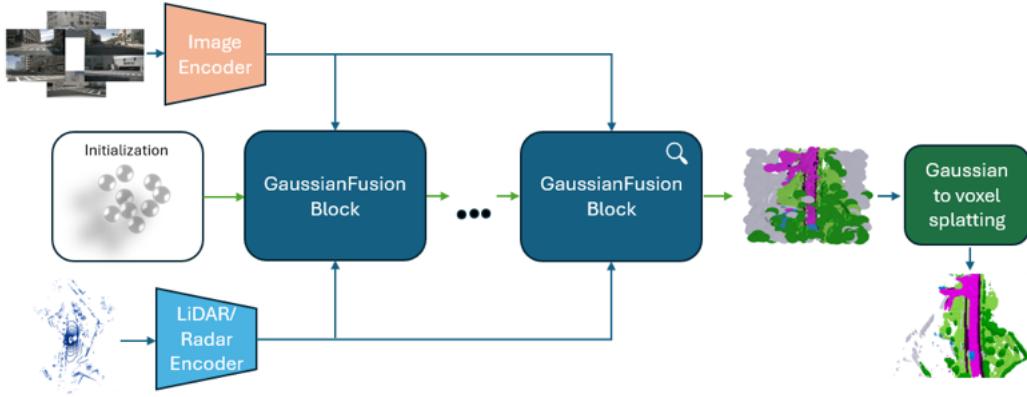


# GAUSSIANFUSIONOCC: A SEAMLESS SENSOR FUSION APPROACH FOR 3D OCCUPANCY PREDICTION USING 3D GAUSSIANS

006 **Anonymous authors**

007 Paper under double-blind review



026 **Figure 1: GaussianFusionOcc pipeline:** Our method utilizes modality-specific encoders to extract  
 027 feature maps from the input data. Extracted features are then fed to GaussianFusion blocks along  
 028 with Gaussians and queries from the previous block. The GaussianFusion block, visualized in Figure  
 029 2, extracts per-Gaussian features from each modality, fuses them into a unified feature vector, which  
 030 is then used to refine the input Gaussians. Resulting Gaussians from the last block are splatted to a  
 031 voxelized representation using the Gaussian-to-voxel splatting module.

## ABSTRACT

035 3D semantic occupancy prediction is one of the crucial tasks of autonomous  
 036 driving. It enables precise and safe interpretation and navigation in complex  
 037 environments. Reliable predictions rely on effective sensor fusion, as different  
 038 modalities can contain complementary information. Unlike conventional methods  
 039 that depend on dense grid representations, our approach, GaussianFusionOcc,  
 040 uses semantic 3D Gaussians alongside an innovative sensor fusion mechanism.  
 041 Seamless integration of data from camera, LiDAR, and radar sensors enables  
 042 more precise and scalable occupancy prediction, while 3D Gaussian representation  
 043 significantly improves memory efficiency and inference speed. GaussianFusionOcc  
 044 employs modality-agnostic deformable attention to extract essential features from  
 045 each sensor type, which are then used to refine Gaussian properties, resulting in a  
 046 more accurate representation of the environment. Extensive testing with various  
 047 sensor combinations demonstrates the versatility of our approach. By leveraging  
 048 the robustness of multi-modal fusion and the efficiency of Gaussian representation,  
 049 GaussianFusionOcc outperforms current state-of-the-art models.

## 1 INTRODUCTION

051 Accurate 3D semantic occupancy prediction is a foundational task for safe autonomous navigation.  
 052 Reliable perception of the surrounding environment enables precise situational awareness necessary

for informed decision-making, efficient route planning, and collision avoidance (Ming et al., 2024; Lu et al., 2024; Caesar et al., 2020; Wang et al., 2023). While recent advances in voxel-based methods (Li et al., 2023a; Jiang et al., 2024; Miao et al., 2023; Wei et al., 2023; Zhang et al., 2023) have pushed performance on benchmarks like nuScenes (Caesar et al., 2020; Wei et al., 2023; Wang et al., 2023), their reliance on dense volumetric grids creates prohibitive computational and memory costs (Huang et al., 2023; 2024b;a). Furthermore, these methods struggle with the inherent sparsity of relevant information in driving scenes, where most of the volume remains empty. Concurrently, sensor fusion remains critical for robustness and reliability in dynamic environments, as cameras, LiDAR, and radar provide complementary strengths: cameras capture rich semantics, LiDAR offers precise geometry, and radar ensures reliability in adverse conditions. However, existing fusion frameworks are constrained by their underlying representations, which lack adaptability to scene complexity and scalability across modalities (Liu et al. (2023); Ming et al. (2024); Wang et al. (2023)).

Emerging 3D Gaussian splatting techniques (Kerbl et al., 2023; Huang et al., 2024b;a; Yang et al., 2024) offer a promising alternative to voxels by modeling scenes with anisotropic, learnable Gaussians, enabling faster rendering and lower memory usage. The inherent sparsity and adaptability of Gaussian representations naturally align with the characteristics of driving scenes, where information density varies across the environment (Huang et al., 2024b). By concentrating representational capacity where it matters most, Gaussian-based methods achieve superior memory efficiency.

We present GaussianFusionOcc, a novel framework that extends the capabilities of Gaussian-based 3D scene representations to the multi-modal setting. While existing Gaussian-based approaches (Huang et al., 2024b;a) have been limited to single-modality inputs, our approach introduces a modality-agnostic Gaussian encoder, capable of extracting per-Gaussian features from any individual sensor modality, such as camera, LiDAR, or radar, using a deformable attention mechanism. This attention-based approach enables the encoder to focus computational resources on the most informative regions of each sensor’s feature map, ensuring robust performance even when certain sensors are impaired or provide conflicting information. Unlike traditional multi-modal fusion approaches that aggregate features at fixed spatial locations with uniform resolution, we propose a novel fusion paradigm that constructs unified Gaussian feature vectors from modality-specific features, fusing information directly on learnable geometric-semantic primitives. This shift enables the model to adaptively allocate Gaussians based on scene complexity, focusing modeling capacity on detailed or semantically rich regions while maintaining sparsity elsewhere.

The main contributions of this work are the following:

We introduce the first framework that leverages 3D Gaussian splatting for multi-modal 3D semantic occupancy prediction and fuses the sensor features in Gaussian space.

We propose a modality-agnostic Gaussian encoder based on a deformable attention mechanism that effectively extracts per-Gaussian features from diverse sensor modalities, and a fusion method that creates a unified representation based on the extracted features.

We demonstrate state-of-the-art performance on the nuScenes dataset, especially on the rainy and nighttime subsets, achieving superior performance accuracy while reducing memory requirements compared to leading voxel-based approaches.

We tested the model with various sensor combinations under different scenarios to demonstrate the performance gains.

## 2 RELATED WORK

Earlier 3D semantic occupancy prediction approaches (Li et al., 2023a; Cao & De Charette, 2022; Jiang et al., 2024; Miao et al., 2023; Wei et al., 2023; Zhang et al., 2023) mostly relied on dense voxel grids to represent the 3D scene. While effective in capturing fine-grained details, these dense grid-based methods often suffer from high computational and memory overhead due to the inherent sparsity of real-world environments and the need for high-resolution grids. To mitigate these limitations, recent approaches (Li et al. (2024); Huang et al. (2023); Li et al. (2023b)) explored alternative scene representations like Bird’s-Eye-View (BEV) and Tri-Perspective View (TPV), achieving strong performance with improved efficiency. However, these planar representations often

108 involve a compression of 3D information, potentially leading to a loss of fine-grained geometric  
 109 details necessary for accurate 3D occupancy prediction.  
 110

111 In pursuit of more efficient and scalable 3D scene representations, object-centric approaches (Huang  
 112 et al., 2024b;a; Gan et al., 2024; Yang et al., 2024; Lu et al., 2024; Wang et al., 2024) have emerged as  
 113 efficient alternatives to dense voxel grids, representing scenes using primitives centered around objects  
 114 or regions of interest to avoid computational redundancy. Among these, 3D Semantic Gaussians have  
 115 recently gained traction as a flexible and efficient representation for 3D scenes (Huang et al., 2024b;a;  
 116 Gan et al., 2024; Yang et al., 2024), with each Gaussian primitive capturing local geometric and  
 117 semantic information. GaussianFormer (Huang et al., 2024b) introduced an object-centric approach  
 118 using 3D semantic Gaussians for vision-based 3D semantic occupancy prediction, demonstrating  
 119 comparable performance to state-of-the-art methods with significantly reduced memory consumption.  
 120 By representing the scene with a set of learnable 3D Gaussians and employing efficient Gaussian-to-  
 121 voxel splatting for occupancy prediction, GaussianFormer showcases the potential of object-centric  
 122 representations for achieving both accuracy and efficiency. Probabilistic extensions (Huang et al.,  
 123 2024a) further aim to improve the utilization and efficiency of 3D Gaussian representations.

