IEEE Open Journal of Vehicular Technology

Received 24 March 2025; accepted 20 April 2025. Date of publication 23 April 2025;
date of current version 21 May 2025. The review of this article was coordinated by Editor Levent Guvenc.

Digital Object Identifier 10.1109/OJVT.2025.3563677

Leveraging Frozen Foundation Models and Multimodal Fusion for BEV Segmentation and Occupancy Prediction

SEAMIE HAYES [1,2,3] (Graduate Student Member, IEEE), GANESH SISTU[1,3], AND CIARÁN EISING [1,2,3] (Senior Member, IEEE)

[1]Department of Electronic and Computer Engineering, University of Limerick, V94 T9PX Limerick, Ireland
[2]SFI CRT Foundations in Data Science, University of Limerick, V94 T9PX Limerick, Ireland
[3]Data Driven Computer Engineering (D [2] iCE) Research Centre, University of Limerick, V94 T9PX Limerick, Ireland

CORRESPONDING AUTHOR: SEAMIE HAYES (e-mail: hayes.seamie@ul.ie).

This work was supported by Science Foundation Ireland under Grant 18/CRT/6049.

ABSTRACT In Bird's Eye View perception, significant emphasis is placed on deploying well-performing, convoluted model architectures and leveraging as many sensor modalities as possible to reach maximal performance. This paper investigates whether foundation models and multi-sensor deployments are essential for enhancing BEV perception. We examine the relative importance of advanced feature extraction versus the number of sensor modalities and assess whether foundation models can address feature extraction limitations and reduce the need for extensive training data. Specifically, incorporating the self-supervised DINOv2 for feature extraction and Metric3Dv2 for depth estimation into the Lift-Splat-Shoot framework results in a 7.4 IoU point increase in vehicle segmentation, representing a relative improvement of 22.4%, while requiring only half the training data and iterations compared to the original model. Furthermore, using Metric3Dv2's depth maps as a pseudo-LiDAR point cloud within the Simple-BEV model improves IoU by 2.9 points, marking a 6.1% relative increase compared to the Camera-only setup. Finally, we extend the famous Gaussian Splatting BEV perception models, GaussianFormer and GaussianOcc, through multimodal deployment. The addition of LiDAR information in GaussianFormer results in a 9.4-point increase in mIoU, a 48.7% improvement over the Camera-only model, nearing state-of-the-art multimodal performance even with limited LiDAR scans. In the self-supervised GaussianOcc model, incorporating LiDAR leads to a 0.36-point increase in mIoU, representing a 3.6% improvement over the Camera-only model. This limited gain can be attributed to the absence of LiDAR encoding and the self-supervised nature of the model. Overall, our findings highlight the critical role of foundation models and multi-sensor integration in advancing BEV perception. By leveraging sophisticated foundation models and multi-sensor deployment, we can further model performance and reduce data requirements, addressing key challenges in BEV perception.

INDEX TERMS Bird's eye view, foundation model, LiDAR, multimodal, semantic occupancy.

I. INTRODUCTION

Autonomous vehicle technology has rapidly evolved, with machine learning advancements driving the development of sophisticated vehicle segmentation and perception models. The task of achieving accurate 3D perception from RGB images is complex, primarily due to the inherent challenge of depth estimation from flat 2D images. This issue has led to significant research integrating additional sensor modalities, such as LiDAR and radar, to supplement camera data. Specifically, the integration of these modalities has shown considerable performance improvements due to their complementary nature. Cameras offer detailed 2D color images that are useful for vehicle and lane segmentation, while LiDAR and radar contribute point clouds crucial for depth estimation. As various models show, these modalities deliver superior performance compared to single-modal setups. For

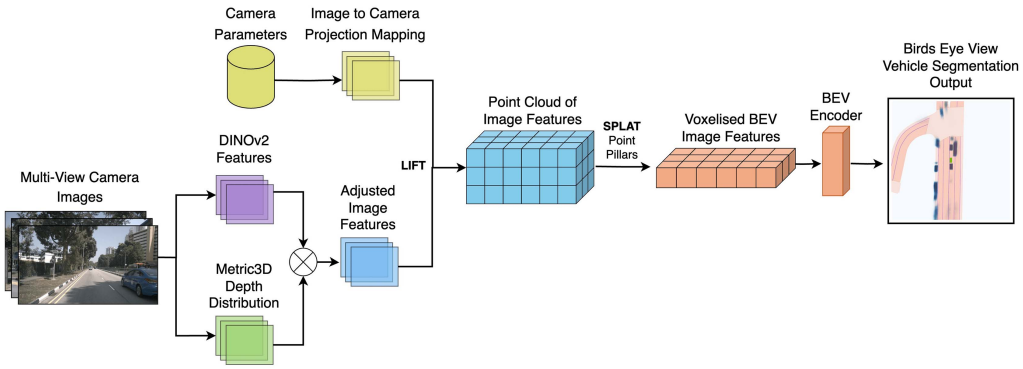

FIGURE 1. Our Modified Lift-Splat-Shoot Architecture: Incorporates DINOv2 for image feature extraction and Metric3Dv2 for depth estimation, replacing the original EfficientNet-b0 backbone.

example, in the Simple-BEV model, incorporating LiDAR with camera data boosts performance metrics by 28.3%, and combining camera data with radar yields a 17.5% increase compared to using the camera alone [1]. Simple-BEV leverages LiDAR and radar data in their raw occupancy forms, bypassing the explicit encoding processes typically used in other Bird's Eye View (BEV) models, such as those employing the PointPillars encoder [2] and others [3], [4]. However, it's important to note that this data undergoes implicit encoding through the BEV compressor [1]. In multimodal fusion, a raw occupancy representation highlights the advantages of unprocessed depth information.

Regardless of the method used, it still stands that training these models for BEV perception remains a challenge due to the large datasets required and the extensive computational resources involved. The nuScenes dataset [5], a standard benchmark, contains 700 annotated driving scenes, each with 40 keyframes, which are used during the training phase of a model. For example, Lift-Splat-Shoot (LSS) [6] requires 40 epochs to reach a reasonable Intersection over Union (IoU) score of approximately 33, which was the state-of-the-art (SOTA) performance when it was published. Reducing the training time and data requirements while maintaining high performance would be significant. One potential approach to this challenge is using foundation models, which are large pre-trained models that generalize very well across various tasks. The advancement of foundation models cannot be understated, with the advent of models including EfficientSAM [7] and Depth Pro [8], which show substantial improvements over their predecessors. EfficientSAM significantly reduces both total model parameters and inference time by 20-fold compared to the SAM model [9], with only a slight decrease in average precision. Depth Pro offers runtime improvements over its predecessors (such as Metric3Dv2) by providing faster inference times—approximately 350 ms for Depth Pro compared to 1300 ms for Metric3Dv2 on an NVIDIA V100 GPU [8].

The contribution of this paper to BEV perception lies in integrating foundation models and multi-sensor fusion to push the boundaries of autonomous vehicle perception. Firstly, starting with BEV vehicle segmentation, two prominent models are modified: Lift-Splat-Shoot and Simple-BEV. Lift-Splat-Shoot was chosen due to its being a well-established framework, and Simple-BEV was chosen for its ability to utilize LiDAR and radar in conjunction with the camera modality. Specifically, for LSS, DINOv2 [10] is deployed for image feature extraction and Metric3Dv2 [11] for depth estimation. DINOv2 extracts high-quality image features, and Metric3Dv2 efficiently projects these features into 3D space. The modified architecture is illustrated in Fig. 1. Simple-BEV is modified in a multimodal approach to replace traditional LiDAR data with a point cloud generated from the depth maps of Metric3Dv2, named pseudo-LiDAR [12], for improvement over the Camera-only model whilst retaining the single-modal aspect. Maintaining these foundation models in their frozen state eliminates the need for fine-tuning parameters while still enhancing performance metrics, demonstrating their generalizability and relevance to various tasks, such as BEV perception. Traditionally, backbones in BEV perception models such as EfficientNet [13] and ResNet [14] are fine-tuned, which is necessary for even achieving adequate performance.

Finally, multimodal fusion is further explored in 3D Gaussian Splatting-based semantic occupancy prediction models: GaussianFormer [15] and GaussianOcc [16]. For GaussianFormer, instead of concatenating point cloud features/occupancy with image-derived voxel features, as seen in traditional multimodal fusion methods [1], [3], [17], the initial Gaussian *xyz* properties are replaced with point cloud data. This allows us to leverage the Gaussian properties more effectively, iteratively refining them to create a sparse and efficient 3D representation of the scene. The chosen approach significantly boosts semantic and occupancy prediction performance. Lastly, the implementation of point cloud data within the self-supervised GaussianOcc model is investigated. The approach in this paper mirrors that of traditional multimodal fusion methods, whereby raw occupancy data is concatenated directly with the image voxel features, and our work shows interesting results considering multimodal fusion in self-supervised models is not well-researched.

This research aims to enhance the performance of existing BEV perception models to their full potential, demonstrating that developing new models can often be redundant if existing

TABLE 1. Summary of Our Modifications and Their Impacts on Model Performance: A Tick Indicates the Implementation of a Model/Modality Within the Architecture. Metric3Dv2 Denotes the Use of Depth Maps, While Pseudo-LiDAR Refers to Point Clouds Derived From These Depth Maps

Model Architecture	DINOv2	Metric3Dv2	LiDAR	Radar	Pseudo-LiDAR	Result versus Baseline
Lift-Splat-Shoot	✓	✓				+7.4 IoU increase
Simple-BEV					✓	+2.9 IoU increase compared to Camera-only
GaussianFormer			✓	✓	✓	+9.4 mIoU, +1.7 mIoU, +1.1 mIoU increase for LiDAR, radar, and pseudo-LiDAR, respectively
GaussianOcc			✓	✓	✓	+0.36 mIoU, -0.03 mIoU, +0.04 mIoU change for LiDAR, radar, and pseudo-LiDAR respectively

ones are not fully optimized. The specific modifications made to the models and the corresponding results are detailed in Table 1. Specifically, in this paper, the following contributions are made:

- The implementation of foundation models DINOv2 and Metric3Dv2 in Lift-Splat-Shoot reduces training data requirements and enhances model performance.
- The novel integration of Metric3Dv2 depth maps projected into a 3D pseudo-LiDAR point cloud, substituting LiDAR data in Simple-BEV. This delivers increased performance over the baseline Camera-only model whilst retaining the single-modal nature of the model.
- Multimodal integration in the semantic occupancy prediction model GaussianFormer, which rivals the performance of SOTA multimodal models.
- The first exploration of a multimodal, self-supervised approach in the semantic occupancy predictor GaussianOcc, yielding a minimal increase in performance metrics.

The preliminary findings of this study were previously published in [18]. This paper builds upon that research and introduces significant enhancements, including:

- Broadening the scope to include semantic occupancy prediction in BEV research with models GaussianFormer and GaussianOcc.
- Advancing BEV methods through a multimodal fusion of Camera, LiDAR, radar, and pseudo-LiDAR, achieving near state-of-the-art performance.
- Providing a deeper analysis of the results, model parameters, and the limitations of this work.

This paper is organized as follows: Section II reviews related work, including BEV vehicle segmentation and the foundation models utilized in this study. Section III provides a detailed explanation of the modifications made to the four chosen model architectures. Section IV presents both quantitative and qualitative results derived from the methodologies employed, along with a discussion of results with detailed limitations. Finally, the paper concludes with Section V.

II. LITERATURE REVIEW

This section discusses various related works to provide insight into the motivation behind this research and to explore the models utilized in the paper.

A. BEV VEHICLE SEGMENTATION

For vehicle segmentation, representing a scene in Bird's Eye View has recently been a commonly researched method for autonomous vehicle perception due to its more simplistic representation of a scene [19], [20]. BEV detection methods often rely on the deployment of large CNN architectures such as a ResNet backbone [14] to detect objects in images, and the more complex deployment of transformer models for better retention of temporal and spatial information has been thoroughly explored through attention [4], [21], [22]. Each BEV perception model competes for top performance on real-world datasets [5], [23], [24] or even virtual datasets such as Virtual KITTI [25].

For the experiments in this paper, two well-established BEV model architectures are utilized: Lift-Splat-Shoot (LSS) [6] and Simple-BEV [1]. LSS has been extensively studied and, most notably, utilizes probabilistic depth-based splatting to lift 2D image features to 3D, with downstream tasks including vehicle segmentation and ego-vehicle trajectory prediction. In this paper, the quality of extracted image features is improved using high-quality DINOv2 image features, and the depth-based splatting method is improved with high-resolution Metric3Dv2 depth maps. The second model, Simple-BEV, improves over LSS's splatting method via bilinear sampling, a dense parameter-free splatting method. Additionally, Simple-BEV improves over the EfficientNet-b0 backbone [13] via the deployment of the more powerful ResNet-101 [14]. Additionally, Simple-BEV uses LiDAR and radar point cloud information from the nuScenes dataset to increase vehicle segmentation quality due to inherent depth information. Here, a pseudo-LiDAR point cloud will be integrated to improve over the Camera-only model, which will be detailed later.

B. SEMANTIC OCCUPANCY PREDICTION

Compared to a BEV representation, semantic occupancy aligns much more accurately with how one perceives a driving scenario, which is 3D. With the advent of the creation of ground truth semantic occupancy datasets [26], [27], [28], [29], many models have been developed which all aim for top performance on these datasets, building on the already existing BEV perception methods while also creating new 3D centric methods [30], [31], [32], [33], [34], [35], [36]. One

issue that many of these models face is memory constraints, mainly due to their dense voxel representation of the sparse scenes.

