Gau-Occ: Geometry-Completed Gaussians for Multi-Modal 3D Occupancy Prediction

Supplementary Material

6. Datasets and Metrics

nuScenes and SurroundOcc-nuScenes. nuScenes is a large-scale autonomous-driving dataset collected in Boston and Singapore that contains over 1,000 urban scenes. Each scene lasts roughly 20 seconds and is captured with six surround-view cameras and one LiDAR sensor; keyframes are annotated at 2 Hz. Following SurroundOcc [47], we adopt dense semantic-occupancy annotations that voxelize the region $[-50, 50]\,\text{m} \times [-50, 50]\,\text{m} \times [-5, 3]\,\text{m}$ with $0.5\,\text{m}$ voxel resolution, assigning one of 18 classes (16 semantic categories, plus `empty` and `unknown`) to every voxel. We use the official nuScenes split (700/150/150 train/val/test) and follow the standard sensor configuration and annotation protocol. Accordingly, for each keyframe our input consists of the synchronized six-camera images and the LiDAR sweep, and the target is the voxelized 3D occupancy ground-truth.

Occ3D-nuScenes. To further evaluate our method under a distinct semantic-occupancy protocol derived from nuScenes, we also consider Occ3D-nuScenes. Following the official Occ3D devkit [41], we construct voxel volumes over $[-40, 40]\,\text{m} \times [-40, 40]\,\text{m} \times [-1, 5.4]\,\text{m}$ at a $0.4\,\text{m}$ resolution, with 17 semantic categories (16 base classes plus a `General Object` class). We use the standard 600/150/150 train/val/test split, totaling 40,000 annotated keyframes. Similar to SurroundOcc-nuScenes, each keyframe provides six surround-view camera images and a single LiDAR scan. However, the differing voxel range, resolution, and label space make Occ3D-nuScenes a complementary benchmark for assessing robustness to changes in occupancy definitions and grid configurations.

KITTI-360. KITTI-360 offers over 320k multi-view images and 100k LiDAR sweeps from long urban drives. We adopt the dense semantic occupancy annotations released by SSCBench-KITTI-360 [24], which provide ground-truth semantic occupancy for 12,865 keyframes across nine sequences with the standard 7/1/1 train/validation/test split. The voxel grid covers $[0, 51.2]\,\text{m} \times [-25.6, 25.6]\,\text{m} \times [-2, 4.4]\,\text{m}$ at a $0.2\,\text{m}$ resolution, with each voxel labeled as one of 19 categories (18 semantic classes plus `empty`). Following common practice, we use only the left-front perspective camera (image_00 subset) of each keyframe together with the corresponding raw LiDAR point cloud as model input.

We adopt standard evaluation metrics for semantic occupancy prediction tasks. Following common practice, we use the **Intersection over Union (IoU)** of all occupied voxels to evaluate the geometry reconstruction performance of the model, and the **mean Intersection over Union (mIoU)** of all semantic classes to evaluate its semantic perception ability. The IoU and mIoU are computed as follows:

$$\text{IoU} = \frac{TP_{c_0}}{TP_{c_0} + FP_{c_0} + FN_{c_0}}, \tag{18}$$

$$\text{mIoU} = \frac{1}{|C|} \sum_{i \in C} \frac{TP_i}{TP_i + FP_i + FN_i}, \tag{19}$$

where c_0 denotes the nonempty (occupied) class; TP_i, FP_i, and FN_i are the number of true positive, false positive, and false negative predictions for class i, C is the set of semantic classes. These metrics jointly provide a comprehensive evaluation of both geometric reconstruction quality and semantic occupancy prediction accuracy.

7. Experimental Setup

LCD pre-training. The *LiDAR Completion Diffuser (LCD)* is pre-trained on dense targets built by ego-motion alignment and accumulation of $K{=}20$ consecutive sweeps. We train LCD for 20 epochs on the respective training split before joint optimization with the proposed Gau-Occ framework. The forward process follows DDPM with $T{=}1000$ steps and a linear schedule $\{\beta_t\}_{t=1}^{T}$ (default $\beta_0{=}3.0{\times}10^{-5}$, $\beta_T{=}7.0{\times}10^{-3}$), $\alpha_t{=}1{-}\beta_t$, $\bar{\alpha}_t{=}\prod_{i=1}^{t}\alpha_i$. During training, we use DPM-Solver sampling with 50 denoising steps. All hyper-parameters above are held fixed across datasets unless stated.

