2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

VG3T: Visual Geometry Grounded Gaussian Transformer

Junho Kim and Seongwon Lee†

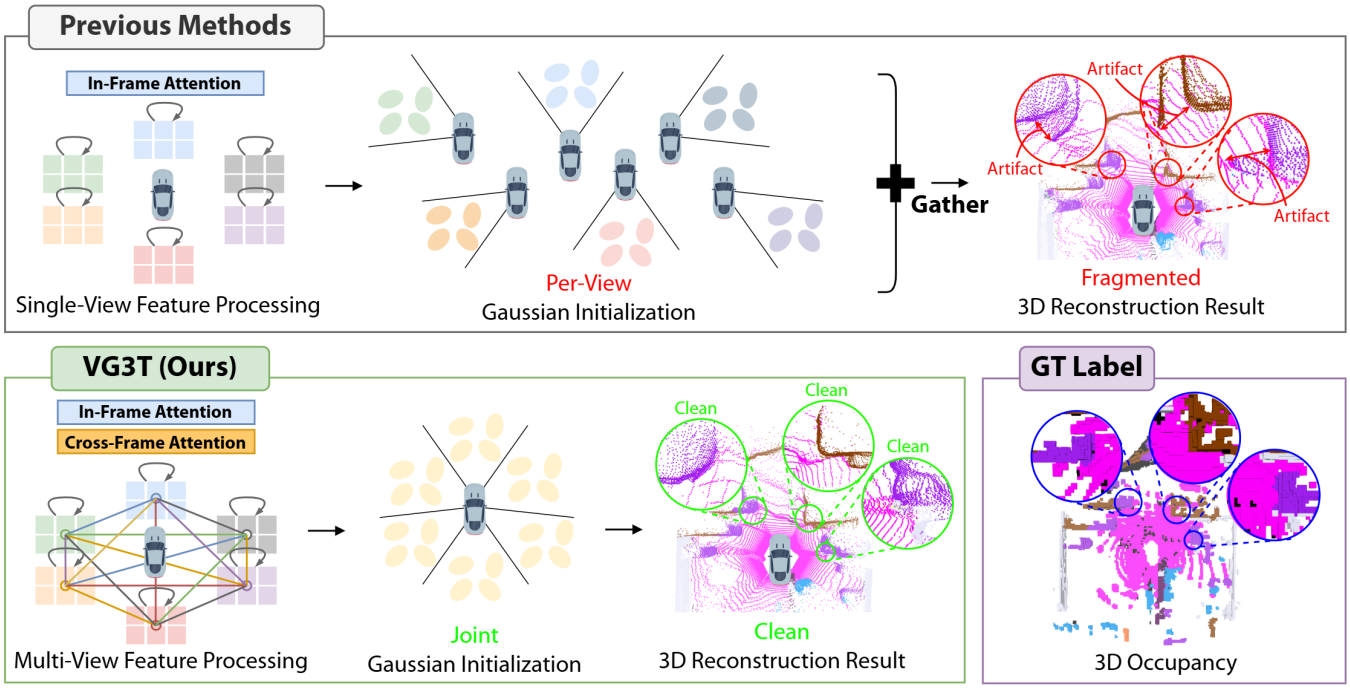

Fig. 1: **Coherent 3D Scene Representation via Early Multi-View Fusion.** Unlike prior work that processes each camera view independently, leading to fragmented and inconsistent 3D representations, our method VG3T, leverages cross-view correlation. This produces a coherent and geometrically accurate 3D representation.