In the area of real-time radiance field rendering, 3D Gaussian Splatting [37] has taken over, replacing its predecessor Neural Radiance Fields (NeRF) [38]. This can be attributed to the 98% reduction in training time, a 0.9 increase in the PSNR quality metric, and real-time rendering capabilities, which render frames over 1300 times faster [37]. Following this, a semantic occupancy prediction model GaussianFormer [15] was developed, which utilizes these 3D Gaussians for a sparse representation of the driving scene, ditching the need for a dense voxel of features. By starting with just 25,600 Gaussians equipped with position (xyz), scale, opacity, rotation, and semantic data and subsequently refining them through processes such as deformable attention and feed-forward networks, this model achieves near SOTA results. We further improve this model via a unique implementation of point cloud information from LiDAR, radar, and pseudo-LiDAR, which is then evaluated on the SurroundOcc dataset [26]. We will compare this multimodal GaussianFormer model to three distinct, well-performing alternatives: OccFusion, Co-Occ, and DAOcc.

OccFusion processes surround-view images and LiDAR/Radar point clouds through separate 2D and 3D backbones, extracting multi-scale features. These are fused at each scale via dynamic 3D/2D modules and a global-local attention mechanism. The final 3D volumes are upsampled with skip connections under multi-scale supervision, producing feature representations [32]. Co-Occ employs a GSFusion module to leverage camera features for semantics and LiDAR for geometric accuracy. During training, an implicit volume rendering regularization enhances LiDAR-camera fusion, improving 3D semantic prediction [39]. DAOcc begins with BEV View Range Extension to expand spatial context, followed by feature extraction from camera and LiDAR data, which are fused before passing through an occupancy prediction head [17].

Recently, self-supervised models have gained significant traction by eliminating the need for manually annotated ground truth labels. Yet, they continue to achieve remarkable performance despite this limitation [10], [40], [41]. GaussianOcc is a self-supervised model that represents the scenes via a dense voxel of features, utilizing 3D Gaussians for loss computation via view synthesis [16]. This paper explores the implementation of point cloud information in the GaussianOcc model.

C. FOUNDATION MODELS

Larger processing power lends itself to larger models. Following the recent machine learning boom, no models fit this statement better than foundation models. A foundation model is trained on a large dataset, often millions if not tens of millions of data points, to create a model that generalizes well across various tasks such as image classification or segmentation [7], [9], [42], [43]. Two foundation models will be used in this research: DINOv2 [10], and Metric3Dv2 [11]. DINOv2 is a self-supervised model trained on approximately 140 million images from diverse datasets, which extracts a wide array of features from an image, all in zero-shot inference. DINOv2 utilizes Vision Transformers (ViTs) [44], dividing an image into patches where each patch is a 14×14 pixel region and is given a patch embedding that describes the features present. For the Small model, this patch embedding is of size 384. In this paper, the frozen model is utilized, which requires no computation of gradients, thus speeding up the training process. For implementation in LSS, a single convolutional layer is implemented that downsamples the image feature vectors from the DINOv2 embeddings size to 64 for compatibility with the EfficientNet BEV encoder. As shown in Table 2, DINOv2's extensive parameter set and robust pretraining regimen set it apart from standard CNN architectures, along with its transformer-based architecture, shown to effectively capture long-range dependencies [45].

TABLE 2. Model Comparison of DINOv2 and Two CNN Bakcbones: Dataset Total Indicates the Number of Images on Which Each Model Was Pretrained. Note That the Embedding Dimensions of EfficientNet-B0 and ResNet-101 Can Be Changed

Model	*Model Parameters*	*Dataset Total*	*Embedding Size*
DINOv2 Small	21M	142M	384
DINOv2 Base	86M	142M	768
DINOv2 Large	300M	142M	1024
DINOv2 Giant	1,100M	142M	1536
EfficientNet-b0	5.3M	1.2M	64
ResNet-101	44.5M	1.2M	128

Secondly, Metric3Dv2 is deployed, which is pretrained on 8 million images and estimates pixel-level metric depth for a given image. The model utilizes standardized canonical camera space, featuring a standard origin at the optical center, normalized scale of intrinsics, and axis alignment relative to the camera orientation. This allows it to overcome the metric ambiguity typically associated with standard monocular cameras. In our experiments, this depth information will be utilized to aid the projection of image features into 3D for LSS and, in addition to this, create a point cloud for emulating LiDAR point cloud data in Simple-BEV, GaussianFormer, and GaussianOcc.

For depth estimation models, several key nuances must be considered. The three primary factors are metric versus relative depth estimation, zero-shot estimation, and inference time. We analyze five candidate models, with their configurations detailed in Table 3, based on information from [46], [47]. These factors are particularly relevant to the selection of Metric3Dv2 for deployment in this paper.

First, not all models provide metric depth; some instead produce absolute depth, which is not suitable for our purposes [46], [47], [48]. While these models can be adapted for metric depth estimation through fine-tuning on specific

TABLE 3. Comparison of Depth Estimation Models: Inference Time Is Approximate as All Models Output Different Depth Map Sizes. Image Input Size to MonoDepth Is 512 × 256 pixels, Whereas It Is 640 × 480 pixels to the Remaining Four Models

Model	*Metric Depth*	*Model Parameters*	*Inference Time*
MonoDepth	✗	28M	35ms
DPT	✗	123M	33ms
DepthAnythingv2	✗	335M	91ms
Metric3Dv2	✓	1,378M	1299ms
DepthPro	✓	504M	341ms

datasets, this process requires access to ground truth depth maps, which are unavailable for nuScenes. Furthermore, our objective is to perform zero-shot metric depth estimation—demonstrating the generalizability of foundation models without fine-tuning the model on the dataset—thereby excluding the previously mentioned models from consideration.

As a result, our choices are limited to metric depth estimation models capable of generalizing diverse images well. These models tend to be large, leading to slower inference times. Among the viable candidates, Metric3Dv2 [11], and Depth Pro [8] stand out. However, we acknowledge that their inference times are suboptimal for real-time autonomous vehicle applications involving multi-surround-view camera setups, as they exceed typical human reaction times, which range from 400 ms to 1300 ms, with a mean of approximately 680 ms in driving scenarios [49]. Despite this limitation, we select Metric3Dv2 due to its superior spatial consistency across camera views compared to Depth Pro, as observed in our testing.

D. MULTIMODAL MODELS

Multimodal models have been proven to outperform unimodal models in the Birds Eye View vehicle segmentation space, particularly models that utilize LiDAR information (e.g., [1], [3]). The advantage of the pseudo-LiDAR point cloud generated by our approach over traditional LiDAR and radar is that it originates from camera data, thus eliminating the need for deploying additional sensors, which reduces costs and avoids complications with sensor synchronization. This approach will be implemented in several models to outperform the Camera-only model while attempting to rival the Camera+LiDAR and Camera+Radar models.

With the emergence of more semantic occupancy prediction models, multimodal fusion is becoming more common with various models utilizing multiple modalities, namely the conjunction of camera and LiDAR data [17], [32], [39], [50]. GaussianFormer and GaussianOcc rely solely on camera data; however, particularly in the case of GaussianFormer, the use of 3D Gaussians to model the scene represents spatial information through *xyz* coordinates. This continuous nature makes it well-suited for point cloud data, as both share a common spatial representation. Additionally, already existing multimodal models employ a LiDAR encoder, which adds additional computation and model parameters. Revisiting the points made in Section I, this research exclusively uses the raw occupancy representation of multimodal data to highlight the advantages of employing unprocessed depth information without explicit encoding. The objective is to rival existing multimodal models that rely on explicit point cloud encoding.

III. METHODOLOGY

This section outlines the proposed methodology for integrating foundation models and multimodal techniques into various BEV perception architectures. In Section III-A, the adaptation of DINOv2 and Metric3Dv2 within the LSS architecture replaces the feature and depth extraction to examine the generalizability of foundation models. In Section III-B, LiDAR data is substituted with pseudo-LiDAR in Simple-BEV to enhance performance over the Camera-only model while maintaining its single-modal nature.

Furthermore, considering the substantial performance gains of multimodal models over single-modal setups, the application of multimodal strategies in the single-modal Gaussian Splatting-based semantic occupancy prediction models, GaussianFormer and GaussianOcc, is explored in Sections III-C and III-D. A summary of these modifications and their outcomes is detailed in Table 1.

A. LIFT-SPLAT-SHOOT

One notable bottleneck in the LSS architecture is the EfficientNet-b0 backbone, which is tasked with feature extraction and depth estimation. However, standard for its time, newer and larger networks have since emerged. Notably, the ResNet-101 CNN architecture, which incorporates skip connections, has been shown to outperform preceding models by facilitating the training of much deeper networks [14]. Given these advancements, employing even larger models pretrained on extensive datasets seems prudent. Consequently, the foundation models DINOv2 and Metric3Dv2 will be implemented for feature extraction and depth estimation, respectively. These models are used in their frozen states to demonstrate their versatility and potential to enhance model performance. A key advantage of DINOv2 over EfficientNet-b0 is its finer downsampling rate: EfficientNet downsamples by a factor of 16, while DINOv2 does so by a factor of 14. This allows DINOv2 to retain more detail and improve the precision of feature extraction. However, ResNet-101 still leads with its even finer downsampling factor of 8. For compatibility with the modified model, Metric3Dv2's continuous depth maps need to be converted into depth distributions, a process detailed in the following Section III-A1. Fig. 1 illustrates the updated model architecture.

1) METRIC DEPTH DISTRIBUTION

One key difference between the Metric3Dv2 depth map and LSS's depth representation is that Metric3Dv2 provides a continuous pixel-level depth in meters for each image. At the

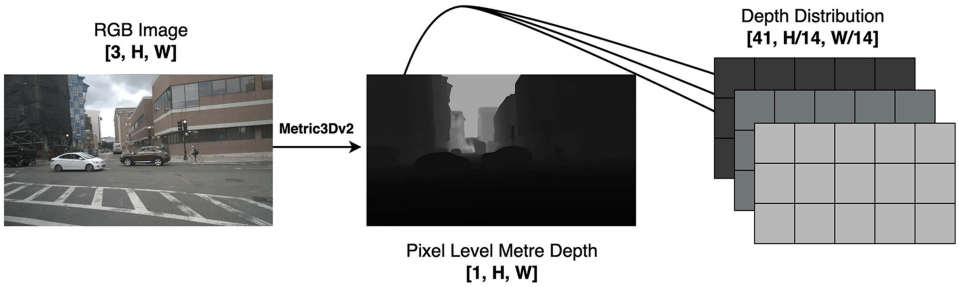

FIGURE 2. Metric3Dv2 Depth Distribution: The RGB image is transformed into a depth map and subsequently into a depth distribution for integration with the LSS model.

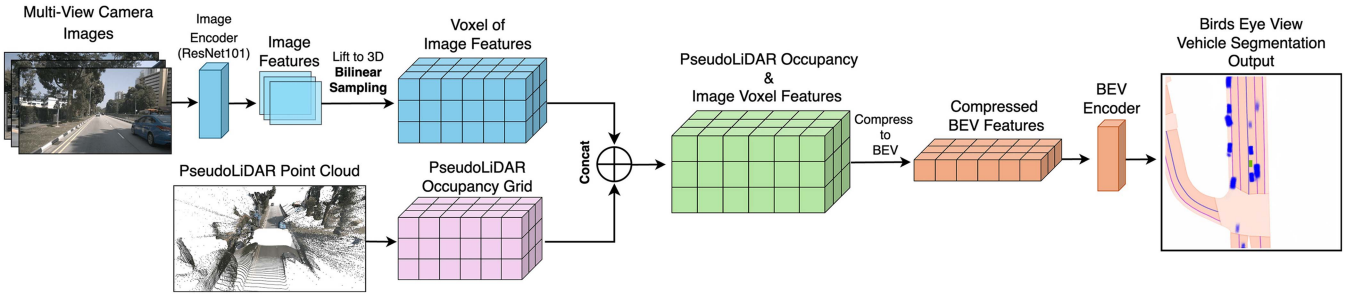

FIGURE 3. Our Modified Camera+Pseudo-LiDAR Simple-BEV Architecture: The Camera-only model does not include any point cloud data, whereas the Camera+LiDAR and Camera+Radar models replace pseudo-LiDAR with LiDAR and Radar, respectively.

same time, LSS utilizes a discrete, probabilistic depth format. In LSS, each 16 × 16 pixel patch of the image is associated with a depth distribution across 41 uniform bins, ranging from 4 to 45 meters, indicating the probability of an image feature being a certain distance from the given camera [6]. To adapt Metric3Dv2's continuous depth map to LSS's discrete format, a convolutional approach is employed: a non-overlapping 16 × 16 pixel patch is moved across the depth map, pooling the depths into the corresponding bins based on each pixel's value within the patch. This process converts the continuous $[1, H, W]$ depth map into a discrete $[41, H/16, W/16]$ depth distribution. The method is illustrated in Fig. 2.

A convolutional layer with batch normalization and ReLU is introduced to enhance model performance and ensure training stability. The resulting tensor accurately reflects the probability that each feature from DINOv2 or EfficientNet is located at specific depths, aligning with the original model's structured approach. For DINOv2, a 14 × 14 pixel patch size is used, which instead downsamples the depth map by a factor of 14.