Semantic Gaussians. We instantiate a dataset-specific number of semantic Gaussians: $N_G{=}25{,}600$ for nuScenes and $N_G{=}40{,}000$ for KITTI-360. Hybrid initialization selects centers from the completed cloud $\mathcal{P}'$ via density-based selection and random coverage: the default split is $N_d{:}N_r{=}70\%{:}30\%$. Each new Gaussian has an axis-aligned initial scale $\mathbf{s}_i \sim \mathcal{U}([0.20, 1.00])$ per axis. Local splatting uses a neighborhood radius $R_{\text{geo}}{=}k\,\bar{s}_i$ with $\bar{s}_i{=}\frac{1}{3}(s_x{+}s_y{+}s_z)$ and default $k{=}1.5$.

LiDAR voxel features. We voxelize the completed cloud $\mathcal{P}'$ into a sparse 3D grid (bounds and voxel size as in the main paper) and keep at most 10 points per voxel [48]. Per-voxel features $\mathbf{F}_v$ are obtained by averaging point embeddings $\psi(p)$ to $\mathbf{f}_v^0$ and feeding a sparse 3D CNN encoder that outputs d_{pc}-dimensional descriptors where $d_{pc}{=}128$. For

a Gaussian G_i centered at $\boldsymbol{\mu}_i$ with scale $\mathbf{s}_i{=}(s_x, s_y, s_z)$, we aggregate neighboring voxels within an adaptive radius $R_{\text{geo}}{=}k\,(s_x{+}s_y{+}s_z)/3$ where $k{=}1.5$, using an exponential kernel $w_v{=}\exp(-\gamma\|\mathbf{p}_v{-}\boldsymbol{\mu}_i\|_2)$ (default $\gamma{=}3.0$), yielding the geometry descriptor $\mathbf{f}_{\text{pc},i} \in \mathbb{R}^{d_{pc}}$.

Image backbone and pyramid. Unless otherwise noted, we use ResNet-50 with a 4-level FPN ($L{=}4$) at strides $s_l \in \{4, 8, 16, 32\}$. Each level has channel width $d{=}128$. The number of camera views is dataset-specific: $V{=}6$ for nuScenes and $V{=}1$ for KITTI-360 (image_00). Each anchor predicts $N_{\text{off}}{=}9$ geometry-guided offsets per level/view; sampling radius are $R_l \in \{4, 8, 16, 32\}$ feature pixels . The geometry weight uses $\sigma_l{=}\kappa R_l$ with default $\kappa{=}1.0$.

Geo-VLAD resampler, cross-attention, and update head. Sampled tokens $\mathbf{X}_i \in \mathbb{R}^{N\times d}$ are compressed by a geometry-aware VLAD-style resampler with M codewords $\{\mathbf{C}_m\}_{m=1}^{M}$, where $M{=}32$. Linear maps follow the shapes in the main text. FiLM modulation predicts per-channel (γ_i, β_i) from $\mathbf{f}_{\text{pc},i}$ to rescale/shift the resampled tokens; multi-scale fusion uses learnable non-negative weights $\{\lambda_l\}_{l=1}^{L}$ (softmax-normalized). The Gaussian update head is a two-layer feed-forward network (FFN) with GELU and hidden size 128, regressing $[\widehat{\boldsymbol{\mu}}_i, \widehat{\mathbf{s}}_i, \widehat{\mathbf{r}}_i, \widehat{\mathbf{c}}_i]$; updated Gaussians $\mathbf{G}_i^{\text{new}}{=}(\boldsymbol{\mu}_i{+}\widehat{\boldsymbol{\mu}}_i, \widehat{\mathbf{s}}_i, \widehat{\mathbf{r}}_i, \widehat{\mathbf{c}}_i)$ are then splatted locally to produce O.

Optimization and implementation details. We minimize $\mathcal{L}_{\text{CE}}{+}\mathcal{L}_{\text{Lov}}$ following [14]. AdamW is used with weight decay 0.01; the learning rate warms up linearly for 500 iters to $2{\times}10^{-4}$ and then follows cosine decay to $1{\times}10^{-6}$. Unless specified, training runs for 20 epochs on nuScenes and 25 on KITTI-360 with batch size 8. We implement in PyTorch 1.12.1 (Python 3.9, Ubuntu 22.04). nuScenes experiments are trained/inferred on RTX 4090 (24 GB); KITTI-360 on A100 (40 GB).

