-grid voxelization, while radar and event cameras [29] can complement the panoramic input with long-range or high-dynamic-range motion cues. A promising direction is to treat each modality as an additional "view" in the View2View sampling scheme, letting the model learn how to combine them into a unified 3D grid without hard-coded fusion rules.

7.5. Societal Impacts

Dense 3D semantic occupancy opens up positive applications for legged and humanoid robots. More reliable 360° perception can improve safety when robots operate in proximity to humans, reduce collisions with obstacles in cluttered environments, and enable new capabilities in search-and-rescue, inspection, and assistive robotics. In such contexts, a geometry-aware volumetric representation is often more interpretable and task-aligned than raw images, and can form a safer intermediate layer for downstream planning or language-conditioned policies.

At the same time, always-on full-surround perception could raise some societal questions. A robot equipped with a panoramic camera and a strong occupancy predictor could be used for pervasive surveillance or long-term monitoring of public or semi-private spaces. While our representation focuses on occupancy and coarse semantics rather than identity, it still encodes where humans and vehicles are likely to be, and could be combined with other modules to infer more sensitive attributes. Moreover, our datasets are biased toward specific geographies, weather patterns, and human behaviors, which may lead to failure modes or unfair performance in underrepresented environments. Responsible deployment will require transparency about sensing capabilities, adherence to local privacy regulations, careful consideration of where such robots are allowed to operate, and continued auditing of failure cases, especially near vulnerable populations.

7.6. Future Work

Beyond the immediate extensions discussed above, we see three main research directions.

Temporal occupancy, flow, and world models. First, moving from static occupancy to temporal occupancy sequences is a key step toward *world models* [30–32] for legged and humanoid robots. Instead of predicting only the current occupancy, one can learn to forecast future volumes conditioned on past panoramas and robot actions, or to infer an *occupancy flow* [33, 34] that encodes voxel-wise 3D motion over time. Such a flow field would provide a sparse, geometry-aware analogue of scene flow, tailored to dynamic obstacles and multi-contact locomotion. Training such models will likely require a mix of supervised signals (from synthetic data or reconstructed trajectories) and self-supervised objectives that enforce temporal consistency and physical plausibility.

Adaptive resolution and manipulation-centric perception. Second, we plan to investigate adaptive-resolution