B. SIMPLE-BEV

Foundation models are not only applicable to revitalizing older architectures but are also useful for enhancing newer ones, as will be evident in this section. A key distinction between LSS and Simple-BEV is that Simple-BEV does not perform depth estimation and instead uses bilinear sampling to splat image features directly to 3D, a method that has proven more effective than depth-based splatting [1]. Motivated by this, Metric3Dv2's depth information is integrated into Simple-BEV while utilizing its effective bilinear sampling technique. This integration involves utilizing camera intrinsic and extrinsic information to project Metric3Dv2's depth information into a pseudo-LiDAR point cloud, which replaces traditional point cloud data in the modified model architecture, as illustrated in Fig. 3. The method for constructing this pseudo-LiDAR point cloud is detailed in Section III-B1.

Regarding the implementation of DINOv2 in Simple-BEV, the experiments with this model have been previously explored and have shown that when fine-tuned with a low-rank approximation, it achieves comparable results to the original model with fewer training iterations [51], [52]. However, the frozen variant of DINOv2 yields unsatisfactory results compared to the original model.

1) CONSTRUCTION OF PSEUDO-LIDAR POINT CLOUD

Pseudo-LiDAR is a point-cloud representation of a scene solely attained from the depth maps generated from camera images. In previous works, this was used to substitute LiDAR for superior results in 3D bounding box estimation [12]. For our purpose, we generate it from depth maps produced by the monocular depth estimation model, Metric3Dv2. While previous studies favored stereo depth estimation algorithms [12] for improved accuracy, recent advancements in foundation models have significantly enhanced monocular depth estimation,

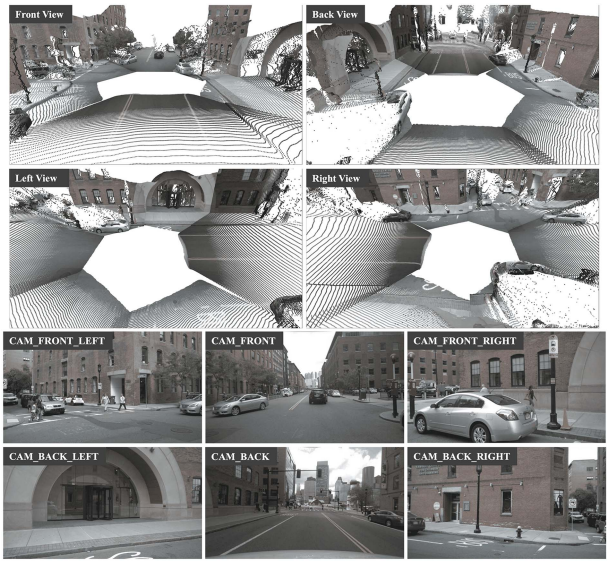

FIGURE 4. Visualisation of our Pseudo-LiDAR Point Cloud: RGB is overlayed purely for visualization purposes.

thus eliminating the need for temporal information or a stereo set-up [8], [11], [47].

Our generation of pseudo-LiDAR follows a process similar to that of previous work. We first generate a depth map with Metric3Dv2, then project it into 3D space using camera intrinsic parameters. Then, we unify these point clouds into a common coordinate system aligned with the voxel grid. Fig. 4 shows a visualization of the resulting point cloud.

Prior studies utilized the KITTI dataset, which provides images at a resolution of 1242 × 375 pixels [23]. KITTI features two side-by-side RGB cameras for stereo vision, requiring only a single depth map per scene. In contrast, the nuScenes dataset equips the ego-vehicle with six non-stereo RGB cameras, offering a 360-degree field of view with images at 1600 × 900 pixels. Consequently, six depth maps must be generated, making smaller depth map sizes of 200 × 112 and 400 × 224 pixels preferable, as larger sizes significantly increase point cloud generation time and data-loading overhead.

The pseudo-LiDAR point cloud is integrated into the following model architectures: Simple-BEV, GaussianFormer, and GaussianOcc. One significant advantage of the pseudo-LiDAR point cloud is its inherent synchronization with the camera, providing alignment that traditional sensors like those in nuScenes' LiDAR or radar may lack. Additionally, the size of the point cloud can be adjusted dynamically by altering the size of the Metric3Dv2 depth map. A comparison of point cloud sizes for various depth map dimensions and LiDAR sweep counts is detailed in Table 4. One sweep of LiDAR corresponds to one full rotational scan of the environment by the LiDAR sensor; this occurs every 50 ms in the nuScenes dataset [5].

One crucial distinction between pseudo-LiDAR and LiDAR lies in the nature of the point clouds, particularly

TABLE 4. Comparison of Point Cloud Size Between Pseudo-LiDAR and LiDAR: The Size of the Pseudo-LiDAR Point Cloud Is Influenced by the Depth Map Size Produced by Metric3Dv2

Metric3Dv2 Depth Map Size	*LiDAR Sweeps*	*Point Cloud Size*
200×112	-	125k
400×224	-	500k
800×400	-	2M
-	1 sweep	25k
-	3 sweeps	75k
-	5 sweeps	125k
-	10 sweeps	250k

regarding acquisition and temporal information. In our implementation, even though a larger depth map significantly increases the size of the point cloud, these additional points originate from the same sample and thus provide limited new information. We suspect that these extra points likely cluster near existing points from the previously smaller depth map. Furthermore, in Simple-BEV, the world is represented discretely with 0.5 m^3 voxels, meaning that larger depth maps do not substantially increase the number of occupied voxels. However, for LiDAR, additional sweeps are collected from different times and different positions in space, hence offering a richer representation of the environment when aggregated.

C. GAUSSIANFORMER

For implementing multi-sensor data in GaussianFormer, the 3D Gaussians *xyz* information is replaced with the point cloud *xyz* information during the initialization phase. Note that it is not possible to use the traditional method of concatenating occupancy due to the lack of a voxel representation of the scene. By doing this, the *xyz* component of the 3D Gaussians is left unlearned, and the remaining properties are to be learned: rotation, scale, opacity, and semantics. It is important to note that despite this, the *xyz* positions will be altered during the training phase; it is just at the initial starting phase that the *xyz* positions remain fixed with no gradient computation. The modified model architecture is illustrated in Fig. 5. Two implementation methods will be delved into in the results Sections IV-D1 and IV-D2, maintaining a consistent underlying concept throughout the exploration:

1) Substitute traditional 3D Gaussian *xyz* positions with data derived from LiDAR, radar, or pseudo-LiDAR.
2) Maintain the originally learned Gaussians while adding new Gaussians that incorporate the point cloud *xyz* data.

To adapt general point cloud data for implementation, it is transformed into the coordinate system of the current frame's LiDAR sensor. Next, points that fall outside the voxel boundaries of [−50,50] in the X and Y dimensions and [−5,3] in the Z dimension are removed as they are considered redundant. Following this, the points are translated into new voxel bounds of [0, 100] for X and Y and [0, 8] for Z. Finally, each dimension is divided by its total span, standardizing the point cloud

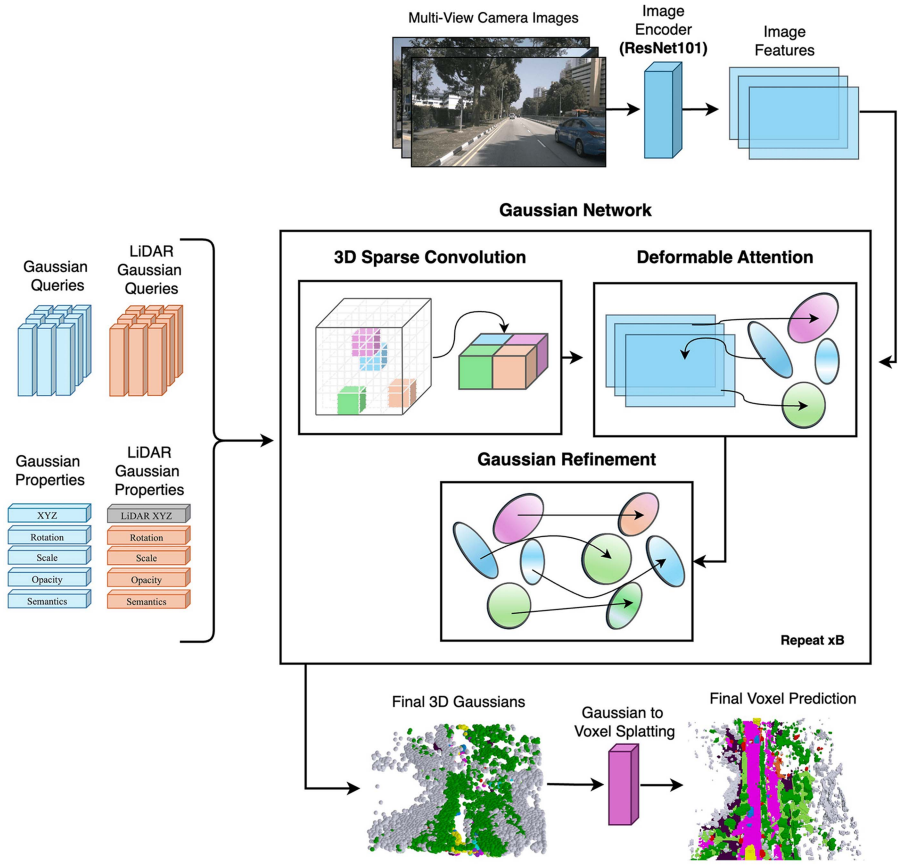

FIGURE 5. Our Modified GaussianFormer Architecture: Illustrated for the supplementation method with an additional set of Gaussians initialized with LiDAR *xyz* information for multimodal deployment. Pseudo-LiDAR and radar would replace the LiDAR data.

to a (1,1,1) grid with the LiDAR coordinate center positioned at [0.5, 0.5, 0.5], completing the process. Next, the generation of the Pseudo LiDAR point cloud for the specific purpose of implementation in GaussianFormer will be discussed.

1) PSEUDO-LIDAR POINT CLOUD

Following the construction process for Simple-BEV in Section III-B1, a different approach is required to construct the pseudo-LiDAR point cloud for GaussianFormer as a fixed number of 3D Gaussians are utilized. An example for 25,600 Gaussians will be provided. It may seem trivial to choose an RGB image size that contains approximately 25,600 pixels across all six camera views, however, this strategy has not been implemented due to some redundant pixels, such as the sky. To fully maximize the pseudo-LiDAR point cloud, an RGB image size of 90 $\times$ 55 pixels is used, which maintains close to the original aspect ratio while providing greater than 25,600 points, permitting fine-tuning of the point cloud. The initial step in processing this data involves eliminating points outside predefined voxel boundaries. Following this, downsampling is necessary to adjust the point total to less than or equal to 25,600. This downsampling is executed by identifying and selectively removing points within a 0.1m radius of five or more other points. Should the count exceed 25,600 following this reduction, the downsampling process is continued by incrementally increasing the search radius until the target number is achieved. This method ensures the maintenance of the structural integrity of the point cloud.

D. GAUSSIANOCC

Multi-sensor fusion in GaussianOcc must use a different approach to GaussianFormer because of the lack of usage of 3D Gaussians for occupancy prediction and solely for loss calculation. Here, we take a more traditional approach, as seen in [1]. Taking LiDAR as an example, it is voxelized into a 3D binary occupancy grid and then concatenated with the voxel of image features, adding one feature to the 64 image features. LiDAR, radar, and pseudo-LiDAR all add one additional feature, whereas radar, coupled with its metadata, will add 16 features. Passing this to the 3D convolutional network, the end result is a voxel of semantics refined with rich point cloud information derived from LiDAR, pseudo-LiDAR, or radar data. For the pseudo-LiDAR point cloud, the construction method is the same as that for Simple-BEV as discussed in Section III-B1. The modified architecture is illustrated in Fig. 6.

IV. RESULTS

This section starts with an overview of the datasets and configurations outlined in Section IV-A. The subsequent four

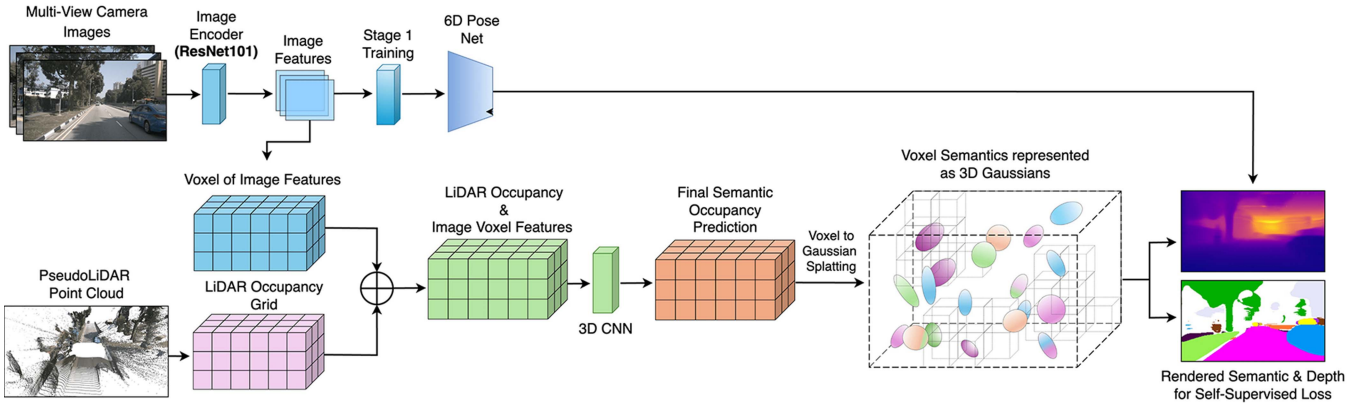

FIGURE 6. Our Modified GaussianOcc Architecture: Utilizing point cloud information from pseudo-LiDAR for multimodal deployment. Pseudo-LiDAR can be replaced with LiDAR, radar, or radar enhanced with metadata.

sections present the results from training and evaluating the modified models described in Section III. A detailed discussion of these findings is provided in Section IV-F.