8. Model Efficiency

Tab. 4 compares the *latency, memory, accuracy* trade-off of different 3D occupancy prediction pipelines.

For the camera-only baselines at the top of the table, BEVFormer, TPVFormer, and SurroundOcc all rely on dense BEV or volumetric queries, leading to relatively high computational cost, they run at 310 $\sim$ 340 ms with 4.5 $\sim$ 5.9 GB memory, while only achieving around 31 IoU and 16 $\sim$ 17 mIoU. Under the 12.8k-query setting, our model attains 124 ms latency and 3.3 GB memory, which is about $2.5\times$ faster and $27 \sim 44\%$ more memory-efficient than these BEV-based camera-only methods, while delivering much higher IoU and mIoU. Even the higher-parameter 25.6k-query configuration still runs faster than BEV-based camera models (230 ms vs. 310$\sim$340 ms) with comparable or lower memory (5.4 GB), and sets the best overall accuracy (44.3 IoU and 32.7 mIoU). GaussianFormer and GaussianFormer-2 are camera-only Gaussian-query baselines that process multi-view tokens with several global transformer blocks. Their IoU/mIoU are competitive within the camera-only group but remain below multi-modal entries in Tab. 4.

Method	Modality	Query Number	Lat. (ms)	Mem. (GB)	IoU↑	mIoU↑
BEVFormer[27]	C	200×200	310	4.5	30.5	16.8
TPVFormer[14]	C	200×200×16	320	5.1	30.9	17.1
SurroundOcc[47]	C	200×200×16	340	5.9	31.5	16.3
GaussianFormer[15]	C	25600	195	4.8	28.7	16.0
	C	144000	372	6.2	29.8	19.1
GaussianFormer-2[16]	C	12800	323	**3.0**	30.4	19.9
	C	25600	357	**3.0**	31.0	20.3
M-CONet[45]	L+C	100×100×8	670	7.8	39.2	24.7
Co-Occ[35]	L+C	100×100×8	595	12.1	41.1	27.1
DAOcc*[50]	L+C	456×456	130	4.2	40.5	30.3
DAOcc[50]	L+C	720×720	291	8.6	42.8	32.1
Ours	L+C	12800	**124**	3.3	42.4	31.5
	L+C	25600	230	5.4	**44.3**	**32.7**

Table 4. Comparison of inference efficiency on the nuScenes validation set. All results are measured with a batch size of 1 on a single NVIDIA RTX 4090 GPU.

We next compare with the multi-modal methods M-CONet and Co-Occ. Both methods fuse LiDAR and camera inputs with dense BEV queries ($100 \times 100 \times 8$) and therefore incur substantial latency and memory. M-CONet requires 670 ms and 7.8 GB to reach 39.2/24.7 IoU/mIoU, while Co-Occ takes 595 ms and 12.1 GB for 41.1/27.1. In contrast, our sparse Gaussian design is substantially more efficient in both time and space while achieving clearly better accuracy. With only 12.8k Gaussian queries, our model runs at 124 ms and 3.3 GB, which is about $5.4\times$ and $4.8\times$ faster than M-CONet (670 ms) and Co-Occ (595 ms), respectively, while reducing memory consumption by roughly 58% and 73% (from 7.8 GB and 12.1 GB to 3.3 GB), and simultaneously improving IoU/mIoU from 39.2/24.7 and 41.1/27.1 to 42.4/31.5.

The more recent method DAOcc adopts efficient per-query operations but relies on a very dense BEV grid with 720×720 queries (over $5\times$ more queries than M-CONet/Co-Occ and over $40\times$ more than our 12.8k-Gaussian setting), which leads to a total latency of 291 ms and 8.6 GB memory. Although the downsampled variant DAOcc* reduces the BEV resolution to 456×456, lowering latency and memory to 130 ms and 4.2 GB, it still remains slower and heavier than our sparse Gaussian model at 12.8k queries (124 ms, 3.3 GB), while achieving lower accuracy (40.5/30.3 vs. 42.4/31.5). These results highlight that representing the scene with a compact set of semantic Gaussians, combined

Table 5. Quantitative comparison on the KITTI-360 validation set. The best results are in **bold**, second best are underlined.