***Abstract*— Generating a coherent 3D scene representation from multi-view images is a fundamental yet challenging task. Existing methods often struggle with multi-view fusion, leading to fragmented 3D representations and sub-optimal performance. To address this, we introduce VG3T, a novel multi-view feed-forward network that predicts a 3D semantic occupancy via a 3D Gaussian representation. Unlike prior methods that infer Gaussians from single-view images, our model directly predicts a set of semantically attributed Gaussians in a joint, multi-view fashion. This novel approach overcomes the fragmentation and inconsistency inherent in view-by-view processing, offering a unified paradigm to represent both geometry and semantics. We also introduce two key components, Grid-Based Sampling and Positional Refinement, to mitigate the distance-dependent density bias common in pixel-aligned Gaussian initialization methods. Our VG3T shows a notable 1.7%p improvement in mIoU while using 46% fewer primitives than the previous state-of-the-art on the nuScenes benchmark, highlighting its superior efficiency and performance. Codes are available at `https://github.com/junho2000/VG3T`.**

I. INTRODUCTION

Vision-centric systems [1], [2], [3], [4], [5], [6], [7] are gaining prominence in autonomous driving due to their inherent cost-effectiveness compared to LiDAR-based solutions [8], [9], [10], [11], [12], [13], [14], [15]. However, these systems face a critical challenge, a lack of explicit 3D geometric and semantic understanding. This limitation manifests as fragmented 3D scene representations and can compromise safety when navigating complex and dynamic environments. The emergence of 3D semantic occupancy prediction directly addresses this limitation by generating a fine-grained representation of the surrounding environment, jointly capturing both its geometry and semantic meaning [16], [17], [18], [19], [20], [21], [22], [23]. While promising, 3D occupancy methods have struggled with efficiency, primarily due to the use of dense representations like voxels. These voxel-based methods, while offering high fidelity, are computationally intensive and inefficient as they fail to leverage the inherent spatial sparsity of real-world scenes. This has motivated a shift towards more efficient and sparse representations [24], [25]. A promising direction models occupancy as 3D Gaussians with learnable attributes [26], [27]. This approach is particularly compelling due to its ability to capture the spatial sparsity of real-world scenes, providing a flexible and differentiable representation that models complex geometries with high computational efficiency.

Despite the promise of 3D Gaussian representations, existing methods, such as GaussianFormer [26] and

J. Kim, and S. Lee are with the School of Electrical Engineering, Kookmin University, Seoul 02707, South Korea, {jhk00, sungonce}@kookmin.ac.kr.
† denotes Corresponding Author.

GaussianFormer-2 [27], face two critical challenges. First, they rely on a fragmented, view-by-view paradigm where features from each camera are processed independently. This approach makes it difficult to establish coherent cross-view correspondences, resulting in inconsistent and geometrically fragmented 3D representations. A second challenge is a distance-dependent density bias, where Gaussians are over-sampled near the camera and under-represented in distant regions. This imbalance compromises both efficiency and accuracy, as it leads to redundant primitives and a lack of fine-grained detail where it is most needed. These two limitations, fragmented multi-view fusion and density bias, collectively hinder the generation of accurate and efficient 3D occupancy predictions vital for autonomous driving.

In this paper, we introduce Visual Geometry Grounded Gaussian Transformer, dubbed as VG3T, a novel multi-view feed-forward network that directly addresses these challenges. VG3T models a set of semantically-attributed 3D Gaussians from surround-view images. Unlike prior work [26], [27], our model is a unified, end-to-end framework that leverages early multi-view feature correlation. This allows us to fuse multi-view information more coherently, leading to precise geometric and semantic predictions that capture the scene's full 3D structure without relying on external supervisory data. To mitigate the critical distance-dependent density bias, our approach employs a two-staged strategy. First, a Grid-Based Sampling module efficiently removes redundant Gaussians from over-represented areas, guaranteeing an even spatial distribution. Second, the Residual Refinement module precisely adjusts the properties of the remaining Gaussians, enabling them to better capture fine details in previously under-represented areas.

Our main contributions can be summarized as follows:

- We present VG3T, a novel multi-view feed-forward network that directly predicts 3D semantic occupancy with a Gaussian representation. By leveraging early cross-view feature correlation, our model overcomes the fragmentation inherent in existing view-by-view approaches, enabling a more unified and coherent 3D scene understanding.
- To address the fundamental problem of distance-dependent density bias, we propose a two-staged strategy: Grid-Based Sampling, which efficiently removes redundant primitives from over-represented areas, and a Positional Refinement module, which precisely adjusts remaining Gaussians to capture fine-grained details.
- Our end-to-end trainable model, VG3T, achieves a new state-of-the-art on the nuScenes 3D semantic occupancy benchmark. We demonstrate a notable 1.7%p improvement in mIoU while using 46% fewer primitives than the previous state-of-the-art method, highlighting the superior efficiency and effectiveness of our approach.