A. DATASET AND CONFIGURATION

All four models are trained following the configurations detailed in their respective papers. However, there are some discrepancies in the datasets used. Lift-Splat-Shoot and Simple-BEV are trained on the complete nuScenes training dataset, which includes 700 scenes. Additionally, we train Lift-Splat-Shoot on half of the dataset (350 scenes) to demonstrate the generalizability of foundation models. Inference tests are conducted on 150 validation scenes, with performance evaluated solely on vehicle segmentation using the IoU metric. The nuScenes SDK provides ground truth labels for these models to use.

For GaussianFormer, training also utilizes the full 700 scenes from nuScenes, but the ground truth labels are sourced from the SurroundOcc dataset [26], as they are not available in the nuScenes SDK. Conversely, GaussianOcc is trained on 600 scenes as specified by the Occ3D dataset [29], with ground truth labels used only for evaluation purposes due to self-supervision.

Hyperparameter comparisons between the original and modified models are detailed in Table 5. The Lift-Splat-Shoot model, with the Giant variants of the foundation models, sees an increase of 1127 M additional parameters, primarily non-trainable, with some exceptions in the BEV encoder due to an increased patch count. In contrast, Simple-BEV shows no change in parameters, as it merely substitutes point cloud data. GaussianFormer experiences a 1.9 M increase in parameters when incorporating three LiDAR sweeps, attributed to additional parameters in the newly introduced Gaussians. Lastly, GaussianOcc, with the addition of three LiDAR sweeps, requires only 864 extra parameters in the 3D CNN due to the inclusion of an additional feature dimension.

In terms of computational resources, Lift-Splat-Shoot is trained on a single NVIDIA A100-SXM4-40 GB GPU. Meanwhile, Simple-BEV, GaussianFormer, and GaussianOcc each utilize four NVIDIA A100-SXM4-40 GB GPUs for training.

TABLE 5. Total Parameters for Original Models vs. Our Modified Models: Lift-Splat-Shoot + Ours Refers to the Implementation Incorporating the Giant Variants of DINOv2 and Metric3Dv2. The GaussianFormer and GaussianOcc C+L Models Utilize Three LiDAR Sweeps

Model	*Modality*	*Model Parameters*
Lift-Splat-Shoot	C	14.3M
Lift-Splat-Shoot + Ours	C	1141.2M
Simple-BEV	C+L	42.1M
Simple-BEV + Ours	C+PL	42.1M
GaussianFormer	C	53.5M
GaussianFormer + Ours	C+L	55.4M
GaussianOcc	C	64.7M
GaussianOcc + Ours	C+L	64.7M

B. LIFT-SPLAT-SHOOT

This section explores the integration of foundation models DINOv2 and Metric3Dv2 into the Lift-Splat-Shoot architecture. The investigation begins with an examination of the impact of foundation model size on performance in Section IV-B1, followed by a direct comparison to the baseline model in Section IV-B2. Finally, a qualitative analysis of the BEV vehicle segmentation output is conducted in Section IV-B3.

1) LARGENESS OF FOUNDATION MODELS

For CNN backbones such as EfficientNet and ResNet, there is a trade-off between the depth of the model and the increase in trainable parameters, which can enhance performance. This trade-off also applies to the foundation models, such as DINOv2, where an increase in model size corresponds to more model parameters and larger patch embedding sizes, as detailed in Table 2. Various sizes of DINOv2 and Metric3Dv2 are investigated, analyzing how changes in model size affect overall performance in LSS.

Initially, both DINOv2 and Metric3Dv2 were implemented. Referring to Table 6, it becomes apparent that an increase in foundation model size correlates with enhanced performance

TABLE 6. Effect of the Largeness of DINOv2 and Metric3Dv2 on LSS Performance: Models Were Trained on the Full Dataset for 250,000 Iterations. The Best-Performing Model in Each Section Is Highlighted in Bold

Configuration		Peak	
DINOv2 Size	*Metric3Dv2 Size*	*IoU*	*Iterations*
Giant	Small	36.1	50k
Giant	Large	39.6	80k
Giant	Giant	**41.9**	55k
Small	Giant	40.1	120k
Base	Giant	41.0	145k
Large	Giant	41.1	125k
Giant	Giant	**41.9**	55k

TABLE 7. Effect of the Largeness of Just Metric3Dv2 on LSS Performance: Models Were Trained on the Full Dataset for 250,000 Iterations. The Best-Performing Model is Highlighted in Bold

Configuration		Peak	
Feature Extractor	*Metric3Dv2 Size*	*IoU*	*Iterations*
EfficientNet	Small	34.3	315k
EfficientNet	Large	38.6	150k
EfficientNet	Giant	**40.5**	110k

in LSS. Specifically, Metric3Dv2 shows a significant improvement, jumping from an IoU of 36.1 for the Small model to 41.9 for the Giant model—an increase of 5.8 IoU. DINOv2 exhibits a similar trend, though the increase is more modest at a 1.8 IoU increase from Small to Giant. This smaller increase may stem from the significant reduction in feature dimensionality before the features are input into the BEV encoder, as outlined in Section II-C. The Giant model, in particular, experiences a substantial information loss when its feature count is compressed from 1584 to just 64. While necessary for compatibility, this downsampling process likely compromises model performance by diminishing the quality of features.

Significant emphasis is not placed on the iterations required to achieve peak performance, as all variations tend to reach within 0.5 IoU of their peak IoU after approximately 50,000 iterations. Overall, the Giant variations of these models deliver the best performance and do so more rapidly than the original model.

In this second experiment, only Metric3Dv2 is implemented for depth estimation, with the EfficientNet-b0 backbone handling feature extraction. Similar performance improvements to the first experiment are observed, though the iteration count was slightly higher due to using the poorer-performing EfficientNet-b0. The increase in performance from the Small to Giant Metric3Dv2 variant was significant, showing a gain of 6.2 IoU, as noted in Table 7. Additionally, when comparing the Giant variants, the combination of DINOv2+Metric3Dv2 achieved an IoU of 41.9, whereas the pairing of EfficientNet+Metric3Dv2 scored slightly lower at 40.5 IoU. Two factors could explain this marginal difference: firstly, the potential inadequacy of the depth estimation

TABLE 8. Original LSS Model vs. Our Modified Models: Baseline Model Results Are Highlighted in Grey. Iters Indicates the Number of Iterations Required to Achieve Peak IoU Performance. The Best-Performing Model in Each Section Is Highlighted in Bold

Configuration				Peak	
Feature Extractor	*Depth Extractor*	*Model Size*	*Training Data*	*IoU* ↑	*Iters*
EfficientNet	EfficientNet	N/A	Full	33.0	300k
DINOv2	Metric3Dv2	Giant	Full	**41.9**	55k
DINOv2	Metric3Dv2	Small	Full	34.0	115k
EfficientNet	Metric3Dv2	Giant	Full	40.5	110k
EfficientNet	Metric3Dv2	Small	Full	34.1	115k
EfficientNet	EfficientNet	N/A	Half	29.1	150k
DINOv2	Metric3Dv2	Giant	Half	**40.4**	35k
DINOv2	Metric3Dv2	Small	Half	32.1	30k
EfficientNet	Metric3Dv2	Giant	Half	37.4	55k
EfficientNet	Metric3Dv2	Small	Half	31.4	80k

method in LSS, where a single backbone is tasked with predicting both image features and depth, and secondly, the previously mentioned loss of information due to the downsampling of DINOv2 embeddings which does not permit it to reach maximum performance.

Another omitted experiment is implementing DINOv2 for feature extraction and leaving EfficientNet for depth estimation. It is omitted due to the discrepancy in downsampling for both models. For compatibility between these models, different-sized images would be required for DINOv2 and EfficientNet to achieve the same number of image patches, leading to an unfair experiment.

2) COMPARISON WITH THE ORIGINAL MODEL

The final experiment will compare the implementation of foundation models (DINOv2+Metric3Dv2 and EfficientNet+Metric3Dv2) to the original model (EfficientNet+EfficientNet) regarding performance metrics. The LSS models with foundation models are trained with the Giant and Small variants. All models are trained twice, once on the full dataset and once on half of the dataset.

First, all models are trained on the full dataset. According to Table 8, the Giant models significantly outperform the Small variants, aligning with earlier observations. However, the Small foundation models may be more suitable in specific scenarios where reduced training time is crucial. Additionally, all variations of the modified model architecture significantly outpace the original model. Notably, the best-performing setup, the Giant DINOv2+Metric3Dv2, exceeds the original model by a remarkable 8.9 IoU, demonstrating the effectiveness of foundation models. An intriguing find is that the Small EfficientNet+Metric3Dv2 slightly outperforms the Small DINOv2+Metric3Dv2 by 0.1 IoU, likely due to

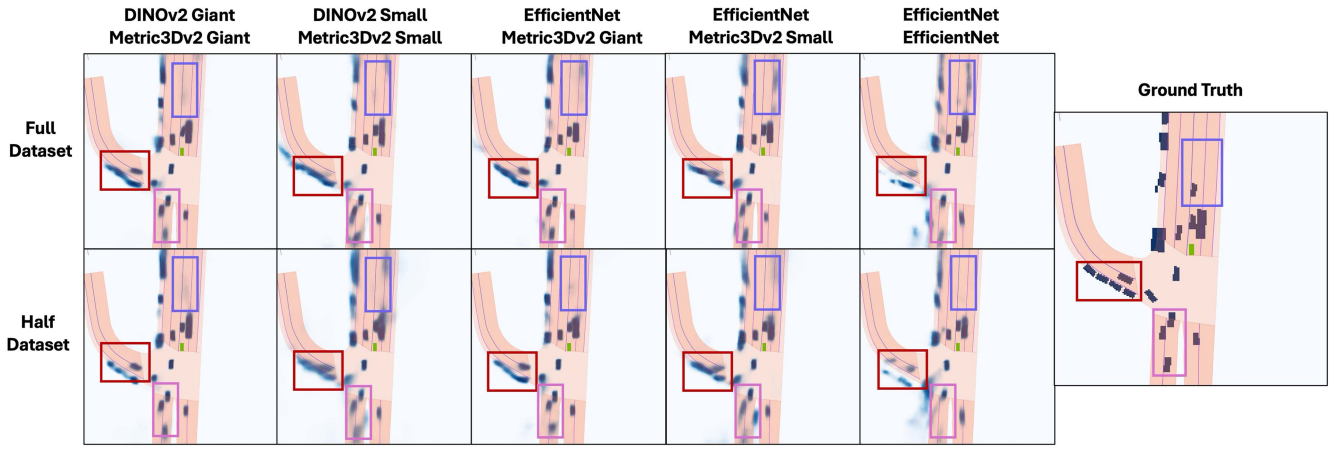

FIGURE 7. Lift-Splat-Shoot Visual Comparison: Visual results comparing models trained on the full dataset and half dataset across the various architectures listed in Table 8.

the limitations of the frozen Small DINOv2 model in comparison to the fine-tuned EfficientNet-b0, especially given its smaller parameter count compared to the Giant variation. The convergence rate of these models is notably rapid, with the Giant variants surpassing the original model within just 5,000 iterations—a dramatic reduction compared to the original model's 300,000 iterations needed for peak performance.

Subsequently, all models are retrained with only half of the training data. Predictably, performance declines across the board. The original model experiences a 3.9 drop in IoU, highlighting its poor generalization with insufficient data. However, the impact on the foundation models is less pronounced, with only a 1.5 IoU reduction in the top-performing modified model, thanks to their extensive pretraining. The Giant variants of both DINOv2+Metric3Dv2 and EfficientNet+Metric3Dv2 even surpass the original model trained on the full dataset, further underscoring their robustness. Conversely, the Small variants struggle without a diverse training dataset and fail to outperform the original model trained with full data.

Overall, the use of DINOv2 and Metric3Dv2 in LSS, particularly the Giant variants, significantly enhances performance over the original setup, even with limited training data.

3) QUALITATIVE ANALYSIS

Although the results from the previous section indicate that the modified model significantly outperforms the baseline, it remains crucial to analyze the visual BEV outputs to gain a comprehensive understanding of the model's behavior, especially where it fails to detect vehicles that the original model successfully identifies.

Upon examining Fig. 7, a clear distinction emerges between the models trained with the full dataset and those with half the data. Models trained with the full dataset deliver more accurate and confident predictions. In contrast, the half dataset models exhibit a 'ghosting' effect around vehicles, characterized by less precise contours due to the model's uncertainty. This issue is primarily due to insufficient training data diversity and volume, most noticeably affecting vehicles that are distant from the ego-vehicle or partially obscured, as are those highlighted in red. This ghosting effect persists even with the implementation of foundation models, underscoring that while pretraining offers substantial benefits, extensive and diverse training data is still essential for optimal generalization.