Method	Modality	IoU↑	mIoU↑	car	bicycle	motorcycle	truck	other-vehicle	person	road	parking	sidewalk	other-ground	building	fence	vegetation	terrain	pole	traffic-sign	other-structure	other-object
LMSCNet[38]	L	47.5	13.7	20.9	0.0	0.0	0.3	0.0	0.0	63.0	13.5	33.5	0.2	43.7	0.3	40.0	26.8	0.0	0.0	3.6	0.0
SSCNet[40]	L	53.6	17.0	32.0	0.0	0.2	10.3	0.6	0.1	65.7	17.3	41.2	3.2	44.4	6.8	43.7	28.9	0.8	0.8	8.6	0.7
L2COcc-L[44]	L	57.6	25.2	40.4	**4.1**	4.2	26.2	10.3	10.0	70.0	**22.6**	46.2	8.8	51.0	14.3	**47.6**	31.3	24.9	21.6	14.0	6.4
MonoScene[3]	C	37.9	12.3	19.3	0.4	0.6	8.0	2.0	0.9	48.4	11.4	28.1	3.2	32.9	3.5	26.2	16.8	6.9	5.7	4.2	3.1
VoxFormer[23]	C	38.8	11.9	17.8	1.2	0.9	4.6	2.1	1.6	47.0	9.7	27.2	2.8	31.2	5.0	29.0	14.7	6.5	6.9	3.8	2.4
TPVFormer[14]	C	40.2	13.7	21.6	1.1	1.4	8.1	2.6	2.4	53.0	12.0	31.1	3.8	34.8	4.8	30.1	17.5	7.5	5.9	5.5	2.7
OccFormer[54]	C	40.3	14.6	22.6	0.7	0.3	9.9	3.8	2.8	54.3	13.4	31.5	3.6	36.4	4.8	31.0	19.5	7.8	8.5	7.0	4.6
Gaussianformer[15]	C	35.4	12.9	18.9	1.0	**4.6**	18.1	7.6	3.4	45.5	10.9	25.0	5.3	28.4	5.7	29.5	8.6	3.0	2.3	9.5	5.1
Gaussianformer-2[16]	C	38.4	13.9	21.1	2.6	4.2	12.4	5.7	1.6	54.1	11.0	32.3	3.3	32.0	5.0	28.9	17.3	3.6	5.5	5.9	3.5
L2COcc-C[44]	C	48.1	20.1	29.6	3.7	4.4	14.9	8.4	7.2	63.3	17.9	40.5	5.2	42.8	8.5	39.4	24.5	16.2	18.4	10.2	6.8
Gau-Occ (Ours)	L+C	**58.9**	**25.8**	**42.1**	3.7	4.4	**27.2**	**10.8**	**10.5**	**70.9**	22.5	**47.1**	**9.4**	**51.7**	**14.6**	47.0	**32.2**	**25.2**	**22.7**	**14.1**	**7.8**