II. Related Work

A. 3D Semantic Occupancy Prediction

3D semantic occupancy prediction is a critical task for autonomous driving, providing a comprehensive and fine-grained understanding of the surrounding environment. While LiDAR-based methods offer strong performance, their high cost and weather-related vulnerabilities have spurred the development of camera-based alternatives. Early vision-centric methods relied on dense voxel-based representations [18], [19]. While these methods achieve high fidelity, they are computationally and memory-intensive as they inefficiently model empty space. This has motivated the exploration of more efficient scene representations. Another notable direction involves planar-based methods BEVFormer [28], TPVFormer [29], which project 3D information onto 2D grids. However, they also suffer from the inefficiency of a dense representation and, more critically, inherently lose crucial spatial information.

This has led to a shift toward more efficient and sparse representations that model only the non-empty parts of a scene. While point-based models offer a sparse representation, each point lacks a defined spatial volume [30], [31]. As a more expressive alternative, 3D Gaussian models GaussianFormer [26], GaussianFormer-2 [27] have emerged. These methods capture complex geometries with a compact set of learnable primitives, each defined by attributes like position, shape, and semantics. This representation offers high expressive power while maintaining computational efficiency. However, existing Gaussian methods have limitations. Specifically, they suffer from two critical shortcomings: (1) an inefficient multi-view aggregation scheme that relies on processing each camera view independently, leading to fragmented 3D representations; and (2) a distance-dependent density bias that compromises both accuracy and efficiency. To the best of our knowledge, our proposed VG3T framework is the first to directly and effectively address both of these fundamental challenges, unlocking the full potential of Gaussian representations for 3D occupancy prediction.

B. Multi-view 3D Reconstruction

For 3D semantic occupancy prediction, a critical aspect of multi-view systems is how they effectively aggregate cross-view information. Most contemporary methods employ a late-fusion strategy, where features are first extracted from each camera independently before being fused into a common 3D representation [18], [26], [27], [28], [29]. The fundamental limitation of this approach is its lack of explicit geometric correspondence at the initial image feature level. This can result in geometric inconsistencies and fragmented 3D representations, a key bottleneck for robust scene understanding.

In contrast, the field of multi-view reconstruction has made significant progress by establishing dense correspondences through the joint processing of multiple images [32], [33], [34]. Recent models, such as Visual Geometry Grounded Transformer (VGGT) [35], demonstrate this by directly ingesting multiple images to produce a unified geometric understanding. Inspired by this success, our work introduces this early-fusion paradigm to the domain of 3D semantic occupancy prediction. By employing a pre-trained VGGT as our feature backbone, we leverage rich, geometrically-