Focusing on the models trained with the full dataset, there is a discernible improvement in vehicle segmentation quality when progressing leftward in the figure, attributed to the integration of foundation models into the architecture. Specifically, the vehicles highlighted in red, which pose challenges for all models, show improved segmentation with DINOv2 and Metric3Dv2, likely due to enhanced depth information provided by these models. In the blue-highlighted area, the original model predicts non-existent vehicles while failing to detect an actual vehicle present. Although all models struggle with accurate segmentation due to occlusion, the foundation models exhibit minimal ghosting. They are more cautious in their predictions, avoiding the over-prediction seen with the original model.

C. SIMPLE-BEV

This section investigates the implementation of pseudo-LiDAR in the Simple-BEV model architecture to replace traditional LiDAR and radar data, as detailed in Section IV-C1. A qualitative analysis follows this in Section IV-C2.

1) MULTIMODAL FUSION: CAMERA+PSEUDO-LIDAR

As outlined in Section III-B, Metric3Dv2's depth maps are implemented into the Simple-BEV model in the form of a pseudo-LiDAR point cloud to maintain the bilinear sampling splatting method for increased performance. To assess the impact of point cloud size on performance, two depth map sizes are employed: 200×112 and 400×224 pixels. The results summarized in Table 9 demonstrate that the Camera+Pseudo-LiDAR models surpass the baseline Camera-only model in

TABLE 9. Baseline Simple-BEV Models vs. Camera+Pseudo-LiDAR Model: All Models Were Trained for 25,000 Iterations With an Input Image Resolution of 800 × 448 to the ResNet Backbone. The Best-Performing Model Is Highlighted in Bold

Configuration		Peak
Modality	*Depth Map Size*	*IoU*
Camera	N/A	47.4
Camera+Radar	N/A	55.7
Camera+LiDAR	N/A	**60.8**
Camera+Pseudo-LiDAR (ours)	200×112	50.3
Camera+Pseudo-LiDAR (ours)	400×224	50.7

TABLE 10. Baseline Simple-BEV vs. Camera+Pseudo-LiDAR Model: All Models Were Trained Using a Reduced Input Image Resolution of 400 × 224 pixels to the ResNet Backbone. The Best-Performing Model Is Highlighted in Bold

Configuration		Peak
Modality	*Depth Map Size*	*IoU*
Camera	N/A	42.3
Camera+Pseudo-LiDAR (ours)	200×112	**48.6**

performance. With a depth map size of 200 × 112 pixels, an improvement of 2.9 IoU is noted, and for a larger depth map size of 400 × 224 pixels, the improvement increases to 3.3 IoU. The marginal improvement observed with the larger depth map size compared to the smaller one suggests that the smaller depth map provides adequate information while offering the advantages of reduced training time and lower computational demands. This highlights that, despite the extensive training of the Camera-only model, incorporating additional data through this novel method can further enhance the model's performance.

An important observation is that despite the pseudo-LiDAR point cloud having five times the density of one LiDAR sweep, as evident in Table 4, it fails to outperform the Camera+LiDAR model, lagging by approximately 10 IoU. This discrepancy can be attributed to the modalities present in the model: the Camera+Pseudo-LiDAR model is inherently single-modal, relying solely on data from the camera, while the Camera+LiDAR model is multimodal, incorporating data from both camera and LiDAR sensors. The LiDAR sensor captures different information from different perspectives compared to camera sensors. In contrast, the pseudo-LiDAR point cloud replicates the information captured by the RGB camera in a depth representation.

With adequate training, the base Camera-only model can effectively extract the valuable information that the Metric3Dv2 depth maps would have provided. This is evidenced when the ResNet-101 backbone is trained with an image size of 400 × 224 pixels, half the size used in the original model's training. At this reduced image resolution, a significant improvement of 6.3 IoU is observed when comparing the Camera+Pseudo-LiDAR to the Camera-only model, as shown in Table 10. This is considerably higher than the +2.9 IoU improvement when comparing the Camera-only versus the Camera+Pseudo-LiDAR model when trained with the full image resolution of 800 × 448 pixels in Table 9. This suggests that the pseudo-LiDAR point cloud provides essential information that the model could not otherwise discern from compressed image features alone.

The pretrained nature of Metric3Dv2 enables it to provide an early boost in performance for the Simple-BEV model. This advantage is highlighted by examining the performance increments from iteration 2,500 to 25,000 for both the Camera-only and Camera+Pseudo-LiDAR models in Table 10. For the Camera-only model, a 28.8% improvement is observed during this period. In contrast, the Camera+Pseudo-LiDAR model shows a more modest increase of 20.3%, despite ultimately achieving a higher final IoU than the Camera-only setup. This indicates that Metric3Dv2 significantly enhances performance early in the training process.

In an additional experiment, the pseudo-LiDAR point cloud was coupled with RGB metadata. Contrary to expectations, this modification led to a decrease in performance.

2) QUALITATIVE ANALYSIS

Similarly to the analysis of the LSS BEV output, the BEV output for Simple-BEV will be examined, which is displayed in Fig. 8. It's immediately apparent that integrating additional sensors, such as LiDAR and radar, significantly enhances segmentation capabilities, confirming observations from the original study [1]. Interestingly, the vehicle highlighted in red, which the Camera-only model fails to detect, possibly due to obstruction from the camera's view, is successfully segmented by the inclusion of depth information via LiDAR, radar, and, notably, pseudo-LiDAR—even though pseudo-LiDAR utilizes only transformed camera data. Additionally, the vehicle highlighted in green is missing from both the Camera-only and Camera+Pseudo-LiDAR models. In contrast, it is accurately detected by models that incorporate LiDAR and radar data due to their longer range, different sensor positioning, and accurate depth.

An intriguing observation emerges with the cluster of three vehicles highlighted in pink. The Camera-only model identifies two vehicles in this space, whereas the other models detect only one. This outcome is unexpected since one might assume that any correct segmentation by the Camera-only model would be replicated in models also utilizing camera data. The discrepancy could be attributed to the possibility that LiDAR, radar, and pseudo-LiDAR did not accurately detect these vehicles, potentially leading the model to believe it was a false positive from the camera features.

Overall, while the pseudo-LiDAR point cloud does improve vehicle segmentation relative to the Camera-only model, its performance still does not match that of the multimodal models that utilize both LiDAR and radar, underscoring the

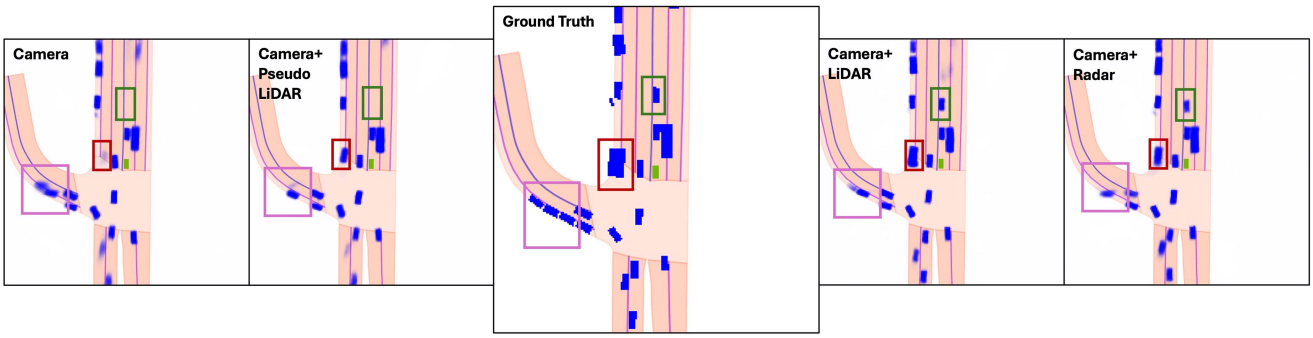

FIGURE 8. Simple-BEV Visual Comparison: Visual results for Camera-only, Camera+Radar, Camera+LiDAR, and Camera+Pseudo-LiDAR corresponding to the models in Table 9.

TABLE 11. GaussianFormer With Multi-Sensor Fusion: The Initial Gaussian *xyz* Points Are Entirely Replaced With Those Derived From Point Cloud Data. The Best-Performing Model Is Highlighted in Bold

Model	*IoU*	*mIoU*
Original	30.1	19.3
w/ LiDAR	**37.0**	**24.0**
w/ Radar	29.9	19.1
w/ Pseudo-LiDAR	27.5	17.6

limitations of relying solely on single-modal models, despite the deployment of a foundation model for aid.

D. GAUSSIANFORMER

This section explores the results of multimodal deployment of LiDAR, radar, and pseudo-LiDAR in the GaussianFormer model. The exploration begins with Section IV-D1, detailing the method of replacement of the Gaussians. It is followed by Section IV-D2, which elaborates on the method of supplementation of additional Gaussians. Section IV-D3 examines the impact of multiple sweeps on model performance. Section IV-D4 presents a comparison of these models against current SOTA models. Finally, Section IV-D5 provides a qualitative analysis.

1) MULTIMODAL FUSION: REPLACING XYZ DATA

Beginning the multimodal fusion analysis, the original Gaussian points in the GaussianFormer model architecture are replaced with point cloud data from LiDAR, radar, or pseudo-LiDAR. Performance is assessed using two metrics: mIoU, which measures semantic IoU and penalizes the model for the incorrect classification of semantics, and IoU, which focuses solely on occupancy without considering semantic classes. More focus is placed on mIoU as the misclassification of voxels is very important; for example, classifying a pedestrian as a static object should be penalized.

As shown in Table 11, LiDAR notably enhances performance over the original model, particularly in IoU, with an increase of 6.9 points, and mIoU also sees a significant boost of 4.7 points. This improvement is mainly due to the LiDAR sensor's long-range capabilities and high density. For the radar data, given its sparsity with the average scene containing about 2,000 points with five sweeps, every Gaussian point is not replaced with radar data. Instead, 22,600 points are retained as learnable, and only 3,000 points are assigned radar *xyz* information, which results in a slight decrease in mIoU of 0.2 points. This decline could be partly attributed to how the data is implemented; for instance, if a scene includes 500 radar points, approximately 2,500 will be dummy points, contributing minimally to scene representation. A more dynamic approach to managing radar's sparsity could potentially enhance performance.

For pseudo-LiDAR, the outcome is a drop of 1.7 mIoU with respect to the original model, which is lackluster considering how this data is relatively high quality, providing accurate depth information. Although it matches LiDAR in terms of density, pseudo-LiDAR does not even match the performance of the original model. This is likely because the original model's ability to learn Gaussian positions is more valuable than the data derived from pseudo-LiDAR. The performance of all multimodal methods is improved in the following Section IV-D2 with a more robust multi-sensor fusion method.

2) MULTIMODAL FUSION: SUPPLEMENTING XYZ DATA

In the previous experiment, for both LiDAR and pseudo-LiDAR, all learned Gaussian *xyz* information was replaced with the corresponding point cloud data; however, more commonly, in multimodal fusion, numerous data types are integrated in a complementary fashion. Following this, the original set of learned Gaussians is retained, and an additional set of Gaussians is supplemented, which contains the point clouds *xyz* information. The original model is retrained with 51,200 learned Gaussians for a fair comparison. However, it still underperforms relative to the original, suggesting the latter is already near peak performance. When comparing our models against the original model, we will refer to the one trained with 25,600 Gaussians. For simplicity, moving forward, we will abbreviate the number of Gaussians, referring to 25,600 as 25 k.

TABLE 12. GaussianFormer With Additional Multi-Sensor Gaussians: The Original Gaussians (Learned Gaussians) Are Supplemented With an Additional Set Derived From Point Cloud Data (Fixed Gaussians). The Best-Performing Model Is Highlighted in Bold

Configuration			Performance	
Model	*Learned Gaussians*	*Fixed Gaussians*	*IoU*	*mIoU*
Original	25k	0k	30.1	19.3
Original	51k	0k	30.0	19.1
w/ LiDAR	25k	25k	**43.2**	**28.5**
w/ Radar	25k	3k	32.4	21.0
w/ Pseudo-LiDAR	25k	25k	30.7	20.4

TABLE 13. Effect of Additional LiDAR Sweeps and Pseudo-LiDAR Point Cloud Size: Supplementation With Additional Gaussians Is Maintained Across Experiments. The Best-Performing Model in Each Section Is Highlighted in Bold

Configuration			Performance	
Model	*Learned Gaussians*	*Fixed Gaussians*	*IoU*	*mIoU*
Original	25k	0k	30.1	19.3
Original	51k	0k	30.0	19.1
w/ LiDAR (1 sweep)	25k	25k	**43.2**	**28.5**
w/ Pseudo-LiDAR	25k	25k	30.7	20.4
Original	102k	0k	28.1	16.7
w/ LiDAR (3 sweeps)	25k	75k	**43.5**	**28.7**
w/ Pseudo-LiDAR	25k	75k	30.6	20.0

Examining Table 12, integrating additional Gaussians with LiDAR data provides a huge boost in performance for both mIoU and IoU, significantly outperforming the baseline model by 9.2 mIoU. For radar, a notable boost in performance of 1.7 in mIoU is seen over the baseline despite it containing only 3 k additional Gaussians. Pseudo-LiDAR finally outperforms the original model, benefiting greatly from the learned Gaussians, outperforming the original model by 1.1 mIoU. Overall, it is clear that coupling data is imperative in maximizing performance.