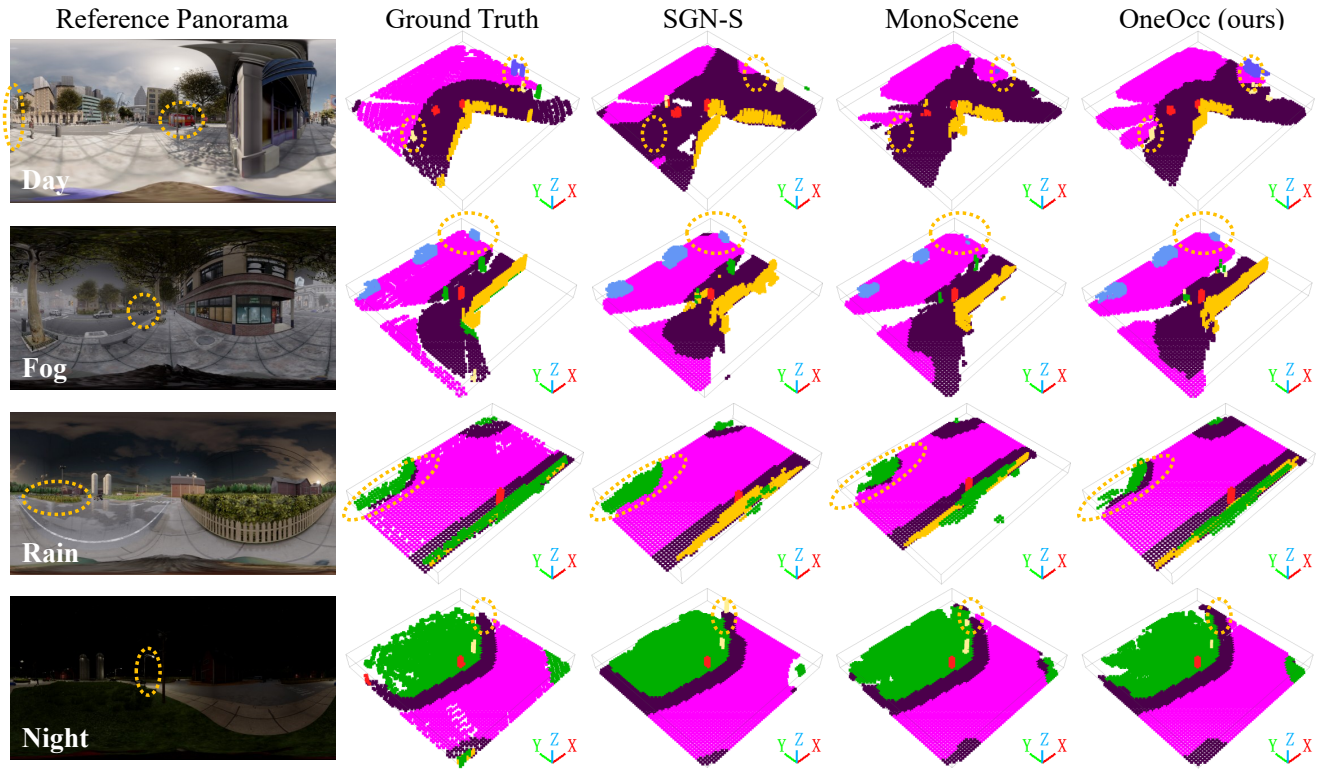

Figure 15. **Qualitative comparison on H3O-Heter under diverse weather and illumination.** From left to right: reference panorama, ground-truth occupancy, SGN-S [16], MonoScene [3], and OneOcc (ours). From top to bottom: clear day, fog, rain, and night. The heter split introduces strong cross-map distribution shift and long-tailed conditions; the performance of all methods degrades under fog/rain and becomes the most brittle at night. Notably, the circled distant lamp post is outside the OCC sensing range in the ground truth (the Night row). SGN-S and MonoScene overestimate their depth and mistakenly project it into the occupancy volume, whereas OneOcc avoids this long-range depth hallucination and yields fewer false positives, while still recovering thin structures (*e.g.*, poles) within range.

schemes that bridge the gap between navigation-centric occupancy and manipulation-centric understanding. This includes foveated occupancy around hands and feet, dynamic refinement of regions with high uncertainty or contact likelihood, and hybrid explicit–implicit models where a coarse grid is locally refined into meshes or signed distance fields for grasp synthesis and force reasoning. A tight coupling between such perception modules and whole-body control could enable humanoid robots to seamlessly transition from long-range navigation to precise object interaction.

Data, generalization, and policy integration. Finally, scaling to broader real-world deployment will require richer data and tighter integration with decision-making. On the data side, we advocate for collecting real 360° occupancy datasets on humanoid and other platforms, with diverse environment types and behaviors, and for exploring self-supervised pretraining on unlabeled panoramic video. On the policy side, occupancy grids can act as an intermediate tokenization layer for vision-language-action models and reinforcement learning policies, allowing language instructions and low-level control to be grounded in a common geometric space. Jointly training such policies and occupancy predictors, while preserving modularity and interpretability, is an exciting avenue toward more capable and trustworthy embodied agents.

8. More Visualizations

Figures 13 and 15 provide additional qualitative comparisons between OneOcc and representative vision-based SSC baselines under diverse illumination and weather conditions. On QuadOcc, we report results for daytime, dusk, and nighttime scenes. Across all methods, the prediction quality degrades noticeably as the illumination decreases: daytime scenes yield the cleanest geometry and semantics, dusk introduces stronger ambiguity around boundaries and small objects, and nighttime remains consistently the most challenging setting due to low signal-to-noise ratio, headlight glare, and reduced texture cues. In particular, existing perspective-based lifting methods [3] often suffer from pronounced projection artifacts and over-smoothed structures at dusk and night, leading to inconsistent free/occupied pat-

terns and spurious semantic regions (*e.g.*, hallucinated road-/sidewalk blobs). OneOcc exhibits improved robustness along this axis: by leveraging panoramic View2View fusion and bi-grid voxelization, it mitigates the typical perspective-sampling artifacts observed in FLOSP-style pipelines, and preserves more coherent occupied structures and semantic layouts at dusk and night. Notably, dynamic categories such as cars and pedestrians are more reliably localized, and their occupied footprints are less fragmented compared to MonoScene [3] and LiDAR-based LMSCNet [8].

On H3O-Heter, we further evaluate generalization under cross-map scenes and adverse weather, including clear daytime, fog, rain, and nighttime. Similar to QuadOcc, all methods deteriorate as the environment becomes more visually ambiguous. Fog reduces contrast and texture, causing baselines to miss thin vertical structures and to blur object boundaries; rain introduces specular reflections and appearance shifts that amplify road-centric priors; nighttime again triggers the largest failures. While baselines frequently collapse thin objects into the background or misclassify them into dominant terrain classes, OneOcc retains sharper geometric contours and more faithful semantics across conditions. We additionally observe a characteristic depth overestimation error of prior methods in the heter split. As highlighted in Fig. 15, a distant lamp post (pole) lies beyond the predefined OCC sensing range in the ground truth. However, SGN-S [16] and MonoScene [3] incorrectly estimate its depth and "pull" the pole into the occupancy volume, generating spurious occupied voxels. In contrast, OneOcc does not introduce such false positives, suggesting that the panoramic View2View fusion and bi-grid lifting lead to more calibrated long-range depth reasoning under cross-map distribution shift.

Overall, these results corroborate our quantitative findings: OneOcc maintains stronger structural consistency and semantic fidelity under both illumination and weather shifts, though extreme low-light conditions remain an open challenge for monocular panoramic SSC.