124 Recognizing the limitations of relying solely on a single sensor, multi-sensor fusion has become a  
 125 critical direction for robust 3D occupancy prediction. Integrating LiDAR data with camera images  
 126 has been shown to improve depth estimation and overall perception accuracy. BEVFusion (Liu  
 127 et al., 2023) proposed fusing LiDAR and camera features in the BEV space for multi-task perception.  
 128 SparseFusion (Xie et al., 2023) further refined the feature fusion module for improved efficiency.  
 129 While these methods primarily focus on 3D object detection (Liu et al., 2023; Xie et al., 2023; Chen  
 130 et al., 2023), there's a growing interest in multi-sensor fusion for 3D semantic occupancy prediction.  
 131 Camera-radar fusion has also been explored for tasks like object detection and tracking (Kim et al.,  
 132 2023; Nabati & Qi, 2021; Chen et al., 2023), but dedicated approaches for 3D semantic occupancy  
 133 prediction are scarce due to the sparsity of radar data. To address the need for robust occupancy  
 134 prediction, OccFusion (Ming et al., 2024) was introduced as a novel framework to integrate features  
 135 from surround-view cameras, radars, and 360-degree LiDAR through dynamic fusion modules,  
 136 demonstrating superior performance in challenging conditions. These efforts highlight the benefits of  
 137 combining complementary sensor data to achieve more reliable and accurate 3D scene understanding  
 138 for autonomous driving. Building upon these advancements, our GaussianFusionOcc leverages  
 139 the efficiency and flexibility of 3D semantic Gaussians while introducing a seamless sensor fusion  
 140 mechanism to harness the complementary strengths of camera, LiDAR, and radar data for robust and  
 141 accurate 3D occupancy prediction.

### 3 METHODS

142 In this section, we present our Gaussian-space sensor fusion approach for 3D semantic occupancy  
 143 prediction, which fuses information directly on learnable 3D Gaussian primitives. This enables  
 144 adaptive spatial allocation that scales sublinearly with respect to the scene resolution. Our approach  
 145 extracts features from sensors using sensor-specific encoders. Sensor-specific features are then fed  
 146 into the Gaussian encoder blocks, where modality-agnostic deformable attention (Zhu et al., 2020)  
 147 is utilized to extract Gaussian-centric features. These features are later fused at the Gaussian level  
 148 and used to iteratively refine the properties of probabilistic semantic 3D Gaussians, which act as  
 149 persistent carriers of multi-modal evidence. Finally, these refined Gaussians are splatted to generate a  
 150 3D voxel grid representation.  
 151

152 **Sensor feature extraction:** We employ tailored feature extraction pipelines for each modality to  
 153 effectively encode information from our diverse set of sensors.  
 154

155 For the surrounding camera images  $I = \{I_i \in \mathbb{R}^{3 \times H \times W}\}_{i=1}^N$  where  $H, W, N$  are height, width,  
 156 and number of cameras, we utilize a combination of ResNet101-DCN (He et al., 2016; Dai et al.,  
 157 2017) as our 2D backbone and a Feature Pyramid Network (FPN) (Lin et al., 2017) as the neck. This  
 158 architecture allows us to generate multi-scale image features  $F_i^{cam} = \{F_{i,j}^{cam} \in \mathbb{R}^{C_i \times H_i \times W_i}\}_{j=1}^M$   
 159 for  $i$ -th image where  $M$  is the number of scales.

160 To process the LiDAR point cloud  $P^{lidar} \in \mathbb{R}^{C \times H \times W}$ , we first perform voxelization of the input data.  
 161 Following this, we employ a VoxelNet (Zhou & Tuzel, 2018) encoder along with an FPN (Lin et al.,  
 162 2017) to produce multi-scale Bird's-Eye-View (BEV) features  $F_i^{lidar} = \{F_i^{lidar} \in \mathbb{R}^{C_i \times H_i \times W_i}\}_{i=1}^M$ .

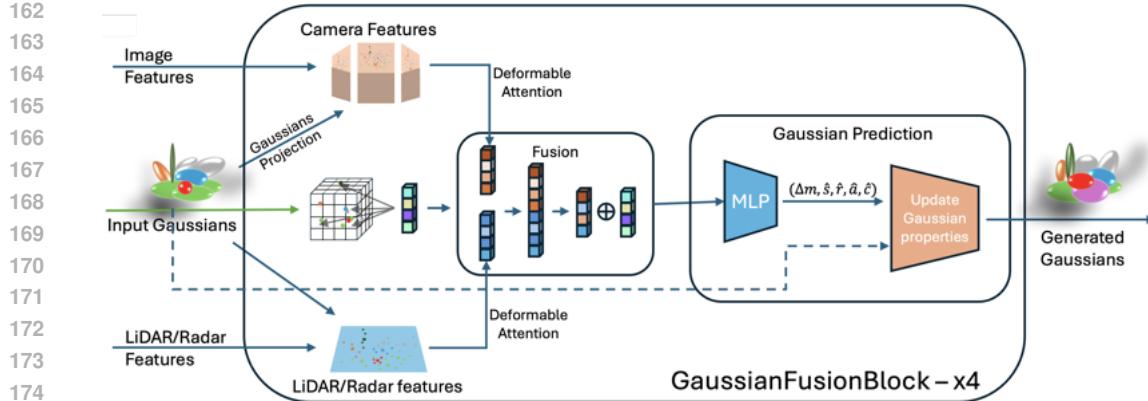


Figure 2: **GaussianFusionBlock architecture.** Modality-agnostic Gaussian encoders employ a deformable attention mechanism to extract relevant features for each initial Gaussian. These per-Gaussian features are then fed to the fusion module along with the resulting features of the sparse convolution layer applied to voxelized Gaussians. Fusion module concatenates Gaussian encoder features, applies an MLP fuser, and adds sparse convolution features to create unified Gaussian features. These features are then used in the Gaussian prediction module to refine initial Gaussians by predicting mean offset, scale, rotation, opacity, and semantic class.

Our radar encoder first pillarizes the input point cloud  $P^{radar} \in \mathbb{R}^{C \times H \times W}$ . The pillarization is a form of voxelization where the 3D space is divided into vertical pillars. It then encodes the radar features using a PointPillars [Lang et al. 2019] voxel encoder and a middle encoder. Radar data, while providing valuable information about object velocity, typically yields sparse features  $F^{radar} \in \mathbb{R}^{C \times H \times W}$  compared to camera and LiDAR.

Inspired by the advancements in GaussianFormer [Huang et al. 2024b] and GaussianFormer-2 [Huang et al. 2024a], our method reconstructs semantic probabilistic 3D Gaussians to represent the scene occupancy. This object-centric approach aims to overcome the limitations of dense grid-based representations by sparsely modeling the scene with learnable Gaussians. To initialize the Gaussian properties, we randomly sample a desired number of Gaussians  $\mathcal{G} = \{m_i, s_i, r_i, a_i, c_i\}_{i=1}^P$ , use learnable parameters for initialization, or choose the desired number of points from the input LiDAR point cloud for Gaussian means. Our model comprises three key modules: Gaussian encoder, fusion module, and Gaussian prediction module.

**Gaussian encoder:** To refine initial Gaussians, our method uses independent instances of the modality-agnostic Gaussian encoder module, which extracts a feature vector for each Gaussian. The encoder samples multiple 3D reference points around the center of a Gaussian  $m$  while taking into consideration the Gaussian's shape and size determined by its scale and rotation matrices  $S, R$ :

$$\Delta m = RS\Phi^{offset}(q), \quad R = \{m + \Delta m_i\}_{i=1}^{N_R} \quad (1)$$

where  $\Phi^{offset}, q$  denote the MLP for offset prediction, and the input query for the specific Gaussian, respectively. Initial queries for the first block are randomly sampled. These 3D reference points are then projected onto the feature maps of the input sensors using the sensors' intrinsic and extrinsic parameters. This projection establishes a correspondence between the Gaussian and the 2D features captured by the sensors. To extract relevant information from the sensor feature maps at the projected reference points, the Gaussian Encoder utilizes a deformable attention function (Zhu et al. [2020]):

$$F^{GE} = \sum_{i=1}^N \sum_{j=1}^{N_R} DA(Q, P(R), F^{sensor}) \quad (2)$$

where  $DA()$ ,  $P()$ ,  $F^{sensor}$ ,  $N$ ,  $N_R$  denote deformable attention function, projection from world to sensor features coordinates, sensor feature maps, number of sensor inputs (number of cameras or radars), and number of reference points, respectively. The attention mechanism computes a weighted sum of the features sampled around these refined reference points, where the weights indicate the

relevance of each sampled feature. This aggregated feature vector represents the visual (or other sensory) information associated with the 3D Gaussian.