3) ABLATION: MULTIPLE SWEEPS

The effect multiple LiDAR sweeps and pseudo-LiDAR point cloud size have on performance is investigated in this section, and the original model is retrained with an equivalent number of Gaussians (102 k) for fairness. However, again, additional Gaussians hinder the model, so when referring to the baseline, the 25 k model is being referred to. Examining Table 13, additional LiDAR sweeps increase model performance, although only slightly, with an increase in mIoU of 0.2 when utilizing 3 sweeps over 1 sweep, outperforming the baseline 25 k model by 9.4 mIoU. Other fusion models, such as Simple-BEV, benefit more greatly with additional LiDAR sweeps than ours [1]. Note that current SOTA models utilize up to ten LiDAR sweeps [17], [32].

For the pseudo-LiDAR implementation, we use 75 k Gaussians for the larger model. A 0.4 mIoU drop in performance is observed, likely due to the additional points coming from the same sample, unlike LiDAR sweeps. Consequently, there are additional Gaussians that contribute little to the scene but still need to be optimized, hence overloading the model. Perhaps additional points from a previous sample would improve performance due to increased diversity. Note that the radar has already been trained with five sweeps due to its sparsity. Further increases in sweeps are omitted due to insufficient computation power, with technicalities explained in Section IV-F.

4) COMPARISON WITH STATE-OF-THE-ART

Following the previous experiments, our best-performing Camera+LiDAR (C+L), Camera+Radar (C+R), and Camera+Pseudo-LiDAR (C+PL) models are taken and compared to the SOTA semantic occupancy prediction models evaluated on the SurroundOcc nuScenes dataset, as evident in Table 14.

Examining Table 14, DAOcc greatly outperforms all other models in all categories except three. OccFusion's Camera+LiDAR+Radar (C+L+R) narrowly outperforms DAOcc for bicycle segmentation. Co-Occ takes supreme for driveable surfaces by quite a margin of 1.3 mIoU and ties DAOcc for bus segmentation. However, Co-Occ fails to generalize well and is vastly outperformed on all other labels by DAOcc and C+L GaussianFormer, with the latter using just three LiDAR sweeps.

Firstly, looking at the camera-only models, GaussianFormer does not outperform the camera-only SurroundOcc model, with the integration of point-cloud data significantly enhancing its capabilities and positioning it alongside the SOTA multimodal models. In comparisons between the Camera+Radar models, the C+R GaussianFormer model and OccFusion demonstrate only minimal improvements over baseline Camera-only setups, indicating that radar may not be optimal for semantic occupancy prediction. Although OccFusion surpasses in occupancy IoU by 1.62, the C+R GaussianFormer model achieves a slightly higher mIoU by 0.27.

Regarding Camera+LiDAR comparisons between our model with both OccFusion and DAOcc, the results are mixed. The C+L GaussianFormer model falls behind OccFusion by 0.81 points in occupancy IoU but exceeds it by 1.84 points in semantic mIoU, underscoring the effectiveness of allowing LiDAR Gaussians to learn semantic information. However, DAOcc significantly outperforms the C+L GaussianFormer model in IoU and mIoU by 1.46 and 1.79 points, respectively. This proves that a robust, simple method such as DAOcc can yield excellent results, as seen in models like Simple-BEV for BEV vehicle segmentation.

TABLE 14. Comparison of Our Multimodal GaussianFormer Models With State-of-the-Art Models. All Methods Are Trained With Dense Occupancy Labels From SurroundOcc [26]. Modality: Camera (C), Lidar (L), Radar (R), Pseudo-LiDAR (PL). Note That InverseMatrixVT3D and Camera-Only OccFusion Are the Same Models And, Hence, Have the Same Performance. Models are Ordered by the Nature of Modality (Single-Modal, C+R, C+L) and Then by Decreasing Performance With Regard to the mIoU Metric. C+L Models With a ∗ Use Ten LiDAR Sweeps, While Those With a † Use Three LiDAR Sweeps. The Best Performer in Each Performance Category Is Highlighted in Bold

Method	Input Modality	IoU	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. surf.	other flat	sidewalk	terrain	manmade	vegetation
MonoScene [30]	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
LMSCNet [35]	L	36.60	14.90	13.10	4.50	14.70	22.10	12.60	4.20	7.20	7.10	12.20	11.50	26.30	14.30	21.10	15.20	18.50	34.20
Atlas [36]	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
BEVFormer [21]	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	22.88	22.17	13.80	22.21
TPVFormer [31]	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
InverseMatrixVT3D [33]	C	31.85	18.88	18.39	12.46	26.30	29.11	11.00	15.74	14.78	11.38	13.31	21.61	36.30	19.97	21.26	20.43	11.49	18.47
OccFusion [32]	C	31.85	18.88	18.39	12.46	26.30	29.11	11.00	15.74	14.78	11.38	13.31	21.61	36.30	19.97	21.26	20.43	11.49	18.47
RenderOcc [34]	C	29.20	19.00	19.70	11.20	28.10	28.20	9.80	14.70	11.80	11.90	13.10	20.10	33.20	21.30	22.60	22.30	15.30	20.90
GaussianFormer [15]	C	30.05	19.31	19.78	11.68	25.65	29.53	10.16	15.83	13.10	8.67	12.83	20.48	39.61	22.04	24.79	23.23	11.45	20.16
SurroundOcc [26]	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
OccFusion* [32]	C+L+R	44.66	27.30	27.09	**19.56**	33.68	36.23	21.66	24.84	25.29	16.33	21.81	30.01	39.53	19.94	24.94	26.45	28.93	40.41
OccFusion [32]	C+R	33.97	20.73	20.46	13.98	27.99	31.52	13.68	18.45	15.79	13.05	13.94	23.84	37.85	19.60	22.41	21.20	16.16	21.81
M-CONet* [50]	C+L	39.20	24.70	24.80	13.00	31.60	34.80	14.60	18.00	20.00	14.70	20.00	26.60	39.20	22.80	26.10	26.00	26.00	37.10
OccFusion* [32]	C+L	44.35	26.87	26.67	18.38	32.97	35.81	19.39	22.17	24.48	17.77	21.46	29.67	39.01	21.94	24.90	26.76	28.53	40.03
Co-Occ* [39]	C+L	41.10	27.10	28.10	16.10	**34.00**	37.20	17.00	21.60	20.80	15.90	21.90	28.70	**42.30**	25.40	29.10	28.60	28.20	38.00
DAOcc* [17]	C+L	**45.00**	**30.50**	**30.80**	19.50	**34.00**	**38.80**	**25.00**	**27.70**	**29.90**	**22.50**	**23.20**	**31.60**	41.00	**25.90**	**29.40**	**29.90**	**35.20**	**43.50**
GaussianFormer [15] + Ours	C+R	32.35	21.00	20.02	14.38	27.60	31.27	11.92	18.08	14.99	10.18	14.07	23.79	39.54	22.79	25.11	22.75	15.90	23.58
GaussianFormer† [15] + Ours	C+L	43.54	28.71	29.12	19.45	33.92	38.21	19.84	25.57	27.60	18.91	21.99	30.86	40.92	24.10	27.49	27.48	32.50	41.41
GaussianFormer [15] + Ours	C+PL	30.65	20.44	19.54	14.37	28.08	31.74	11.67	17.17	15.09	10.16	13.50	23.52	39.03	21.60	24.82	22.52	12.41	21.86

5) QUALITATIVE ANALYSIS

For qualitative analysis, visualizations compare our top-performing multimodal models—Camera+LiDAR, Camera+Radar, and Camera+Pseudo-LiDAR GaussianFormer models—with the baseline model and ground truth. This examination will help us delineate the enhancements and limitations inherent to each approach in real-world scenarios.

As evident in Fig. 9, all multimodal models appear to achieve more accurate segmentation than the baseline Camera-only model in scene segmentation, with particularly notable improvements in road segmentation from the C+L and C+R models. The C+L model excels in identifying more occupied voxels due to its dense point cloud, which, in combination with camera data, allows for precise semantic interpretation of these voxels. These models benefit from additional sensor data, capturing more details than the Camera-only and C+PL models. While pseudo-LiDAR is less effective than LiDAR or radar, it still enhances vehicle segmentation, especially for vehicles further from the ego-vehicle, like those at the bottom of the voxel grid.

No model or ground truth labeling is perfect, and some misclassifications are inevitable. We observe the misclassification of empty space as occupied across all models, highlighted with a green box, and the misclassification of a truck as a construction vehicle, highlighted with a black rectangle in Fig. 9. For the bus, the misclassification may be due to the area in the image being obscured, leading the models to possibly perceive the top of another vehicle as a bus, car, or truck in the distance. This error can solely be attributed to the models. However, the misclassification of the truck is different. Reviewing the CAM_BACK_LEFT image reveals that the vehicle is a cement mixer, which one may classify as a construction vehicle, yet it is labeled as a truck. This ambiguity is likely due to the manual labeling of the LiDAR point clouds used to generate the ground truth data. For the C+PL model, the classification as both a truck and a construction vehicle highlights the ambiguity between these two classes.

E. GAUSSIANOCC

This section investigates the integration of LiDAR, radar, radar+metadata, and pseudo-LiDAR in the GaussianOcc model. It begins with Section IV-E1, which details the multimodal deployment. Section IV-E2 explores the impact of multiple sensor sweeps on performance. Section IV-E3 compares these models against current self-supervised SOTA models. Finally, Section IV-E4 offers a qualitative analysis.

1) MULTIMODAL FUSION: LIDAR, RADAR, AND PSEUDO-LIDAR

For multimodal analysis in the GaussianOcc architecture, a more traditional approach is taken, as opposed to that in GaussianFormer. The image feature voxel grid is supplemented with the occupancy grid constructed from the point cloud data, previously illustrated in Fig. 6. For evaluation, the original paper's two metrics are used: mIoU, which is semantic IoU including all classes, and mIoU*, which is semantic IoU ignoring the classes 'other' and 'other flat', we will place more emphasis on mIoU* as this is more commonly used. Upon reviewing Table 15, it is evident that the results diverge from those in previous sections, with LiDAR and pseudo-LiDAR yielding only marginal performance gains of less than 0.5 mIoU*, and radar even slightly reducing performance. While these findings cannot necessarily be definitively explained, they could be attributed to two factors: the absence of explicit encoding and the self-supervised nature of the model.

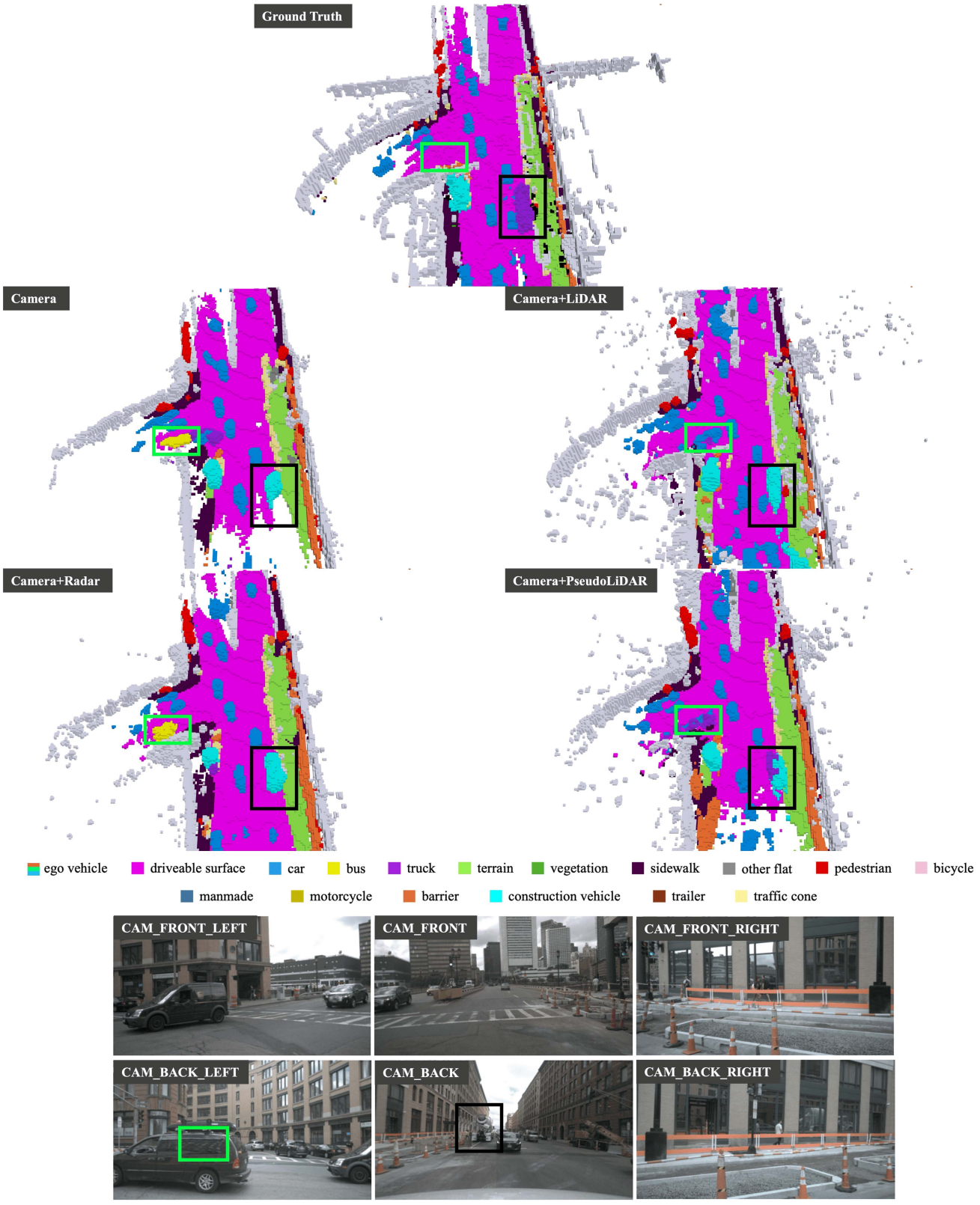

FIGURE 9. GaussianFormer Visual Comparison: Camera-only, Camera+LiDAR, Camera+Radar, and Camera+Pseudo-LiDAR models are compared. Visualizations correspond to the models presented in Table 12. Black and green boxes are overlaid for reference in the qualitative analysis discussion.