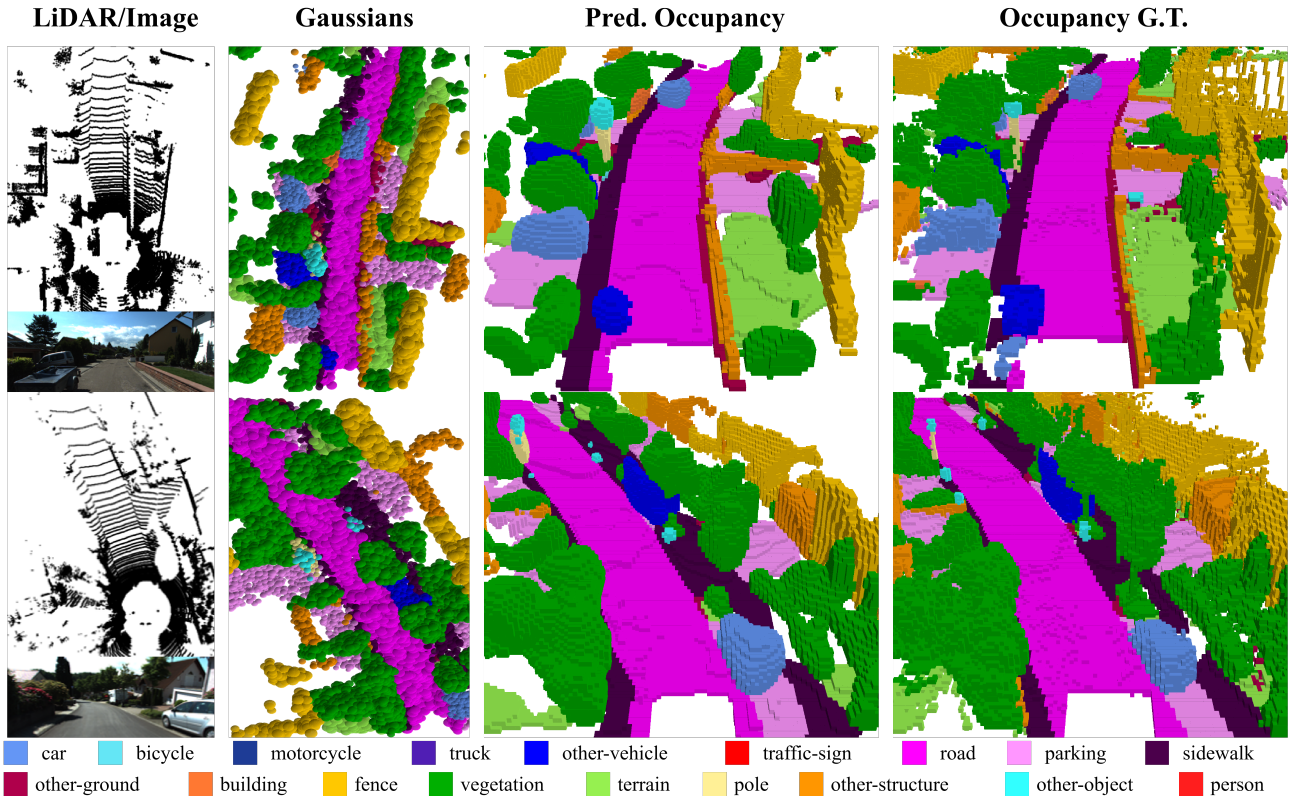

Figure 9. Qualitative results on KITTI-360.

with our highly compressed attention module, enables an architecture that is both simple and scalable: it closes the gap to strong camera-only baselines in terms of efficiency and at the same time clearly outperforms prior multi-modal occupancy methods in the accuracy-efficiency trade-off.

9. KITTI-360 Result

In the KITTI-360 benchmark, we provide a comprehensive comparison between **Gau-Occ** and both LiDAR-only and image-only baselines in Tab. 5. Multi-modal methods are scarce on this dataset, so L2COcc [44] serves as the primary strong LiDAR-only reference. As reported in Tab. 5 (this supplementary material), the proposed Gau-Occ surpasses L2COcc by **+1.3 IoU** and **+0.6 mIoU**, while remaining much superior to camera-only methods. Under this challenging single-camera configuration of KITTI-360, our model yields clear gains on moving vehicles (e.g., *car*, *truck*) and large-scale structural classes (e.g., *road*, *building*), indicating improved capability for reliable scene reconstruction from limited visual coverage.

Qualitative results in Fig. 9 further corroborate these findings: even with a single camera and sparse LiDAR, Gau-Occ reconstructs both global scene layouts and small