**Fusion:** The fusion module is designed to effectively integrate multi-modal sensor information to generate a unified feature representation for each 3D Gaussian. Given a set of per-Gaussian feature vectors extracted from individual sensors such as cameras, LiDAR, and radar, the fusion module aims to create a comprehensive descriptor that leverages the complementary strengths of each modality. The feature vectors derived from each available sensor modality are concatenated along the feature channel dimension. This operation results in a single, extended feature vector that aggregates information from all input sensors. If we denote the feature vector for a Gaussian from the  $i$ -th sensor as  $F_i$ , the concatenation operation can be represented as:  $F^{concatenated} = [F_1, F_2, \dots, F_n]$  where  $n$  is the number of sensor modalities being utilized. Following the concatenation, the combined feature vector is passed through a Multi-Layer Perceptron  $\Phi^{fusion}$ . The MLP serves to learn complex interdependencies between features from different sensors, weigh the contribution of different sensor features based on their relevance for representing the 3D Gaussian, and reduce the dimensionality of the concatenated feature vector to a more manageable and informative unified representation.

The feature vector for each 3D Gaussian on the output of the MLP encapsulates the integrated information from all sensor modalities, providing a robust descriptor of the local 3D space. In parallel, input Gaussians are represented as a point cloud of their means. The voxelized representation is then fed to 3D sparse convolutions to leverage the spatial relationships between these Gaussians, inspired by the self-encoding module from GaussianFormer (Huang et al. 2024b). Sparse convolutions  $SC$  are particularly suitable for processing sparse 3D data as they only operate on occupied voxels, allowing each Gaussian to gather contextual information from neighboring Gaussians. Finally, to combine the benefits of both pathways, the sparse convolution outputs are added to the MLP-generated feature vectors for each corresponding Gaussian:

$$Q = F^{unified} = \Phi^{fusion}(F^{concatenated}) \oplus SC(m) \quad (3)$$

This element-wise addition fuses the modality-specific information (refined by the MLP) with the spatial context information (learned by the sparse convolutions). These unified Gaussian feature vectors are subsequently used for refining the properties of the Gaussians, ultimately contributing to the accurate prediction of 3D semantic occupancy.

**Gaussian prediction:** Leveraging the information aggregated from multi-modal sensor inputs, the Gaussian Prediction module updates the parameters of each Gaussian to better represent the surrounding 3D scene. Each Gaussian acts as a persistent carrier of multi-modal evidence across refinement blocks. The functionality of the Gaussian prediction module draws inspiration from the refinement module employed in GaussianFormer (Huang et al. 2024b). Specifically, for each 3D Gaussian, the unified feature vector from the Fusion module serves as input to a multi-layer perceptron  $\Phi^{refine}$ . This MLP decodes intermediate Gaussian properties, including offset to the mean ( $\hat{m}$ ), scale ( $\hat{s}$ ), rotation ( $\hat{r}$ ), opacity ( $\hat{a}$ ), and semantic logits ( $\hat{c}$ ):

$$(\hat{m}, \hat{s}, \hat{r}, \hat{a}, \hat{c}) = \Phi^{refine}(Q) \quad (4)$$

Following the refinement step, the updated Gaussian properties and unified Gaussian feature vectors are fed to the next Gaussian fusion block for subsequent refinement. This iterative refinement process allows for a progressively more accurate and semantically meaningful representation of the 3D scene. After the final refinement, the collection of refined 3D Gaussians is used to generate the final 3D semantic occupancy prediction. To achieve this, we employ the Gaussian-to-Voxel splatting method  $GS$  proposed in GaussianFormer-2 (Huang et al. 2024a):

$$o(x) = GS(m, s, r, a, c) \quad (5)$$

where  $x$  denotes the voxel position. This approach utilizes probabilistic Gaussian superposition to transform the sparse Gaussian representation into a dense voxel grid occupancy prediction.

To train GaussianFusionOcc, we use lovasz-softmax loss (Berman et al. 2018)  $L_{lovasz}$  and binary cross entropy loss  $L_{BCE}$ . During the training, the model splats Gaussians to a voxel grid representation in each iteration of the refinement, so the losses could be calculated for each iteration. If  $N$  denotes the number of refinement iterations, we can then express the total loss as:

$$L^{total} = \sum_{i=1}^N (L_i^{lovasz} + L_i^{BCE}) \quad (6)$$

270 **4 EXPERIMENTS**  
 271

272 In this section, we present a comprehensive evaluation of our proposed GaussianFusionOcc framework  
 273 for 3D semantic occupancy prediction. We assess its performance on the nuScenes validation set,  
 274 including challenging rainy and nighttime scenarios, and analyze the contribution of key components  
 275 through ablation studies. Efficiency comparisons are also provided.  
 276

277 **4.1 DATASET DETAILS**  
 278

279 We use the nuScenes dataset (Caesar et al., 2020) for training and evaluation, which contains 1000  
 280 scenes with keyframes annotated at 2Hz, each equipped with 6 surround-view cameras, 5 radars, and 1  
 281 LiDAR. Since nuScenes lacks native dense 3D semantic occupancy ground truth, we use annotations  
 282 generated by SurroundOcc (Wei et al., 2023). The occupancy prediction range spans [-50m, 50m] for  
 283 X and Y axes and [-5m, 3m] for the Z axis, with 0.5-meter voxel resolution. We trained the model on  
 284 the official training split (700 scenes) and evaluated on the validation split (150 scenes).  
 285

286 **4.2 IMPLEMENTATION DETAILS**  
 287

288 GaussianFusionOcc uses a ResNet101-DCN (He et al., 2016; Dai et al., 2017) as the image feature  
 289 backbone, initialized with weights from the FCOS3D (Wang et al., 2021) checkpoint. Resulting  
 290 image feature maps are then processed using FPN (Lin et al., 2017) to create multi-scale image  
 291 features with 1/4, 1/8, 1/16, and 1/32 of the original resolution. For the LiDAR encoder, it uses  
 292 VoxelNet (Zhou & Tuzel, 2018) with weights from the FUTR3D (Chen et al., 2023) checkpoint,  
 293 and FPN (Lin et al., 2017) producing multi-scale image features with the same downsampling as  
 294 for image features. Radar encoder uses voxel encoder and middle encoder from PointPillars (Lang  
 295 et al., 2019) network initialized with weights from FUTR3D (Chen et al., 2023) checkpoint. For  
 296 the 3D Gaussian representation, we use 6400 Gaussians in our experiments. The model consists of  
 297 4 Gaussian fusion blocks, each consisting of Gaussian encoders, a fusion module, and a Gaussian  
 298 prediction module. The Gaussian encoder and fusion module produce features with 128 channels. We  
 299 train our model for 20 epochs with a batch size of 8 on an NVIDIA RTX A6000 GPU. We employ  
 300 the AdamW optimizer (Loshchilov & Hutter, 2017) with a weight decay of 0.01. The learning rate is  
 301 warmed up to 2e-4 in the first 500 iterations, and subsequently decreased following a cosine schedule.  
 302 For data augmentation, we apply photometric distortions to the input images.  
 303

304 **4.3 MAIN RESULTS**  
 305

306 Table 1: 3D semantic occupancy prediction results on nuScenes validation set. \* marks the model  
 307 supervised by dense occupancy annotations, while the original was trained with LiDAR segmentation  
 308 labels. \*\* marks the model using LiDAR initialization, \*\*\* the random initialization, while other  
 309 models use learnable initialization. Modality notation: Camera (C), LiDAR (L), Radar (R)