TABLE 15. GaussianOcc With Multimodal Fusion: The Image Feature Voxel Is Concatenated With an Occupancy Grid Derived From Point Cloud Data. MIoU Includes All Classes, While mIoU* Excludes the 'Other' and 'Other Flat' Classes. The Best-Performing Model Is Highlighted in **Bold**

Configuration	mIoU	mIoU*
Camera (original)	11.26	9.94
Camera+LiDAR	**11.60**	**10.25**
Camera+Radar	11.15	9.91
Camera+Radar Metadata	11.22	9.90
Camera+Pseudo-LiDAR	11.34	9.98

TABLE 16. Ablation on Additional LiDAR Sweeps and Pseudo-LiDAR Point Cloud Size: mIoU Includes All Classes, While mIoU* Excludes the 'Other' and 'Other Flat' Classes. The Best-Performing Model Is Highlighted in **Bold**

Configuration	mIoU	mIoU*
Camera (original)	11.26	9.94
Camera+LiDAR (1 sweep)	11.60	10.25
Camera+LiDAR (3 sweeps)	11.52	10.23
Camera+LiDAR (5 sweeps)	**11.65**	**10.30**
Camera+LiDAR (10 sweeps)	11.10	9.78
Camera+Pseudo-LiDAR 200×112	11.34	9.98
Camera+Pseudo-LiDAR 400×224	11.35	9.96
Camera+Pseudo-LiDAR 800×448	11.25	9.91

Firstly, it's noted that in the supervised occupancy prediction field, multimodal models typically employ an encoder before fusing with camera data [17], [32], [39]. In our approach, an encoder is not used, opting to utilize the data in its raw occupancy format, which proved beneficial in Simple-BEV. Similar to Simple-BEV, the model does implicitly encode this information, and GaussianOcc does this through a 3DCNN, as depicted in the architecture diagram in Fig. 6. Given that this is the first self-supervised, multimodal, semantic occupancy prediction model, it is challenging to attribute the performance to this aspect alone.

Secondly, the lack of performance gains can also be attributed to the self-supervised nature of the model. Unlike supervised models that compare the prediction directly against ground truth labels, the self-supervised approach involves back-projecting the 3D space to the camera view and comparing it to semantic labels generated from Grounded SAM [53], [9], [54], as first introduced in OccNeRF [55]. This method inherently limits the loss computation to only objects visible to the camera, excluding objects detectable solely by LiDAR, which do not contribute to the model's performance metrics. This often underscores the substantial benefits LiDAR brings to most BEV models, as it introduces new information not captured by camera-based systems.

This is further cemented by the fact that performance metrics are not calculated for the whole voxel grid and instead are only calculated on a subsection from which the camera can view. Consequently, a performance plateau may be reached, as we can only maximize what is observable through the camera.

2) ABLATION: MULTIPLE SWEEPS

An ablation is conducted to examine the effect that the total point cloud number has on performance in Table 16. For LiDAR, a general increase in performance is seen for up to five LiDAR sweeps; however, for ten sweeps, there is a notable drop in performance, lower than that of the original model. This is contrary to how supervised models tend to utilize ten LiDAR sweeps for optimal performance. When incorporating ten sweeps, the oldest sweep is roughly 450 ms out of date, which can introduce inaccuracies regarding vehicle positioning. For instance, a vehicle traveling at 50 km/h would move approximately 6.3 meters in this timeframe. A hypothesis might be that this outdated temporal information can potentially degrade the model's performance by providing misleading cues about current vehicle locations and such. For pseudo-LiDAR, utilizing a depth map of 200 × 112 or 400 × 224 pixels results in minimal performance differences, whereas an 800 × 448 pixel depth map leads to a drop in performance, which may be due to the inherent variability in training, where repeated runs can produce slightly different outcomes.

3) COMPARISON WITH STATE-OF-THE-ART

Similarly to GaussianFormer's analysis, the best-performing GaussianOcc Camera+LiDAR, Camera+Radar, and Camera+Pseudo-LiDAR models are compared to the SOTA **self-supervised** semantic occupancy prediction models evaluated on the Occ3D nuScenes dataset in Table 17.

One important aspect is the absence of other self-supervised multimodal models for comparison. This area of research has not received sufficient emphasis, which is strange, particularly given that the absence of required ground truth labels allows for significant flexibility regarding the resolution of the predictions and related aspects.

For the metrics, mIoU and mIoU*, the original GaussianOcc model surpasses all other Camera-only models, notably outperforming the next best model OccNeRF by 0.4 mIoU*. This makes the C+L GaussianOcc model the top performer in these categories compared to the Camera-only models. However, more nuanced observations emerge when analyzing specific labels. Static road-specific labels such as drivable surface, sidewalk, terrain, manmade, and vegetation do not see their best performance with GaussianOcc; instead, SelfOcc achieves superior results for most of these categories. In contrast, GaussianOcc excels with non-static labels like bicycles, cars, pedestrians, etc. While the top performers for these labels vary among the different GaussianOcc models, whether single-modal or multimodal, GaussianOcc demonstrates consistency across all labels. Conversely, SelfOcc, while excelling in categories like sidewalks, shows markedly poor performance for labels such as trailers and traffic cones.

TABLE 17. Comparison of Our Multimodal GaussianOcc Model With State-of-the-Art Self-Supervised Models. All Methods Are Validated Against Occupancy Labels From Occ3D [29]. GT Occ Refers to Using Dense Supervised Labels During Training. GT Pose Refers to Using the Ground Truth Pose During Training. MIoU Includes All Classes, and mIoU* Ignores the Classes 'Other' and 'Other Flat.' Modality: Camera (C), Lidar (L), Radar (R), Radar Metadata (RM), Pseudo-LiDAR (PL). The Best Performer in Each Category Is Highlighted in Bold

Method	Input Modality	GT Occ.	GT Pose	mIoU	mIoU*	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. surf.	sidewalk	terrain	manmade	vegetation
SimpleOcc [56]	C	✗	✓	7.99	7.05	0.67	1.18	3.21	7.63	1.02	0.26	1.80	0.26	1.07	2.81	40.44	18.30	17.01	**34.28**	10.84
SelfOcc [57]	C	✗	✓	10.54	9.30	0.15	0.66	5.46	12.54	0.00	0.80	2.10	0.00	0.00	8.25	**55.49**	**26.30**	**26.54**	14.22	5.60
OccNeRF [55]	C	✗	✓	10.81	9.54	0.83	0.82	5.13	12.49	3.50	0.23	3.10	1.84	0.52	3.90	52.62	20.81	24.75	18.45	**13.19**
GaussianOcc [16]	C	✗	✗	11.26	9.94	1.79	5.82	14.58	**13.55**	1.30	2.82	**7.95**	9.76	0.56	9.61	44.59	20.10	17.58	8.61	10.29
GaussianOcc [16] + Ours	C+L	✗	✗	**11.65**	**10.30**	2.05	5.76	14.13	13.31	3.75	3.43	7.36	**10.11**	0.87	**11.62**	43.04	20.07	18.40	8.82	12.05
GaussianOcc [16] + Ours	C+R	✗	✗	11.15	9.91	1.92	**5.99**	13.39	12.58	2.24	**3.58**	7.85	9.47	**1.08**	9.93	44.49	18.87	17.37	8.65	9.91
GaussianOcc [16] + Ours	C+RM	✗	✗	11.22	9.90	1.69	5.82	**14.30**	11.64	2.62	2.98	7.04	9.58	0.71	9.75	44.69	20.28	18.12	8.88	10.22
GaussianOcc [16] + Ours	C+PL	✗	✗	11.34	9.98	**2.15**	4.84	13.35	12.94	**4.35**	2.90	7.43	9.58	1.03	11.56	41.11	20.60	18.57	8.76	10.95

4) QUALITATIVE ANALYSIS

For qualitative analysis, we analyze the same sample used in the GaussianFormer section for consistency. Comparing the Camera-only model to the ground truth in Fig. 10, a marked disparity in the segmentation quality can be observed, especially in areas obscured from the camera's view, such as behind vehicles and buildings. This discrepancy occurs primarily due to the absence of 3D labels, limiting the model's ability to learn and infer information beyond what is visible from the camera's perspective, significantly restricting its predictive capabilities.

Additionally, a 'ghosting' effect is present for all models, previously discussed in Section IV-B3 during the qualitative analysis for Lift-Splat-Shoot. Previously, this effect was attributed to uncertainty stemming from insufficient training data. In the current context, while the model accurately detects vehicles in the scene, the lack of comprehensive ground truth labels and reliance on simple 2D semantic supervision prevents it from resolving depth ambiguities. Although one might expect that depth information from LiDAR would mitigate this issue, the ghosting effect persists, indicating that additional model adjustments or data integration strategies might be necessary.

Furthermore, a common error across all models is the misclassification of the road in the rear camera view, highlighted by a black rectangle, where they incorrectly predict the presence of a wall. This contradicts the ground truth and the RGB image, where the road is visible. This misprediction highlights a critical area where the models need refinement to enhance their environmental awareness and accuracy in real-world scenarios.

Overall, it is challenging to conclusively determine which model performed better based on these observations. The self-supervised nature of the models limits the effectiveness of depth information, suggesting that more innovative approaches need to be explored to enhance the viability of self-supervised multimodal models. This may involve developing new methods to integrate and utilize the depth data in a self-supervised learning context.

F. DISCUSSION

This section provides a summary of the experimental results.

Lift-Splat-Shoot: The implementation of DINOv2 and Metric3Dv2 resulted in a significant enhancement in performance, achieving an increase of 7.4 IoU compared to the baseline model, which recorded 33.0 IoU. This improvement was realized even though only half of the dataset was used for training. Foundation models, due to their pretrained nature, serve as a robust starting point for BEV perception models and present a strong alternative to the commonly used ResNet-101 backbone. Nevertheless, the Giant variant of the DINOv2 model required 900 seconds to complete 1,000 training iterations, compared to 120 seconds for the original model, indicating that increased performance comes with certain trade-offs. The Small variant completed the same number of iterations in 300 seconds, with only a slight reduction in performance relative to the Giant variant.

Simple-BEV: Leveraging LiDAR and radar to enhance performance beyond the baseline Camera-only model proved useful in the original paper's study. In a similar fashion, the substitution with pseudo-LiDAR yielded a 2.9 IoU increase over the Camera-only setup by resolving depth ambiguities without the need for additional sensors. However, Metric3Dv2's Giant variant demonstrated slow processing times, handling only one image per second at the original nuScenes image size, which limits its practical application in real-time scenarios. Despite this, our experiment illustrates that foundation models are versatile, serving not only as replacements for traditional model backbones but also in conjunction with them.

GaussianFormer: Multimodal deployment improved performance relative to the baseline model by 9.4 mIoU and nearly matched the SOTA DAOcc model, which exceeded it by 1.79 mIoU. Employing Gaussians for scene representation effectively facilitates multimodal integration. Replacing Gaussian *xyz* with point cloud data and adding additional Gaussians proved most beneficial, aligning well with the model's learning of *xyz* positions. A significant issue is a