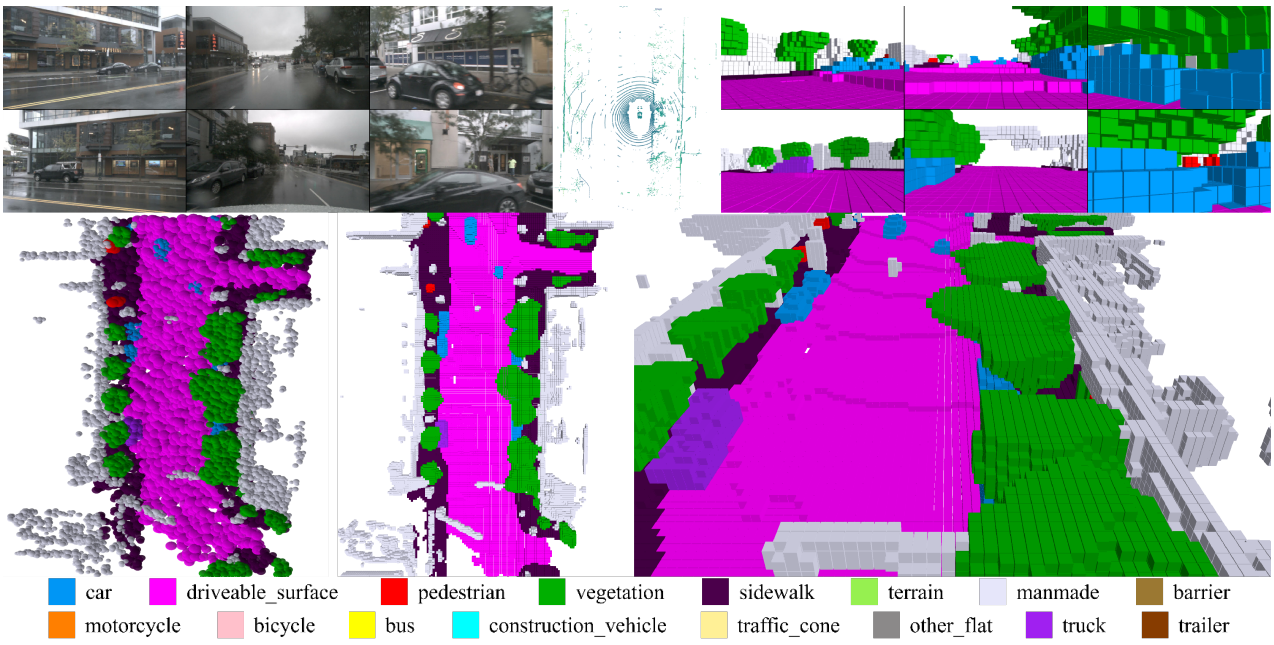

Figure 10. Additional qualitative results on the SurroundOcc-nuScenes validation set under **adverse weather** scenarios.**Top** shows multi-view images (left), raw LiDAR input (center), and predicted image-view occupancy (right); **Bottom** presents predicted 3D Gaussians, BEV occupancy, and front-view occupancy.