Method	Modality	IoU	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	driveable surf.	other flat	sidewalk	terrain	man made	vegetation
MonScene (Cao & De Charette, 2022)	C	23.96	7.31	4.03	0.35	8.00	8.04	2.90	0.28	1.16	0.67	4.01	4.35	27.72	5.20	15.13	11.29	9.03	14.86
Atlas (Murez et al., 2020)	C	28.66	15.00	10.64	5.68	19.66	24.94	8.90	8.84	6.47	3.28	10.42	16.21	34.86	15.46	21.89	20.95	11.21	20.54
BEVNet (Li et al., 2024)	C	30.50	16.75	14.22	6.58	23.46	28.28	8.66	10.77	6.64	4.05	11.20	17.78	37.28	18.00	32.88	22.17	13.80	22.21
TPVFormer (Huang et al., 2023)	C	11.51	11.66	16.14	7.17	22.63	17.13	8.83	11.39	10.46	8.23	9.43	17.02	8.07	13.64	13.85	10.34	4.90	7.37
TPVFormer (Huang et al., 2024)	C	30.86	17.10	15.96	5.31	23.86	27.32	9.79	8.74	7.09	5.20	10.97	19.22	38.87	21.25	24.26	23.15	11.73	20.81
OcFormer (Zhang et al., 2023)	C	31.39	19.03	18.65	10.41	23.92	30.29	10.31	14.19	13.59	10.13	12.49	20.77	38.78	19.79	24.19	22.21	13.48	21.35
SurroundOcc (Wei et al., 2023)	C	31.49	20.30	20.59	11.68	28.06	30.86	10.70	15.14	14.09	12.06	14.38	22.26	37.29	23.70	24.49	22.77	14.89	21.86
GaussianFormer (Huang et al., 2024b)	C	29.83	19.10	19.52	11.26	26.11	29.78	10.47	13.83	12.58	8.67	12.74	21.57	39.63	23.28	24.46	22.99	9.59	19.12
GaussianFormer-2 (Huang et al., 2024a)	C	31.74	20.82	21.39	13.44	28.49	30.82	10.92	15.84	13.55	10.53	14.04	22.92	40.61	24.36	26.08	24.27	13.83	21.98
L-CONEt (Wang et al., 2023)	L	39.40	17.70	19.20	4.00	15.10	26.90	6.20	3.80	6.80	6.00	14.10	13.10	39.70	19.10	24.00	23.90	25.10	35.70
M-CONEt (Wang et al., 2023)	C+L	39.20	24.70	24.80	13.00	31.60	34.80	14.60	18.00	14.70	20.40	26.60	39.20	22.80	26.10	26.00	26.00	37.10	
OcCFusion (Ming et al., 2024)	C+R	33.97	20.73	20.46	13.98	27.99	31.52	13.68	18.45	15.79	13.05	13.98	23.84	37.85	19.60	22.41	21.20	16.16	21.81
OcCFusion (Ming et al., 2024)	C+L	44.35	26.87	26.67	18.38	32.97	35.81	19.39	22.17	24.49	21.46	29.67	39.01	21.94	24.90	26.76	28.53	40.03	
OcCFusion (Ming et al., 2024)	C+L+R	44.66	27.30	27.09	<b>19.56</b>	33.68	36.23	21.66	24.84	25.29	17.33	21.81	30.01	39.53	19.94	24.94	26.45	28.93	40.41
GaussianFusionOcc** (Ours)	C	37.05	22.43	22.46	14.19	28.66	29.93	15.10	17.08	18.22	9.71	18.21	25.12	37.19	20.59	23.39	23.65	22.91	32.55
GaussianFusionOcc** (Ours)	C+R	37.37	22.80	22.66	13.95	29.70	31.30	15.84	18.16	19.02	9.61	17.63	25.51	37.72	20.02	23.30	23.57	23.66	33.21
GaussianFusionOcc** (Ours)	L	<b>45.32</b>	29.75	30.02	16.46	35.02	38.93	22.25	24.65	29.64	18.41	<b>24.64</b>	30.93	<b>43.31</b>	<b>26.30</b>	28.95	29.29	34.33	42.92
GaussianFusionOcc (Ours)	C+L	45.16	30.21	30.22	18.70	35.91	39.57	<b>22.67</b>	<b>27.36</b>	30.10	18.59	24.45	31.25	43.06	25.76	29.12	29.33	34.65	42.70
GaussianFusionOcc (Ours)	C+L+R	45.20	<b>30.37</b>	<b>30.43</b>	18.54	<b>36.23</b>	<b>39.66</b>	22.57	27.35	<b>30.30</b>	<b>19.14</b>	24.56	<b>31.95</b>	42.60	25.82	<b>29.48</b>	<b>29.70</b>	<b>34.78</b>	<b>42.95</b>

321 The superior performance of our proposed GaussianFusionOcc framework is clearly demonstrated  
 322 in Table I, where it outperforms a range of state-of-the-art methods across various categories of 3D  
 323 semantic occupancy prediction on the nuScenes validation set (Caesar et al., 2020).

324 We compare our GaussianFusionOcc with several categories of existing methods:  
 325

326 **Comparison with planar-based models:** Our results indicate a significant improvement over planar-  
 327 based methods such as BEVFormer (Li et al., 2024) and TPVFormer (Huang et al., 2023). While  
 328 these models lift 2D image features to a Bird’s-Eye View (BEV) or Tri-Perspective View (TPV), the  
 329 object-centric nature of Gaussians in GaussianFusionOcc allows for a more efficient and flexible  
 330 representation. GaussianFusionOcc outperforms planar-based methods across all semantic categories,  
 331 with most pronounced gains for small or vertically complex classes like pedestrians or motorcycles.

332 **Comparison with grid-based models:** GaussianFusionOcc demonstrates improved performance  
 333 over grid-based representations like OccFormer (Zhang et al., 2023). This is attributed to the sparse  
 334 modeling of the scene with learnable Gaussians, which better allocates representational capacity  
 335 to complex object shapes and details, particularly in capturing fine-grained structures and handling  
 336 varying object scales.

337 **Comparison with existing gaussian-based models:** Even when compared to other Gaussian-  
 338 based methods, such as GaussianFormer-2 (Huang et al., 2024a), GaussianFusionOcc demonstrates  
 339 enhanced performance. This improvement can be attributed to the unique multi-sensor fusion  
 340 mechanism employed in GaussianFusionOcc, which addresses the limitation of previous Gaussian-  
 341 based approaches that rely solely on camera input. The superior results over GaussianFormer-2  
 342 highlight the effectiveness of our fusion framework in leveraging the benefits of the Gaussian  
 343 representation for 3D semantic occupancy prediction with multi-sensor input.

344 **Comparison with sensor-fusion models:** Table I demonstrates that GaussianFusionOcc also sur-  
 345 passes sensor-fusion models such as M-CONet (Wang et al., 2023) and OccFusion (Ming et al., 2024).  
 346 Notably, GaussianFusionOcc achieves this superior performance with a significantly lower number  
 347 of parameters (79.96M) compared to M-CONet (137M) and OccFusion (114.97M). This suggests  
 348 that our novel approach, utilizing a more effective representation and Gaussian-space sensor fusion,  
 349 can surpass the performance of multi-sensor fusion techniques based on grid representations.

350 Overall, the results presented in Table I establish GaussianFusionOcc as a new state-of-the-art for  
 351 3D semantic occupancy prediction. Our model’s superior performance across different architectural  
 352 paradigms underscores the efficacy of its core design principles.

#### 353 4.4 PERFORMANCE UNDER CHALLENGING SCENARIOS

356 Table 2: 3D semantic occupancy prediction results on rainy scenario subset of nuScenes validation  
 357 set. All methods are trained with dense occupancy labels from (Wei et al., 2023). Modality notation:  
 358 Camera (C), LiDAR (L), Radar (R).

Method	Modality	IoU	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	driveable surf.	other flat	sidewalk	terrain	man made	vegetation
OccFusion (Ming et al., 2024)	C	31.10	18.99	18.55	14.29	22.28	30.02	10.19	15.20	10.03	9.71	13.28	20.98	37.18	23.47	27.74	17.46	10.36	23.13
SurroundOcc (Wei et al., 2023)	C	30.57	21.40	21.40	12.75	25.49	31.31	11.39	12.65	8.94	9.48	14.51	21.52	35.34	<b>25.32</b>	29.89	18.37	14.44	24.78
OccFusion (Ming et al., 2024)	C+R	33.75	20.78	20.14	16.33	26.37	32.39	11.56	17.08	11.14	10.54	13.61	22.42	37.50	22.79	29.50	17.58	17.06	26.49
OccFusion (Ming et al., 2024)	C+L	43.26	26.55	24.95	19.11	34.23	36.07	17.01	21.07	18.87	17.46	21.91	28.73	37.82	24.39	30.80	20.37	28.95	43.12
OccFusion (Ming et al., 2024)	C+L+R	43.59	26.72	25.30	18.71	33.58	36.28	17.76	22.44	20.80	15.89	22.01	28.75	39.28	22.72	30.78	20.15	28.99	43.37
GaussianFusionOcc	C	36.83	21.86	21.50	12.56	28.89	29.93	13.16	12.97	12.31	7.60	19.32	23.33	38.12	22.13	29.17	20.05	22.60	36.00
GaussianFusionOcc	L	43.80	28.40	27.16	16.79	36.99	38.09	<b>21.08</b>	16.09	25.48	17.60	<b>26.67</b>	30.37	40.36	24.30	33.59	22.47	33.28	44.16
GaussianFusionOcc	C+R	37.03	21.87	21.97	13.92	29.91	30.64	9.68	12.90	12.70	6.72	19.48	23.97	37.43	21.78	29.02	19.58	23.77	36.51
GaussianFusionOcc	C+L	44.28	29.19	28.10	19.84	36.28	38.90	18.11	21.13	<b>26.14</b>	17.95	25.79	29.92	<b>41.72</b>	<b>27.35</b>	34.99	22.85	33.84	<b>44.10</b>
GaussianFusionOcc	C+L+R	<b>44.36</b>	<b>29.86</b>	<b>28.40</b>	<b>19.88</b>	<b>38.87</b>	<b>39.33</b>	20.61	<b>26.05</b>	25.66	<b>17.97</b>	26.07	<b>31.02</b>	41.70	24.94	<b>35.27</b>	<b>24.08</b>	<b>33.90</b>	44.00

367 We further evaluate the performance of GaussianFusionOcc under challenging weather and lighting  
 368 conditions on subsets of the nuScenes validation set, specifically focusing on rainy and nighttime  
 369 scenarios. The results are presented in Table 2 and Table 3.