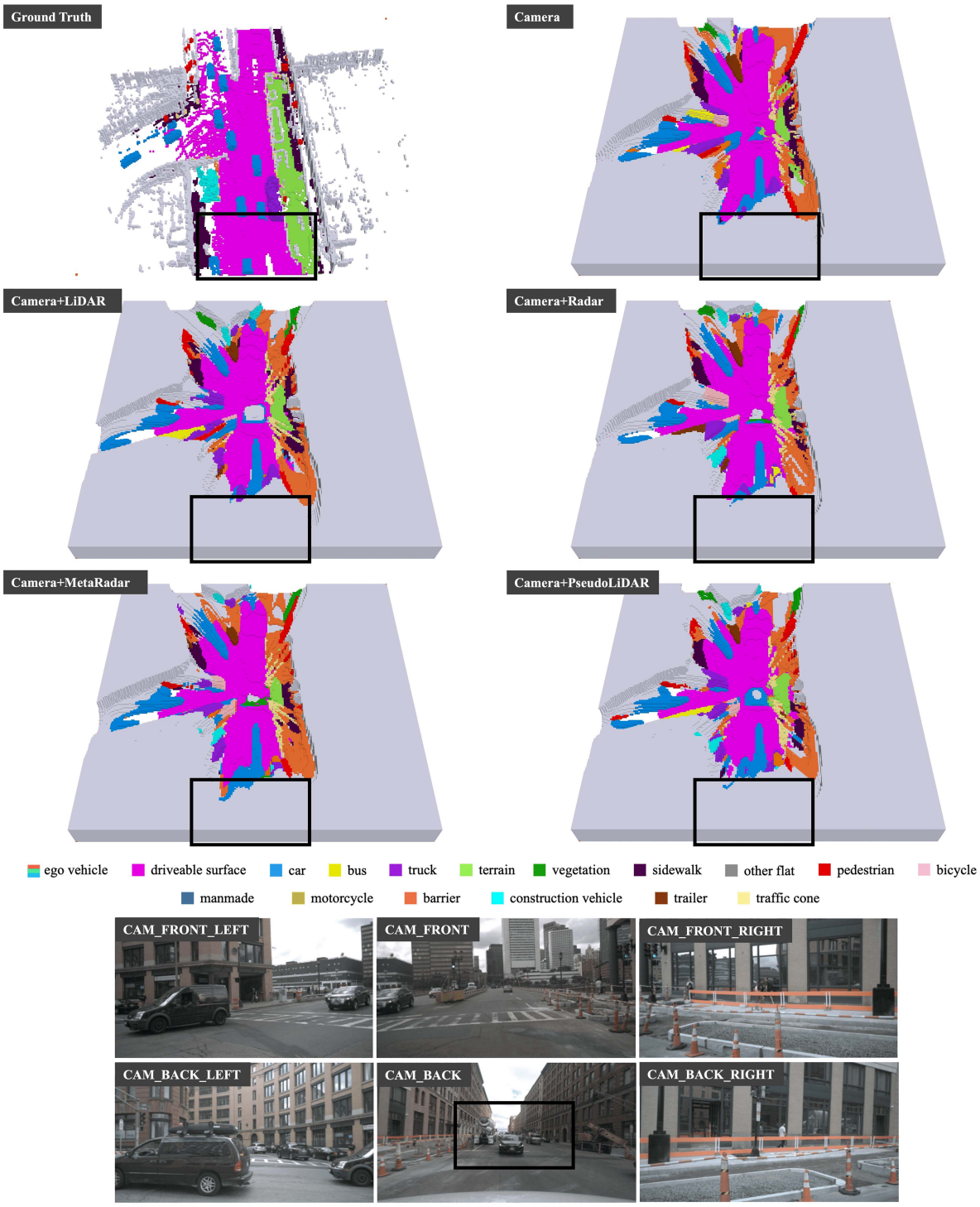

FIGURE 10. GaussianOcc Visual Comparison: Comparison of Camera-only, Camera+LiDAR, Camera+Radar, Camera+Radar Metadata, and Camera+Pseudo-LiDAR models. Visualizations are generated from the GaussianOcc models listed in Table 17. Ground truth labels are not used in loss computation due to the model's self-supervised training paradigm. A black rectangle is overlaid to support the discussion in the qualitative analysis.

substantial increase in memory usage with additional LiDAR sweeps; for instance, configurations with five sweeps exceeded the available 40GB of GPU memory, though a single sweep already offered a considerable advantage.

GaussianOcc: Our alteration to the model introduced the first self-supervised, multimodal, semantic occupancy prediction model. The inclusion of LiDAR provided only a marginal performance improvement of 0.36 mIoU, likely limited by the self-supervised approach of the model, which constrained its

learning capacity and acted as a bottleneck on performance enhancement. Nonetheless, these findings pave the way for further exploration into self-supervised multimodal models.

Overall, the results demonstrate that foundation models and multi-sensor integration significantly boost BEV perception. Implementation of models such as DINOv2 and Metric3Dv2 dramatically enhances evaluation metrics, even with limited data. Additionally, using diverse sensor setups, including LiDAR and radar, effectively resolves depth ambiguities and improves model robustness, indicating a promising direction for enhancing autonomous vehicle technologies.

V. CONCLUSION

This paper details the deployment of foundation models DINOv2 and Metric3Dv2, as well as multi-sensors, in already existing BEV perception models. The implementation shows great improvement over the baseline models, with additional future work to be conducted:

1) A thorough evaluation of whether foundation models and multimodal fusion are viable for industrial applications, considering cost and processing time. In addition, exploration of the application of these methods in other domains including drones and warehouse autonomation.
2) A more extensive analysis of challenging driving conditions, such as rain and nighttime scenarios, to assess the generalizability of foundation models and the robustness of LiDAR and radar. Additional augmentations such as data noise and sensor misalignment can be investigated.
3) Investigating fine-tuning the DINOv2 model for Lift-Splat-Shoot, with the potential use of low-rank approximations to enhance performance [51], [52].
4) Integrating LiDAR, radar, and pseudo-LiDAR data into additional self-supervised semantic occupancy prediction models to further explore the performance of the multimodal GaussianOcc model.

Overall, this study highlights the essential role of foundation and multimodal models in improving the accuracy and precision of scene predictions for autonomous vehicles. The real-time deployment of large-scale models alongside multiple sensors has become practical. Tesla utilizes a large-scale foundation model for occupancy prediction with commendable performance, using only camera inputs. In contrast, other autonomous vehicle systems, such as Waymo, employ a combination of cameras, LiDAR, and radar. This diversity underscores that this field is a vibrant area of research with strong support from ongoing developments.

ACKNOWLEDGMENT

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant 18/CRT/6049. For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

REFERENCES

[1] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, "Simple-BEV: What really matters for multi-sensor BEV perception?," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 2759–2765.
[2] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12689–12697.
[3] Z. Liu et al., "BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 2774–2781.
[4] J. Schramm et al., "BEVCar: Camera-radar fusion for BEV map and object segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 1435–1442.
[5] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11618–11628.
[6] J. Philion and S. Fidler, "Lift-splat-shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K.: Springer-Verlag, 2020, pp. 194–210.
[7] Y. Xiong et al., "EfficientSAM: Leveraged masked image pretraining for efficient segment anything," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 16111–16121.
[8] A. Bochkovskii et al., "Depth Pro: Sharp monocular metric depth in less than a second," in *Proc. Int. Conf. Learn. Representations*, 2025.
[9] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.
[10] N. Pinetsuksai et al., "Development of self-supervised learning with DINOv2-distilled models for parasite classification in screening," in *Proc. 15th Int. Conf. Inf. Technol. Elect. Eng.*, 2023, pp. 323–328.
[11] M. Hu et al., "Metric3Dv2 v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10579–10596, Dec. 2024.
[12] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8437–8445.
[13] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, vol. 97, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html
[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[15] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "GaussianFormer: Scene as Gaussians for vision-based 3D semantic occupancy prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 376–393.
[16] W. Gan, F. Liu, H. Xu, N. Mo, and N. Yokoya, "GaussianOcc: Fully self-supervised and efficient 3D occupancy estimation with Gaussian splatting," *CoRR*, vol. abs/2408.11447, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2408.11447
[17] Z. Yang, Y. Dong, and H. Wang, "DAOcc: 3D object detection assisted multi-sensor fusion for 3D occupancy prediction," 2024, *arXiv:2409.19972*.
[18] S. Hayes, G. Sistu, and C. Eising, "Revisiting birds eye view perception models with frozen foundation models: DINOv2 and Metric3Dv2," *Electron. Imag.*, vol. 37, no. 15, pp. 1–6, 2025. [Online]. Available: https://library.imaging.org/ei/articles/37/15/AVM-112
[19] H. Li et al., "Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 4, pp. 2151–2170, Apr. 2024.
[20] H. Xu, J. Chen, S. Meng, Y. Wang, and L.-P. Chau, "A survey on occupancy perception for autonomous driving: The information fusion perspective," *Inf. Fusion*, vol. 114, 2025, Art. no. 102671. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253524004494
[21] Z. Li et al., "BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 1–18.
[22] Y. Kim, J. Shin, S. Kim, I.-J. Lee, J. W. Choi, and D. Kum, "CRN: Camera radar net for accurate, robust, efficient 3D perception," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 17615–17626.

[23] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, pp. 1231–1237, 2013.
[24] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9297–9307 .
[25] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4340–4349.
[26] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, J. Zhou, and J. Lu, "SurroundOcc: Multi-camera 3D occupancy prediction for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 21672–21683.
[27] W. Tong et al., "Scene as occupancy," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 8372–8381.
[28] B. Zhu, Z. Wang, and H. Li, "nuCraft: Crafting high resolution 3D semantic occupancy for unified 3D scene understanding," in *Proc. IEEE/CVF Conf. Eur. Conf. Comput. Vis.*, 2024, pp. 125–141.
[29] X. Tian et al., "Occ3D: A large-scale 3D occupancy prediction benchmark for autonomous driving," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, vol. 36, pp. 64318–64330.
[30] A.-Q. Cao and R. de Charette, "MonoScene: Monocular 3D semantic scene completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*., 2022, pp. 3991–4001.
[31] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-perspective view for vision-based 3D semantic occupancy prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*., 2023, pp. 9223–9232.
[32] Z. Ming, J. S. Berrio, M. Shan, and S. Worrall, "OccFusion: Multi-sensor fusion framework for 3D semantic occupancy prediction," *IEEE Trans. Intell. Veh.*, early access, Sep. 03, 2024, doi: 10.1109/TIV.2024.3453293.
[33] Z. Ming, J. S. Berrio, M. Shan, and S. Worrall, "InverseMatrixVT3D: An efficient projection matrix-based approach for 3D occupancy prediction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 9565–9572, doi: 10.1109/IROS58592.2024.10802434.
[34] M. Pan et al., "RenderOcc: Vision-centric 3D occupancy prediction with 2D rendering supervision," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 12404–12411.
[35] L. Roldão, R. de Charette, and A. V. Blondet, "LMSCNet: Lightweight multiscale 3D semantic completion," in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 111–119.
[36] Z. Murez, T. Van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich, "Atlas: End-to-end 3D scene reconstruction from posed images," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 414–431.
[37] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–14, 2023.
[38] B. Mildenhall, P.P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NERF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, 2022.
[39] J. Pan, Z. Wang, and L. Wang, "Co-Occ: Coupling explicit feature fusion with volume rendering regularization for multi-modal 3D semantic occupancy prediction," *IEEE Robot. Automat. Lett.*, vol. 9, no. 6, pp. 5687–5694, Jun. 2024.
[40] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[41] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*. Red Hook, NY, USA, 2020, pp. 9912–9924.
[42] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
[43] S. Aharon et al., "Super-gradients," zenodo, 2021. [Online]. Available: https://zenodo.org/record/7789328
[44] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6578–6587.
[45] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 6000–6010.
[46] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6602–6611.
[47] L. Yang et al., "Depth anything v2," in *Proc. 38th Annu. Conf. Neural Inf. Process. Syst.*, 2024, pp. 21875–21911. [Online]. Available: https://openreview.net/forum?id=cFTi3gLJ1X
[48] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12159–12168.
[49] P. Droździel, S. Tarkowski, I. Rybicka, and R. Wrona, "Drivers 'reaction time research in the conditions in the real traffic," *Open Eng.*, vol. 10, pp. 35–47, 2020.
[50] X. Wang et al., "OpenOccupancy: A large scale benchmark for surrounding semantic occupancy perception," in *Proc. IEEE/CVF Int. Conf. Comput. Vis*., 2023, pp. 17850–17859.
[51] E. J. Hu et al., "Lora: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Representations*, 2022.
[52] M. R. Barin, G. Aydemir, and F. Güney, "Robust bird's eye view segmentation by adapting DINOv2," 2024, *arXiv:2409.10228*.
[53] S. Liu et al., "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," in *Proc. 18th Eur. Conf. Comput. Vis.*, Berlin, Germany, 2024, pp. 38–55, doi: 10.1007/978-3-031-72970-6_3.
[54] T. Ren et al., "Grounded SAM: Assembling open-world models for diverse visual tasks," 2024, *arXiv:2401.14159*.
[55] C. Zhu, R. Wan, Y. Tang, and B. Shi, "Occlusion-free scene recovery via neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 20722–20731.
[56] W. Gan, N. Mo, H. Xu, and N. Yokoya, "A comprehensive framework for 3D occupancy estimation in autonomous driving," *IEEE Trans. Intell. Veh.*, pp. 1–19, 2024, doi: 10.1109/TIV.2024.3403134.
[57] Y. Huang, W. Zheng, B. Zhang, J. Zhou, and J. Lu, "SelfOcc: Self-supervised vision-based 3D occupancy prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 19946–19956.

SEAMIE HAYES (Graduate Student Member, IEEE) received the B.Sc degree in mathematical sciences from the University of Limerick, Limerick, Ireland, in 2023. He is currently working toward a full-time structured Ph.D. degree with the CRT Programme in the Department of Electronic and Computer Engineering, University of Limerick. His research interests include autonomous vehicles with an emphasis on multi-sensor fusion and sensor modality robustness for birds eye view perception.

GANESH SISTU is currently a Principal AI Architect with Valeo, Tuam, Ireland, and also an Adjunct Assistant Professor with the University of Limerick, Limerick, Ireland. He is leading Multiple Global Research Teams working on automated driving and parking. With more than 14 years in computer vision and machine learning, he has authored or coauthored more than 35 publications in top-tier conferences like ICCV and ICRA. He plays a pivotal role in guiding the future of AI education as an Industry Board Member for Science Foundation Ireland's Data Science Ph.D. degree and the National M.Sc. degree in AI programs with the University of Limerick, blending academic insight with industry expertise.

CIARÁN EISING (Senior Member, IEEE) received the B.E. degree in electronic and computer engineering and the Ph.D. degree from the NUI Galway, Galway, Ireland, in 2003 and 2010, respectively. From 2009 to 2020, he was a Computer Vision Architect and Senior Expert with Valeo. In 2020, he joined the University of Limerick, Limerick, Ireland, as an Associate Professor.