References

[1] Xuweiyi Chen, Wentao Zhou, Aruni RoyChowdhury, and Zezhou Cheng. Point-MoE: Towards cross-domain generalization in 3D semantic segmentation via mixture-of-experts. *arXiv preprint arXiv:2505.23926*, 2025. 2

[2] Yaohua Zha, Tao Dai, Hang Guo, Yanzi Wang, Bin Chen, Ke Chen, and Shu-Tao Xia. Point cloud mixture-of-domain-experts model for 3D self-supervised learning. In *IJCAI*, 2025. 2

[3] Anh-Quan Cao and Raoul de Charette. MonoScene: Monocular 3D semantic scene completion. In *CVPR*, 2022. 2, 8, 9, 10, 18, 19, 27, 28, 30, 31

[4] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *ICCV*, 2019. 3, 18, 19, 20, 22, 23, 24, 25, 26

[5] Hui Zhou, Xinge Zhu, Xiao Song, Yuexin Ma, Zhe Wang, Hongsheng Li, and Dahua Lin. Cylinder3D: An effective 3D framework for driving-scene LiDAR semantic segmentation. *arXiv preprint arXiv:2008.01550*, 2020. 4

[6] Yiming Li, Zhiding Yu, Christopher B. Choy, Chaowei Xiao, José M. Álvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. VoxFormer: Sparse voxel transformer for camera-based 3D semantic scene completion. In *CVPR*, 2023. 4, 19, 21, 22

[7] Hao Shi, Song Wang, Jiaming Zhang, Xiaoting Yin, Guangming Wang, Jianke Zhu, Kailun Yang, and Kaiwei Wang. Offboard occupancy refinement with hybrid propagation for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2025. 4, 29

[8] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. LMSCNet: Lightweight multiscale 3D semantic completion. In *3DV*, 2020. 6, 23, 24, 25, 28, 31

[9] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *IROS*, 2006. 10, 11, 12, 18, 19, 20

[10] Shaohua Gao, Kailun Yang, Hao Shi, Kaiwei Wang, and Jian Bai. Review on panoramic imaging and its applications in scene understanding. *IEEE Transactions on Instrumentation and Measurement*, 2022. 10

[11] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded SAM: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 10, 12

[12] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. FAST-LIO2: Fast direct LiDAR-inertial odometry. *IEEE Transactions on Robotics*, 2022. 10, 12

[13] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *CoRL*, 2017. 14

[14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes

dataset for semantic urban scene understanding. In *CVPR*, 2016. 15

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 18, 19

[16] Jianbiao Mei, Yu Yang, Mengmeng Wang, Junyu Zhu, Jongwon Ra, Yukai Ma, Laijian Li, and Yong Liu. Camera-based 3D semantic scene completion with sparse guidance network. *IEEE Transactions on Image Processing*, 2024. 19, 20, 21, 22, 27, 30, 31

[17] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *CVPR*, 2019. 20, 21

[18] Yunpeng Zhang, Zheng Zhu, and Dalong Du. OccFormer: Dual-path transformer for vision-based 3D semantic occupancy prediction. In *ICCV*, 2023. 22, 23, 24

[19] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In *ECCV*, 2020. 22, 23

[20] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 22, 23

[21] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 22

[22] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 23

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 24, 25

[24] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 24, 26

[25] Junming Wang, Wei Yin, Xiaoxiao Long, Xingyu Zhang, Zebin Xing, Xiaoyang Guo, and Qian Zhang. OccRWKV: Rethinking efficient 3D semantic occupancy prediction with linear complexity. In *ICRA*, 2025. 25, 26

[26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 2023. 26

[27] Hao Shi, Qi Jiang, Kailun Yang, Xiaoting Yin, Huajian Ni, and Kaiwei Wang. Beyond the field-of-view: Enhancing scene visibility and perception with clip-recurrent transformer. *IEEE Transactions on Intelligent Vehicles*, 2025. 27

[28] Zhiyu Tan, Zichao Dong, Cheng Zhang, Weikun Zhang, Hang Ji, and Hao Li. OVO: Open-vocabulary occupancy. *arXiv preprint arXiv:2305.16133*, 2023. 27

[29] Shangwei Guo, Hao Shi, Song Wang, Xiaoting Yin, Kailun Yang, and Kaiwei Wang. Event-aided semantic scene completion. *arXiv preprint arXiv:2502.02334*, 2025. 29

[30] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. OccWorld: Learning a 3D occupancy world model for autonomous driving. In *ECCV*, 2024. 29

[31] Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. GaussianWorld: Gaussian world model for streaming 3D occupancy prediction. In *CVPR*, 2025.

[32] Tuo Feng, Wenguan Wang, and Yi Yang. Gaussian-based world model: Gaussian priors for voxel-based occupancy prediction and future motion prediction. In *ICCV*, 2025. 29

[33] Yili Liu, Linzhan Mou, Xuan Yu, Chenrui Han, Sitong Mao, Rong Xiong, and Yue Wang. Let Occ flow: Self-supervised 3D occupancy flow prediction. In *CoRL*, 2024. 29

[34] Dubing Chen, Jin Fang, Wencheng Han, Xinjing Cheng, Junbo Yin, Chengzhong Xu, Fahad Shahbaz Khan, and Jianbing Shen. ALOcc: Adaptive lifting-based 3D semantic occupancy and cost volume-based flow predictions. In *ICCV*, 2025. 29