370 **Rainy scenario:** As shown in Table 2, GaussianFusionOcc (C+L+R) achieves an mIoU of 29.86%  
 371 and IoU of 44.36%. Compared to single-modality variants, the fusion of camera, LiDAR, and radar  
 372 consistently improves performance in rainy conditions. These results highlight the robustness of our  
 373 multi-sensor fusion strategy to adverse weather, where cameras may struggle.

375 **Night scenario:** Table 3 shows performance in nighttime scenarios. LiDAR-only achieves the  
 376 best results (43.00% IoU, 18.76% mIoU), demonstrating active sensing superiority in low-light  
 377 conditions. Multi-modal configurations C+L and C+L+R show modest decreases of 0.10 and 0.31  
 378 mIoU respectively, likely due to noise-dominated gradients from sparse radar signals and cameras’

378  
379  
380  
381

Table 3: 3D semantic occupancy prediction results on night scenario subset of nuScenes validation set. All methods are trained with dense occupancy labels from (Wei et al., 2023). Modality notation: Camera (C), LiDAR (L), Radar (R).

Method	Modality	IoU	mIoU	barrier	bicycle	bus	car	constr. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	driveable surf.	other flat	sidewalk	terrain	man made	vegetation
OccFusion (Ming et al., 2024)	C	24.49	9.99	10.40	12.03	0.00	29.94	0.00	9.92	4.88	0.91	0.00	17.79	29.10	2.37	10.80	9.40	8.68	13.57
SurroundOcc (Wei et al., 2023)	C	24.38	10.80	10.55	14.60	0.00	31.05	0.00	8.26	5.37	0.58	0.00	18.75	30.72	2.74	12.39	11.53	10.52	15.77
OccFusion (Ming et al., 2024)	C+R	27.09	11.13	10.78	12.77	0.00	33.50	0.00	12.72	4.91	0.61	0.00	19.97	29.51	0.94	12.15	10.72	11.81	17.72
OccFusion (Ming et al., 2024)	C+L	41.38	15.26	12.74	13.52	0.00	35.85	0.00	15.33	13.19	0.83	0.00	23.78	32.49	0.92	14.24	20.54	23.57	37.10
OccFusion (Ming et al., 2024)	C+L+R	41.47	15.82	13.27	13.53	0.00	36.41	0.00	19.71	12.16	2.04	0.00	25.90	32.44	0.80	14.30	21.06	24.49	37.00
GaussianFusionOcc	C	32.05	11.28	6.13	4.41	0.00	30.40	0.00	12.51	3.06	0.43	0.00	21.55	28.91	3.54	13.10	13.71	16.12	26.58
GaussianFusionOcc	L	<b>43.00</b>	<b>18.76</b>	<b>19.62</b>	8.32	0.00	39.74	0.00	26.85	13.28	0.21	0.00	<b>39.11</b>	39.01	2.02	<b>20.04</b>	<b>22.81</b>	29.03	<b>40.12</b>
GaussianFusionOcc	C+R	33.08	12.31	9.18	7.68	0.00	32.05	0.00	14.14	5.75	0.00	0.00	20.89	31.17	2.52	12.87	15.10	17.29	28.35
GaussianFusionOcc	C+L	42.78	18.66	16.09	12.27	0.00	<b>39.82</b>	0.00	27.66	13.68	0.07	0.00	38.25	<b>40.10</b>	2.07	19.64	19.82	<b>29.55</b>	39.61
GaussianFusionOcc	C+L+R	42.51	18.45	12.15	11.47	0.00	39.77	0.00	<b>29.55</b>	<b>15.27</b>	0.04	0.00	37.08	37.13	2.58	19.94	20.63	29.42	40.10

389

391 limited signal in darkness. However, adding LiDAR or radar to camera-only significantly improves  
392 performance: camera-only achieves 32.05% IoU and 11.28% mIoU, while C+L gains +7.38 mIoU  
393 and C+R gains +1.03 mIoU. This demonstrates the robustness of multi-modal fusion when individual  
394 sensors are compromised, while leaving room for further research on utilizing more complex methods  
395 than simple concatenation in the fusion module.

396 The experimental results strongly suggest that GaussianFusionOcc shows significant performance  
397 gains in 3D semantic occupancy prediction, particularly in adverse weather and low-light scenarios.  
398

#### 4.5 EFFICIENCY ANALYSIS

400

401

Table 4: Efficiency comparison of multi-modal 3D semantic occupancy prediction on nuScenes validation set. Latency results marked with \* were taken from the paper that introduced the model and were measured on a different GPU.

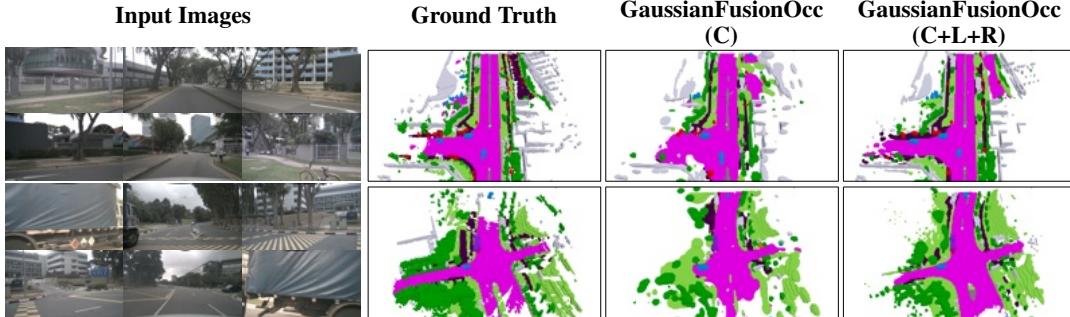
Method	Modality	IoU	mIoU	Params	Memory (GB)	Latency (ms)
L-CONet (Wang et al., 2023)	L	39.40	17.70	-	8.5	-
M-CONet (Wang et al., 2023)	C+L	39.20	24.70	137M	24	-
OccFusion (Ming et al., 2024)	C+R	33.97	20.73	92.71M	5.56	588*
OccFusion (Ming et al., 2024)	C+L	44.35	26.87	92.71M	5.56	591*
OccFusion (Ming et al., 2024)	C+L+R	44.66	27.30	114.97M	5.78	601*
GaussianFusionOcc	C	37.05	22.43	71.39M	2.44	282
GaussianFusionOcc	C+R	37.37	22.80	71.86M	2.58	315
GaussianFusionOcc	L	<b>45.32</b>	29.75	34.14M	0.49	179
GaussianFusionOcc	C+L	45.16	30.21	79.63M	2.61	460
GaussianFusionOcc	C+L+R	45.20	<b>30.37</b>	79.96M	2.90	480

415

416 Table 4 presents the efficiency comparison of multi-modal 3D semantic occupancy prediction  
417 methods, reporting the number of learnable parameters, memory usage, and latency. GaussianFusionOcc  
418 achieves state-of-the-art accuracy while substantially reducing computational and memory  
419 requirements compared to grid-based baselines. In the LiDAR-only setting, GaussianFusionOcc  
420 achieves 45.32% IoU and 29.75% mIoU, while requiring only 34.14 million parameters and 0.49GB  
421 of memory with inference time 179 ms. The GaussianFusionOcc further improves prediction  
422 performance in multi-modal settings, while staying significantly more efficient than comparable  
423 multi-modal methods. The memory and parameter savings of GaussianFusionOcc can be attributed  
424 to its adaptive, object-centric 3D Gaussian representation. This design enables the model to maintain  
425 a low memory footprint even as the number of sensor modalities increases. Latency measurements  
426 further highlight the practical advantages of GaussianFusionOcc, showing significantly faster  
427 inference than compared models. Detailed latency profiling reveals that feature extraction dominates  
428 the computational budget in both camera-only (81.8%) and full-sensor configurations (76.2%,  
429 distributed as 54.2% image, 21.6% LiDAR, 0.4% radar), while the four GaussianFusionBlocks  
430 contribute 13.7% and 15.6% respectively, and Gaussian-to-voxel splatting accounts for 4.5% and  
431 8.1%. This distribution demonstrates that the primary bottleneck lies in the modality-specific  
432 encoders rather than in Gaussian refinement or splatting operations, validating the efficiency of our  
433 Gaussian-based representation. A complete breakdown of component-wise latency measurements