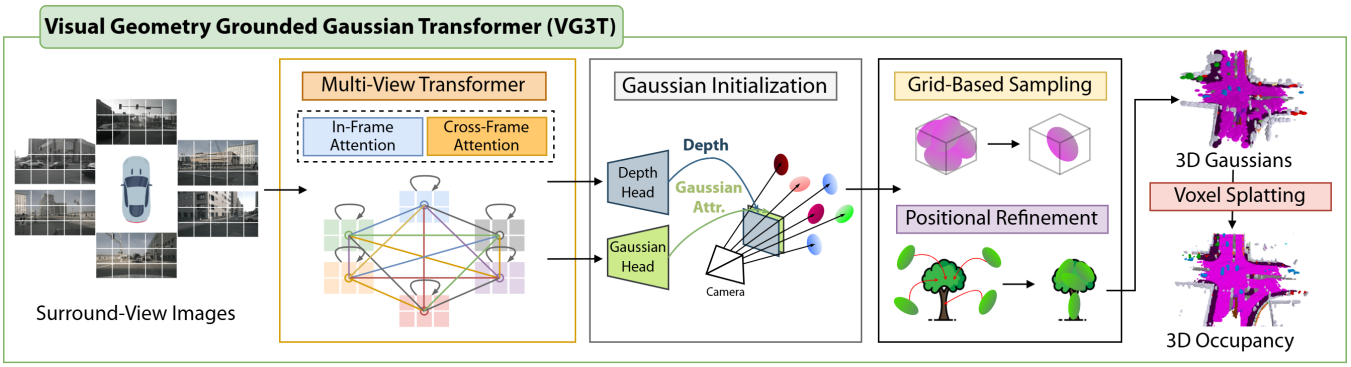

Fig. 2: Overview of the Visual Geometry Grounded Gaussian Transformer (VG3T) Architecture.

consistent features from the very beginning of our pipeline. This early, coherent fusion provides a robust foundation, enabling our model to overcome the limitations of late-fusion methods and produce a more accurate and consistent 3D representation.

III. Visual Geometry Grounded Gaussian Transformer (VG3T)

In this section, we introduce a novel end-to-end multi-view network for 3D semantic occupancy prediction, named Visual Geometry Grounded Gaussian Transformer (VG3T).

A. Problem Setup and Overview

The goal of 3D semantic occupancy prediction is to generate a comprehensive understanding of the surrounding environment from multi-view images, capturing both its fine-grained geometry and semantic meaning. Specifically, given a set of multi-view input images $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^{N}$, where N is the number of camera views, the task is to predict a dense semantic occupancy grid $\mathbf{O} \in \mathcal{C}^{X \times Y \times Z}$. Here, X, Y, Z define the spatial dimensions of the grid, and $\mathcal{C}$ represents the number of semantic classes.

Unlike traditional methods that directly regress dense voxel grids, VG3T adopts a more efficient and flexible intermediate representation: a set of sparse, learnable 3D Gaussian primitives. Our network is designed to predict a set of 3D Gaussians, $\mathcal{G} = \{\mathbf{G}_i\}_{i=1}^{P}$, where P is the number of Gaussians, which collectively model the geometry and semantics of the scene. Each Gaussian primitive $\mathbf{G}_i$ is a parametric entity defined by a set of learnable attributes: its mean (position) $m_i \in \mathbb{R}^3$, scale $s_i \in \mathbb{R}^3$, rotation $r_i \in \mathbb{R}^4$, opacity $a_i \in [0, 1]$, and semantic features $c_i \in \mathbb{R}^C$. The final semantic occupancy grid $\mathbf{O}$ is then rendered by probabilistically aggregating this collection of predicted Gaussians into a dense voxel grid. As shown in Figure 2, our end-to-end framework consists of four main stages: a multi-view feature backbone, an initial Gaussian prediction, a two-staged module comprising Grid-Based Sampling and Positional Refinement, and a final rendering step.

B. Multi-View Transformer Backbone

The core of VG3T is a powerful feature backbone designed to build a holistic and geometrically consistent understanding of the scene from the earliest stages. Existing methods typically extract features from each camera view independently before a late-fusion step, resulting in fragmented 3D representations. To overcome this fundamental limitation, we leverage the Visual Geometry Grounded Transformer (VGGT) [35] as our backbone. The VGGT architecture, characterized by its minimal 3D inductive biases, enables our model to learn complex geometric correlations directly from large-scale multi-view data.

Specifically, we adapt the VGGT architecture to process features from six surround-view cameras. First, each input image $\mathbf{I}_i \in \mathbb{R}^{3 \times H \times W}$ is tokenized via DINOv2 [36] feature extractor into a set of visual tokens $t_i^I \in \mathbb{R}^{K \times C}$, where K is the number of tokens and C is the feature dimension of each token. These tokens are augmented with a small set of learnable register tokens t^R, and the full set of tokens from all six views is aggregated into a single sequence. This sequence is then processed by the VGGT's alternating attention mechanism, which strategically interleaves in-frame attention (refining features within each view) and cross-frame attention (fusing features across all views).