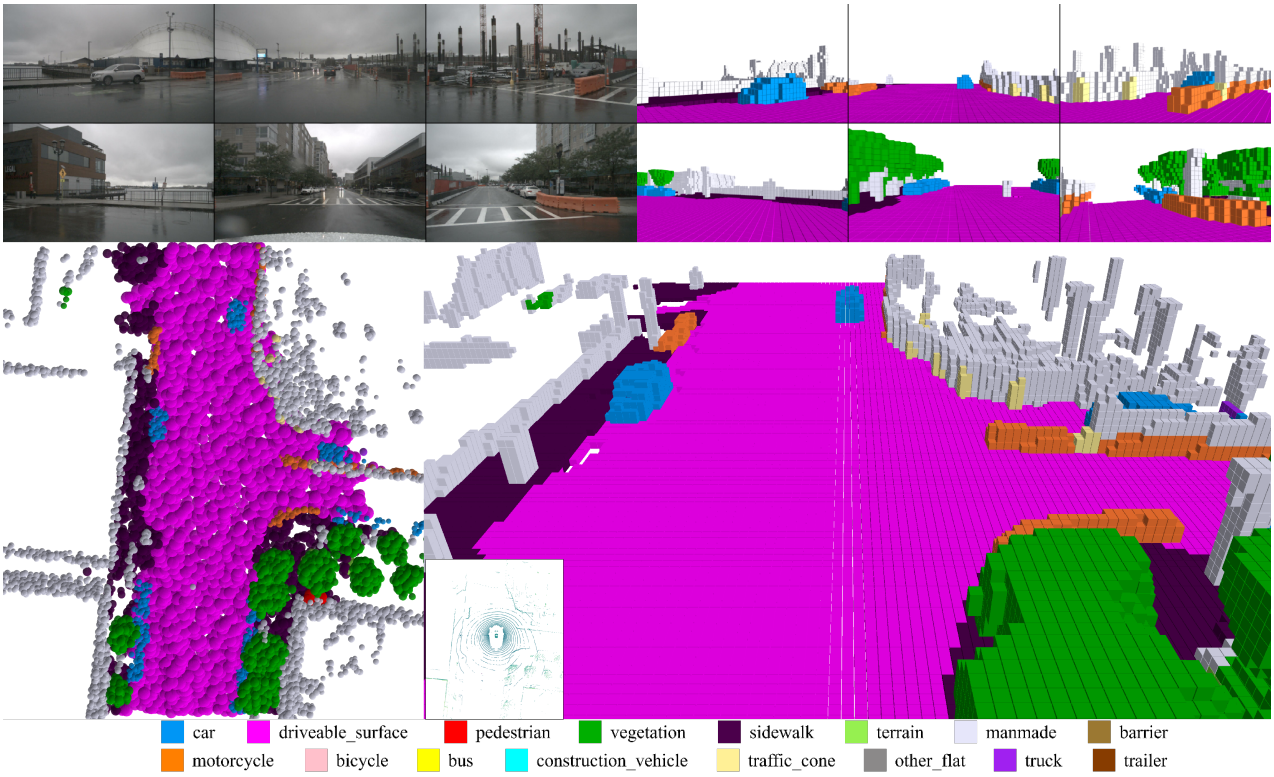

Figure 11. Additional qualitative results on the Occ3D-nuScenes validation set under **adverse weather** scenarios.**Top-left**: multi-view images; **top-right**: predicted image-view occupancy; **bottom-left**: predicted 3D Gaussians; **bottom-right**: front-view occupancy; inset: LiDAR input.

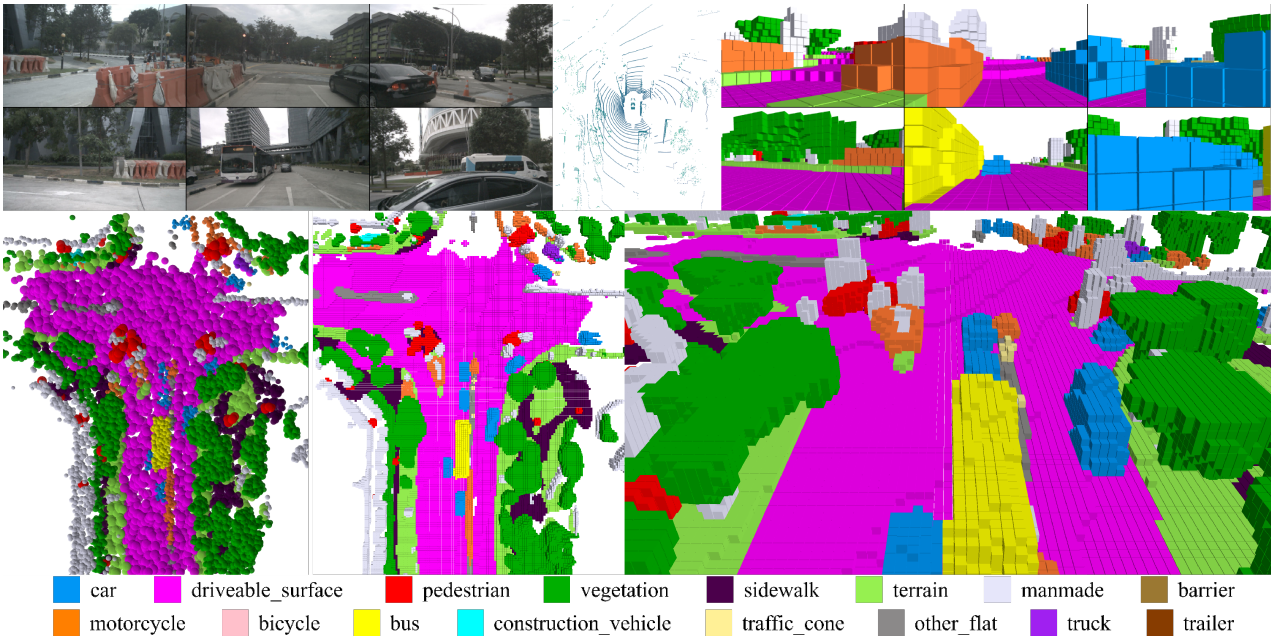

Figure 12. Additional qualitative results on the SurroundOcc-nuScenes validation set under **dense traffic** scenarios.