432 is provided in [A.2](#). We report an additional ablation study on the number of Gaussians, the number  
 433 of channels for feature representation, and the initialization strategy, in the appendix [A.3](#).  
 434

435 **4.6 QUALITATIVE ANALYSIS**  
 436



439 **Figure 3: Occupancy visualizations on nuScenes.** Our model is able to predict both comprehensive  
 440 and realistic semantic 3D occupancy.  
 441

442 The analysis of the visualization results further underscores the benefits of multi-sensor fusion for 3D  
 443 semantic occupancy prediction. In the Figure [3](#), the model incorporating camera, LiDAR, and radar  
 444 data exhibits significantly better predictions in distant regions compared to the camera-only model,  
 445 with LiDAR and radar effectively compensating for camera limitations. Figure [5](#) demonstrates the  
 446 improvement in the allocation of the Gaussians when using the full sensor suite. Figure [4](#) comparing  
 447 the performance of camera-only, camera+radar, and camera+LiDAR models in nighttime scenarios,  
 448 clearly illustrates the critical role of additional sensors in overcoming the challenges posed by low-  
 449 light conditions. The inclusion of both radar and LiDAR data leads to substantial improvements in  
 450 the model’s ability to perceive the environment under these conditions, as vision-centric approaches  
 451 are known to perform poorly in nighttime scenarios due to the sensitivity of cameras to varying and  
 452 limited illumination. LiDAR’s 3D spatial awareness and radar’s ability to detect dynamic objects  
 453 contribute to more robust nighttime perception, with the most noticeable improvements in predicting  
 454 distant dynamic objects and reducing close object hallucinations.  
 455

456 **5 CONCLUSION**  
 457

458 In this paper, we introduced GaussianFusionOcc, a novel and seamless sensor fusion framework for  
 459 3D semantic occupancy prediction that leverages the efficiency and flexibility of the 3D Gaussian  
 460 representation. By fusing sensor information directly on learnable Gaussian primitives, it addresses  
 461 critical limitations of grid-based methods, achieving state-of-the-art performance (30.37 mIoU and  
 462 45.20 IoU) while significantly reducing memory usage, number of parameters, and latency. The  
 463 proposed modality-agnostic Gaussian encoder and fusion mechanism enable efficient integration of  
 464 camera, LiDAR, and radar data, exploiting their complementary strengths. The experiments demon-  
 465 strate the model’s resilience in adverse weather and nighttime conditions, achieving significantly  
 466 better results compared to previous state-of-the-art methods.  
 467

468 Even though sparse radar data provides complementary robustness in specific conditions, the  
 469 concatenation-based fusion mechanism generally shows modest improvements with occasional  
 470 degradation, suggesting that advanced strategies like cross-attention or confidence weighting could  
 471 better balance heterogeneous sensors. Additionally, evaluating the model on other datasets could give  
 472 valuable insight into the performance and robustness of the method. Finally, the framework’s reliance  
 473 on a predefined number of Gaussians limits adaptability in extremely sparse scenarios, potentially  
 474 benefiting from dynamic pruning and densification mechanisms.  
 475

476 Despite these limitations, GaussianFusionOcc demonstrates that Gaussian-space fusion represents a  
 477 promising paradigm shift for multi-modal perception, opening new research directions for primitive-  
 478 based fusion in autonomous driving tasks beyond occupancy prediction, including object tracking,  
 479 motion forecasting, and trajectory planning.  
 480

486 REFERENCES  
487

488 Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable  
489 surrogate for the optimization of the intersection-over-union measure in neural networks. In  
490 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4413–4421,  
491 2018.

492 Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liang, Qiang Xu, Anush  
493 Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for  
494 autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern  
495 recognition*, pp. 11621–11631, 2020.

496 Anh-Quan Cao and Raoul De Charette. Monoscene: Monocular 3d semantic scene completion.  
497 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.  
498 3991–4001, 2022.

499 Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor  
500 fusion framework for 3d detection. In *proceedings of the IEEE/CVF conference on computer vision  
501 and pattern recognition*, pp. 172–181, 2023.

502 Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable  
503 convolutional networks. In *Proceedings of the IEEE international conference on computer vision*,  
504 pp. 764–773, 2017.

505 Wanshui Gan, Fang Liu, Hongbin Xu, Ningkai Mo, and Naoto Yokoya. Gaussianocc: Fully self-  
506 supervised and efficient 3d occupancy estimation with gaussian splatting, 2024. URL <https://arxiv.org/abs/2408.11447>.

507 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image  
508 recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*,  
509 pp. 770–778, 2016.

510 Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for  
511 vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF conference on  
512 computer vision and pattern recognition*, pp. 9223–9232, 2023.

513 Yuanhui Huang, Amonnut Thammatadatrakoon, Wenzhao Zheng, Yunpeng Zhang, Dalong Du, and  
514 Jiwen Lu. Probabilistic gaussian superposition for efficient 3d occupancy prediction. *arXiv preprint  
515 arXiv:2412.04384*, 2024a.

516 Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Gaussianformer: Scene  
517 as gaussians for vision-based 3d semantic occupancy prediction. In *European Conference on  
518 Computer Vision*, pp. 376–393. Springer, 2024b.

519 Haoyi Jiang, Tianheng Cheng, Naiyu Gao, Haoyang Zhang, Tianwei Lin, Wenyu Liu, and Xinggang  
520 Wang. Symphonize 3d semantic scene completion with contextual instance queries. In *Proceedings  
521 of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20258–20267,  
522 2024.

523 Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting  
524 for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL  
525 <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>.

526 Youngseok Kim, Sanmin Kim, Jun Won Choi, and Dongsuk Kum. Craft: Camera-radar 3d object  
527 detection with spatio-contextual fusion transformer. In *Proceedings of the AAAI Conference on  
528 Artificial Intelligence*, volume 37, pp. 1160–1168, 2023.

529 Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Point-  
530 pillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF  
531 conference on computer vision and pattern recognition*, pp. 12697–12705, 2019.

532 Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng,  
533 and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic  
534 scene completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern  
535 recognition*, pp. 9087–9098, 2023a.

540 Zhiqi Li, Zhiding Yu, David Austin, Mingsheng Fang, Shiyi Lan, Jan Kautz, and Jose M Alvarez.  
 541 Fb-occ: 3d occupancy prediction based on forward-backward view transformation. *arXiv preprint*  
 542 *arXiv:2307.01492*, 2023b.

543 Zhiqi Li, Wenhui Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng  
 544 Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal  
 545 transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

546 Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie.  
 547 Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer*  
 548 *vision and pattern recognition*, pp. 2117–2125, 2017.

549 Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han.  
 550 Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023*  
 551 *IEEE international conference on robotics and automation (ICRA)*, pp. 2774–2781. IEEE, 2023.

552 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint*  
 553 *arXiv:1711.05101*, 2017.

554 Yuhang Lu, Xinge Zhu, Tai Wang, and Yuexin Ma. Octreeocc: Efficient and multi-granularity  
 555 occupancy prediction using octree queries. *Advances in Neural Information Processing Systems*,  
 556 37:79618–79641, 2024.

557 Ruihang Miao, Weizhou Liu, Mingrui Chen, Zheng Gong, Weixin Xu, Chen Hu, and Shuchang  
 558 Zhou. Occdepth: A depth-aware method for 3d semantic scene completion. *arXiv preprint*  
 559 *arXiv:2302.13540*, 2023.

560 Zhenxing Ming, Julie Stephany Berrio, Mao Shan, and Stewart Worrall. Occfusion: Multi-sensor  
 561 fusion framework for 3d semantic occupancy prediction. *IEEE Transactions on Intelligent Vehicles*,  
 562 2024.

563 Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew  
 564 Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *European conference*  
 565 *on computer vision*, pp. 414–431. Springer, 2020.

566 Ramin Nabati and Hairong Qi. Centerfusion: Center-based radar and camera fusion for 3d object  
 567 detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*,  
 568 pp. 1527–1536, 2021.

569 Jiabao Wang, Zhaojiang Liu, Qiang Meng, Liujiang Yan, Ke Wang, Jie Yang, Wei Liu, Qibin  
 570 Hou, and Ming-Ming Cheng. Opus: occupancy prediction using a sparse set. *arXiv preprint*  
 571 *arXiv:2409.09350*, 2024.

572 Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage  
 573 monocular 3d object detection. In *Proceedings of the IEEE/CVF international conference on*  
 574 *computer vision*, pp. 913–922, 2021.

575 Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du,  
 576 Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic  
 577 occupancy perception. In *Proceedings of the IEEE/CVF International Conference on Computer*  
 578 *Vision*, pp. 17850–17859, 2023.

579 Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-  
 580 camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF*  
 581 *International Conference on Computer Vision*, pp. 21729–21740, 2023.

582 Yichen Xie, Chenfeng Xu, Marie-Julie Rakotosaona, Patrick Rim, Federico Tombari, Kurt Keutzer,  
 583 Masayoshi Tomizuka, and Wei Zhan. Sparsefusion: Fusing multi-modal sparse representations for  
 584 multi-sensor 3d object detection. In *Proceedings of the IEEE/CVF International Conference on*  
 585 *Computer Vision*, pp. 17591–17602, 2023.

586 Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable  
 587 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the*  
 588 *IEEE/CVF conference on computer vision and pattern recognition*, pp. 20331–20341, 2024.

594 Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based  
 595 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF International Conference on*  
 596 *Computer Vision*, pp. 9433–9443, 2023.

597  
 598 Yin Zhou and Oncel Tuzel. Voxelnets: End-to-end learning for point cloud based 3d object detection.  
 599 In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4490–4499,  
 600 2018.

601  
 602 Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr:  
 603 Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

## 605 A APPENDIX

### 606 A.1 ADDITIONAL VISUALIZATIONS

607 The visualizations in Figure 4 and Figure 5 demonstrate the importance of sensor fusion for effective  
 608 allocation of the semantic Gaussians, especially in low-light scenarios.

### 612 A.2 INFERENCE LATENCY BREAKDOWN

614 To provide deeper insights into the computational characteristics of GaussianFusionOcc, we con-  
 615 ducted comprehensive latency profiling of the inference pipeline across different sensor configura-  
 616 tions. All measurements were performed on an NVIDIA RTX A6000 GPU with a batch size of 1, averaged  
 617 over 100 inference runs on the nuScenes validation set. The pipeline is decomposed into three main  
 618 stages: (1) modality-specific feature extraction using the image backbone (ResNet101-DCN + FPN),  
 619 LiDAR encoder (VoxelNet + FPN), and radar encoder (PointPillars), (2) the four sequential Gaus-  
 620 sianFusionBlocks that iteratively refine Gaussian properties through modality-agnostic deformable  
 621 attention and fusion, and (3) the Gaussian-to-voxel splatting module that transforms the refined  
 622 Gaussian representation into the final dense occupancy prediction.

623 In the camera-only variant, feature extraction from the six surround-view cameras constitutes the  
 624 dominant computational cost at approximately 81.8% of total inference time. The four GaussianFu-  
 625 sionBlocks collectively account for roughly 13.7%, while the Gaussian-to-voxel splatting module  
 626 contributes 4.5%. This distribution reveals that the computational bottleneck lies primarily in the  
 627 image feature extraction backbone rather than in the Gaussian refinement or splatting operations.  
 628 The relatively modest contribution of the GaussianFusionBlocks highlights the efficiency of our  
 629 modality-agnostic deformable attention mechanism. Despite processing 6,400 Gaussians through four  
 630 refinement iterations with 128-channel feature representations, the Gaussian refinement stage adds  
 631 minimal overhead. This efficiency stems from the sparse nature of Gaussian-based representations,  
 632 where attention operations focus computational resources on occupied regions rather than processing  
 633 dense grids uniformly. The Gaussian-to-voxel splatting module’s small contribution further validates  
 634 the computational advantages of our approach compared to dense voxel-based methods that require  
 635 expensive 3D convolutions throughout the entire pipeline.

636 When integrating camera, LiDAR, and radar inputs, the latency distribution shifts to reflect the  
 637 additional computational cost of multi-modal feature extraction and fusion. Collectively, feature  
 638 extraction from all modalities accounts for approximately 76.2% of total inference time, distributed  
 639 as follows: image extraction dominates at 54.2%, LiDAR extraction contributes 21.6%, while radar  
 640 processing introduces negligible overhead at 0.4%. This demonstrates that radar is computationally  
 641 attractive for enhancing robustness without significant latency penalties. The GaussianFusionBlocks’  
 642 contribution increases from 13.7% to 15.6% in the multi-modal setting, reflecting the additional  
 643 computational cost of extracting per-Gaussian features from multiple sensor modalities through  
 644 deformable attention and fusing them via the concatenation-based fusion module. This increase  
 645 is modest considering that the blocks now process features from three distinct sensor types with  
 646 different spatial resolutions and representational characteristics. The Gaussian-to-voxel splatting  
 647 increases from 4.5% to approximately 8.1%, suggesting that splatting from richer multi-modal  
 648 Gaussian representations with more refined geometric and semantic properties requires additional  
 649 computation, yet remains significantly more efficient than grid-based alternatives.

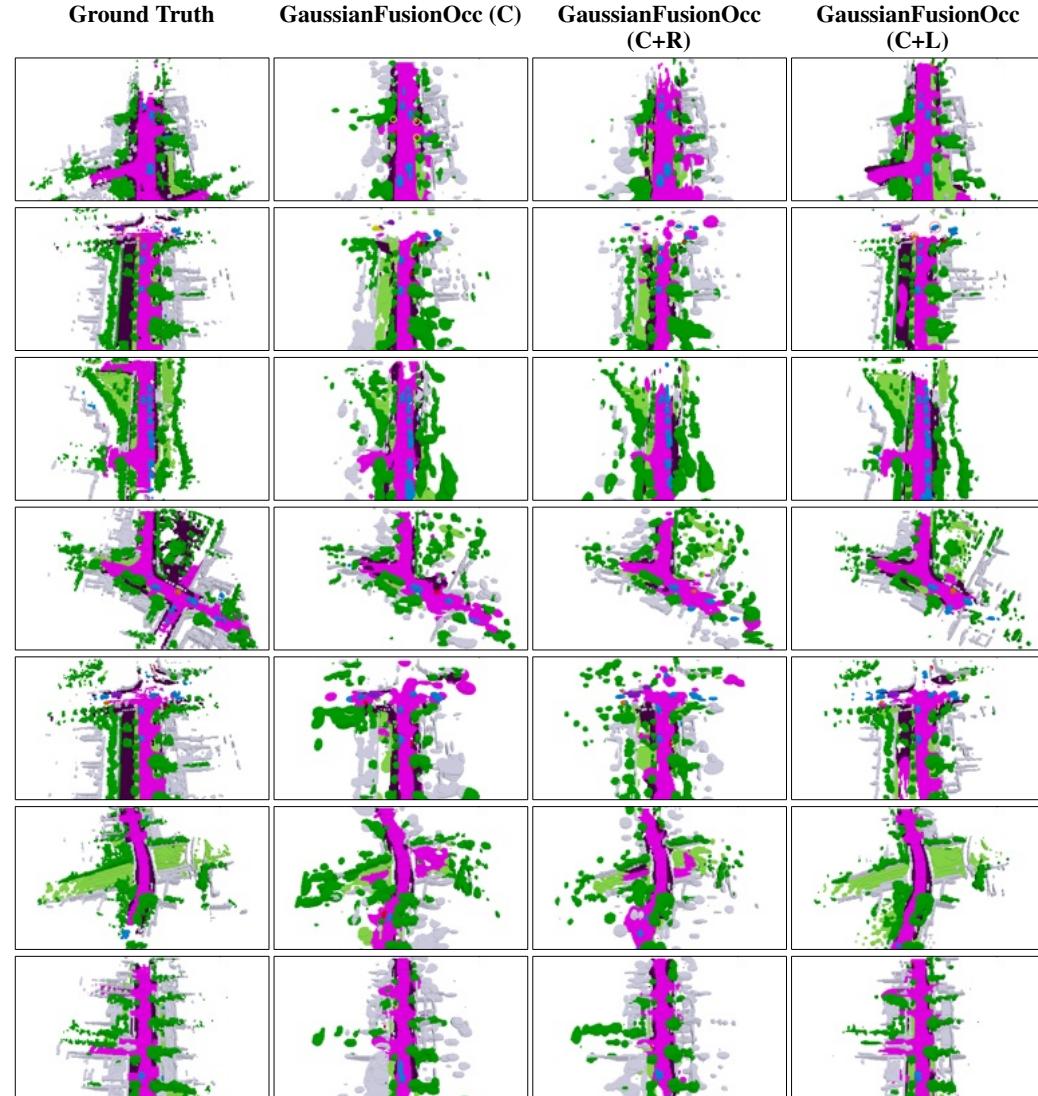


Figure 4: **Occupancy visualizations on nuScenes night scenes.** Visualization shows the importance of additional sensors in low-light conditions.