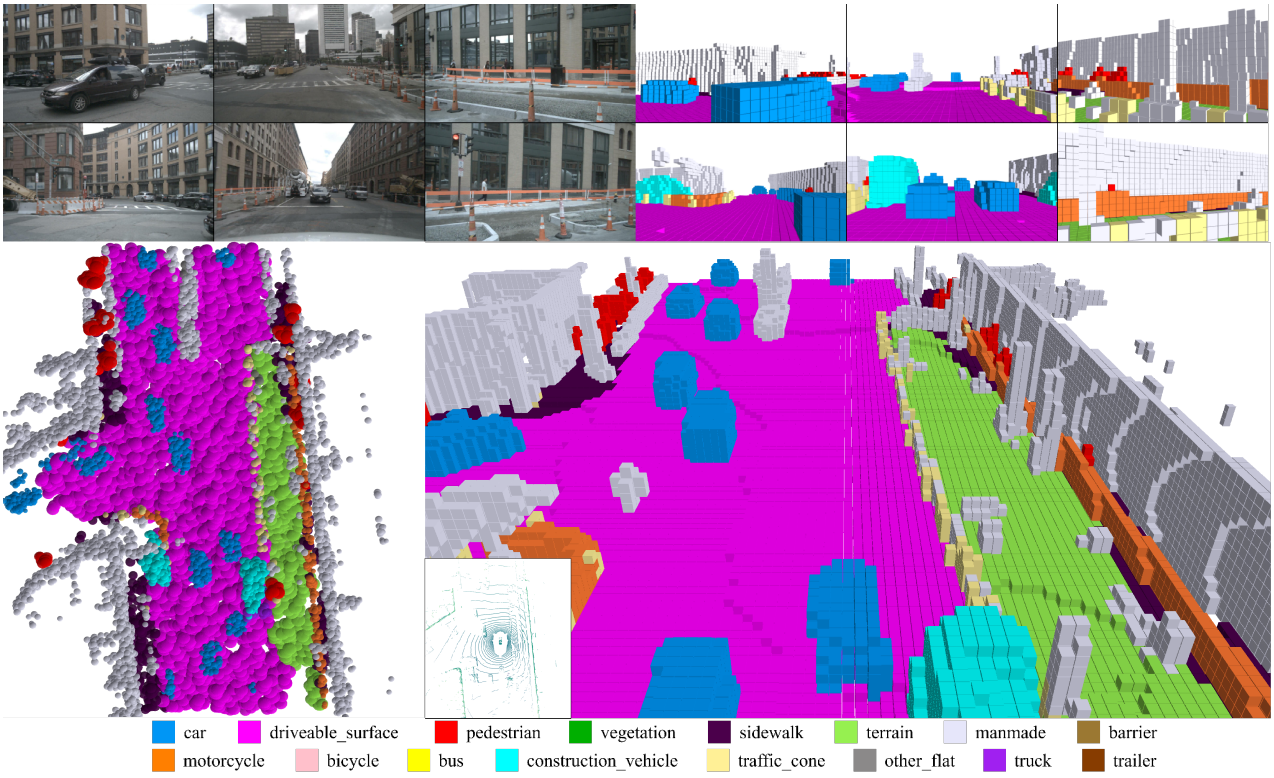

Figure 13. Additional qualitative results on the Occ3D-nuScenes validation set under **dense traffic** scenarios.

instances accurately, illustrating robustness to sparse viewpoints and effective exploitation of LiDAR geometry.

10. Additional Visualizations

We present additional 3D semantic occupancy prediction results on the Suroundocc-nuScenes and Occ3D-nuScenes validation set. Gau-Occ achieves accurate and complete

predictions across diverse challenging scenarios, including **adverse weather** as shown in Fig. 10 and Fig. 11 and **dense traffic** as shown in Fig. 12 and Fig. 13. These results further demonstrate Gau-Occ's strong generalization and robustness in handling sparse/noisy inputs and reasoning over complex or low-frequency scenes via geometry-aware, multi-modal Gaussian fusion.

Supplementary videos provide dynamic visualizations of our comparisons with strong baselines, DAOcc and GaussianFormer-2. Our method achieves noticeably higher occupancy accuracy in long-range and heavily occluded regions, and more reliably distinguishes visually similar on-road categories (e.g., *truck* vs. *car*) without introducing semantic ambiguity, owing to its more effective use of geometric priors.