The profiling results reveal several important insights. First, feature extraction from modality-specific encoders constitutes the primary computational bottleneck across all configurations (81.8% camera-only, 76.2% full-suite), indicating that further optimization efforts should focus on efficient backbone architectures rather than the Gaussian-based components. Second, the GaussianFusion-Blocks demonstrate remarkable efficiency, adding only 13.7-15.6% overhead despite performing complex multi-modal fusion and iterative refinement operations. This validates our design choice of sparse Gaussian representation combined with deformable attention, which avoids the computational redundancy inherent in dense grid-based approaches. Third, the Gaussian-to-voxel splatting module's minimal contribution (4.5-8.1%) confirms that our splatting-based final prediction stage is substantially more efficient than methods that maintain dense volumetric representations throughout the entire pipeline.

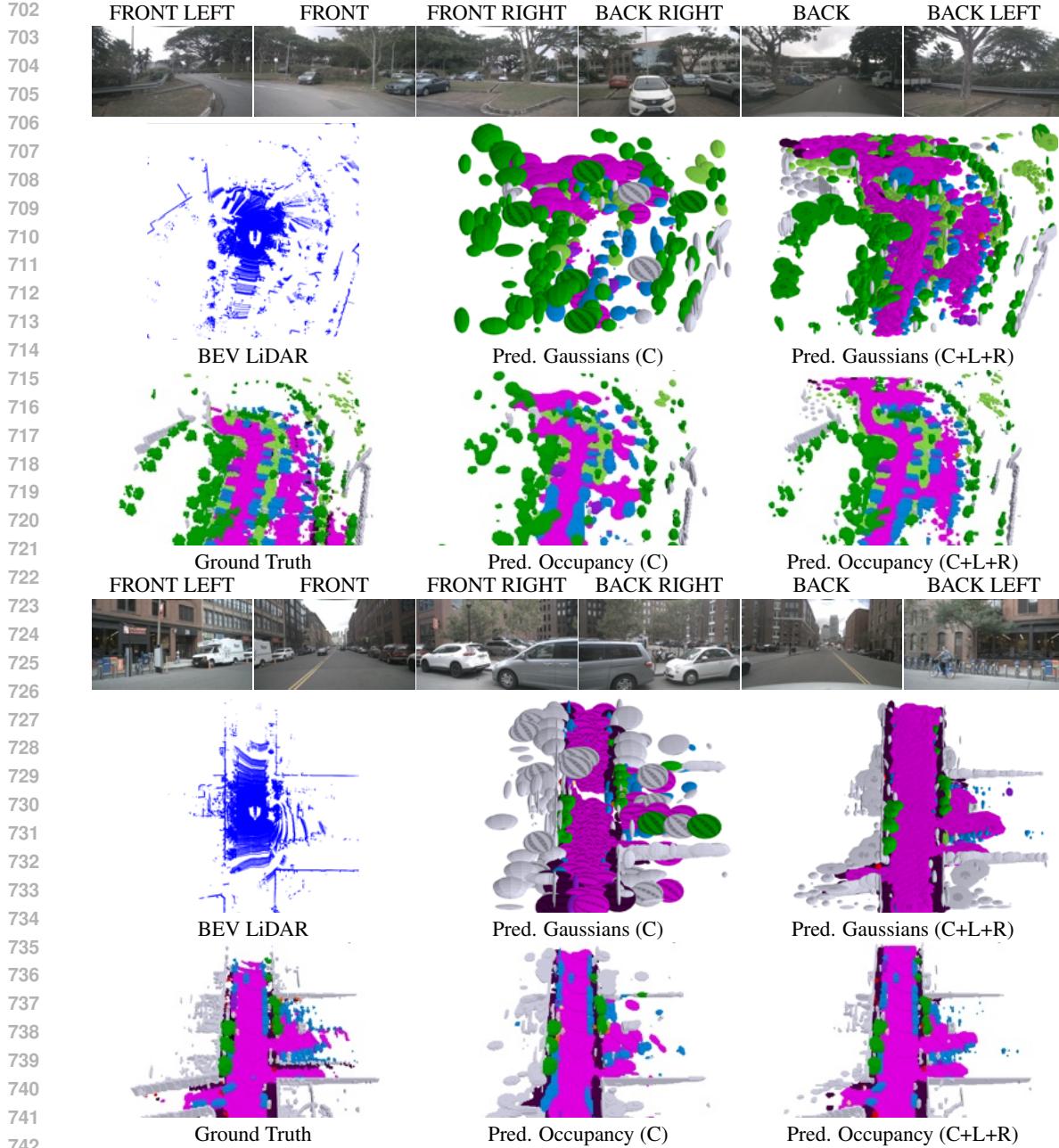


Figure 5: **Occupancy and Gaussian representation visualizations on nuScenes dataset.** Visualizations demonstrate the importance of additional sensors for the allocation of the Gaussians and final occupancy prediction.

### A.3 ABLATION STUDY

We conduct ablation studies to analyze the impact of key design choices and components of our GaussianFusionOcc framework. The latency and memory are tested on an NVIDIA RTX A6000 GPU for all the experiments.

**Number of Gaussians:** Table 5 shows the influence of the number of Gaussians on efficiency and performance. We observe improvement in performance as the number of Gaussians increases. This is due to the enhanced ability to represent finer details with more Gaussians. The number of

756  
 757 Table 5: Ablation on number of Gaussians. The increased number of Gaussians improves the  
 758 performance, without significantly increasing memory usage and the number of parameters, but it  
 759 increases the latency.

Method	Modality	IoU	mIoU	Gaussians	Params	Memory (GB)	Latency (ms)
GaussianFusionOcc	C+L	45.16	30.21	6400	79.63M	2.61	460
GaussianFusionOcc	C+L	45.74	30.83	25600	80M	2.62	547

760  
 761  
 762 Table 6: Ablation on number of channels for feature representation. The increased number of channels  
 763 improves the performance, but significantly increases the number of parameters and memory usage.

Method	Modality	IoU	mIoU	Channels	Params	Memory (GB)	Latency (ms)
GaussianFusionOcc	C+L+R	45.20	30.37	128	79.96M	2.90	480
GaussianFusionOcc	C+L+R	45.69	30.85	192	115M	5.41	486

763  
 764  
 765 Table 7: Ablation on initialization strategy. Probabilistic initialization is taken from [Huang et al.](#)  
 766 [\(2024a\)](#).

Method	Modality	IoU	mIoU	Initinalization	Params	Memory (GB)	Latency (ms)
GaussianFusionOcc	C+L+R	45.31	30.07	Random	79.78M	2.62	468
GaussianFusionOcc	C+L+R	45.20	30.37	Learnable	79.96M	2.90	480
GaussianFusionOcc	C+L+R	44.52	30.13	Probabilistic	79.85M	3.05	844

771  
 772 parameters and memory usage are not significantly increased because they are mostly influenced by  
 773 sensor-specific encoders, which are not influenced by the number of Gaussians.

774  
 775 **Number of channels:** The influence of the number of channels used for extracted sensor features  
 776 and per-Gaussian features is demonstrated in Table 6. An Increased number of channels improves  
 777 the prediction performance with a significant efficiency degradation. The increase in memory and  
 778 parameter number can be attributed to the increased size of sensor feature extractors, as they are also  
 779 influenced by the number of channels.

780  
 781 **Initialization strategy:** We report the influence of initialization strategy on the performance and  
 782 efficiency in Table 7. Learnable initialization shows the highest mIoU with a slightly higher number of  
 783 parameters, memory usage, and latency, compared to random initialization. Probabilistic initialization,  
 784 proposed by GaussianFormer-2 ([Huang et al., 2024a](#)), degrades the performance of the model while  
 785 significantly slowing down the inference.

#### 786 A.4 EVALUATION METRICS

787  
 788 To evaluate our method and compare the results with other state-of-the-art methods, we use Intersec-  
 789 tion over Union (IoU) and mean Intersection over Union (mIoU) metrics:

$$790 \quad IoU = \frac{TP_{\neg c_0}}{TP_{\neg c_0} + FP_{\neg c_0} + FN_{\neg c_0}} \quad (7)$$

$$791 \quad mIoU = \frac{1}{|C|} \sum_{c \in C} \frac{TP_c}{TP_c + FP_c + FN_c} \quad (8)$$

802 where TP, FP, FN denote the number of true positive, false positive, and false negative predictions,  
 803 and C,  $c_0$  denote the set of classes without the empty class, and the empty class, respectively.

#### 804 A.5 THE USE OF LARGE LANGUAGE MODELS

805  
 806 In the preparation of this paper, we utilized Perplexity as a general-purpose writing assist tool.  
 807 Specifically, the LLM was employed to improve grammar, refine wording, and enhance the overall  
 808 clarity and flow of the written content. The LLM was not involved in research ideation, methodology  
 809 design, experimental planning, or the interpretation of results. All scientific contributions, including

810 the conceptualization of GaussianFusionOcc, the development of the fusion framework, experimental  
811 design, and analysis of results, were entirely conducted by the authors. The authors take full  
812 responsibility for all content presented in this work, including any text that was refined with LLM  
813 assistance.

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863