The in-frame attention operates on the concatenated tokens of a single view $\mathbf{T}_i = \{t_i^I, t_i^R\}$. It refines features based on local semantic and geometric context within each camera view independently. The process is formulated for each view $i \in \{1, \ldots, N\}$ by first computing the query, key, and value matrices via linear projections:

$$\mathbf{Q}_i = \mathbf{T}_i W_{in}^Q, \quad \mathbf{K}_i = \mathbf{T}_i W_{in}^K, \quad \mathbf{V}_i = \mathbf{T}_i W_{in}^V. \quad (1)$$

The refined tokens are then computed using the scaled dot-product attention formula:

$$\mathbf{T}_i' = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{d_k}}\right)\mathbf{V}i. \quad (2)$$

The cross-frame attention then applies self-attention across the entire combined set of tokens from all views, $\mathcal{T} = \bigcup_{i=1}^{N} \mathbf{T}_i'$. This is the crucial step for early multi-view fusion, enabling tokens from any one view to aggregate information from tokens across all other views. This global operation is formulated as:

$$\mathbf{Q} = \mathcal{T} W_{cross}^Q, \quad \mathbf{K} = \mathcal{T} W_{cross}^K, \quad \mathbf{V} = \mathcal{T} W_{cross}^V. \quad (3)$$

$$\mathcal{T}' = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}. \quad (4)$$

This alternating pattern ensures that each token is iteratively refined by both its in-frame and cross-frame context. This approach is our key to building a robust 3D representation, as it ensures that the visual tokens are enriched with multi-view geometric awareness from the very beginning of the pipeline. The final output is a set of refined visual tokens, $\hat{t}_i^I$, which serve as a powerful foundation for the subsequent dense prediction of 3D Gaussian primitives.

C. Gaussian Initialization

Following the multi-view feature fusion in the transformer backbone, the cross-view aware visual tokens $\hat{t}_i^I$ for each view are decoded into dense feature maps. Then we use Dense Prediction Transformer (DPT) [37] layer to transform the refined output token $\hat{t}_i^I$, into dense feature maps $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$. These rich, cross-view correlated feature maps serve as the common input for two parallel prediction heads.

The first head, the Depth head, is responsible for predicting the per-pixel geometry. It consists of a convolutional layer, which we denote as f_{depth}. This head processes the dense feature map $\mathbf{F}$ to produce a dense depth map $\mathbf{D} \in \mathbb{R}^{(H/r)\times(W/r)}$ at a down-sampled ratio r, along with corresponding aleatoric uncertainty maps $\Sigma^D \in \mathbb{R}+^{(H/r)\times(W/r)}$. This uncertainty is incorporated into our training objective, as it helps the model reason about its confidence in each depth prediction. The process is summarized as follows:

$$(\mathbf{D}, \Sigma^D) = f_{depth}(\mathbf{F}). \tag{5}$$

Concurrently, the second head, the Gaussian head, predicts the remaining physical and semantic attributes of the Gaussians. This head is implemented as a Multi-Layer Perceptron (MLP), which we denote as f_{attr}. It maps the each feature $\mathbf{F}_i$ in the dense feature map $\mathbf{F}$ to Gaussian attributes $\mathbf{A}_i = \{s_i, r_i, a_i, c_i\}$. The prediction is formulated as

$$\mathbf{A}_i = f_{attr}(\mathbf{F}_i). \tag{6}$$

Finally, we initialize a complete set of 3D Gaussians, $\mathcal{G}$, by unprojecting each pixel from all N views into 3D space. For each pixel, the 3D mean μ is computed using the known camera parameters and the predicted depth d:

$$\mu = \mathbf{o} + d \cdot \mathbf{v}. \tag{7}$$

Here, o represents the camera's origin, and v is the viewing ray direction for the corresponding pixel. The scale, rotation, opacity, and semantics are directly assigned from the corresponding Gaussian attribute A_i, resulting in the final collection of initialized Gaussians set $\mathcal{G}$.

D. Grid-Based Sampling

To address the distance-dependent density bias inherited from the initialization stage, we propose a two-staged approach that first removes redundant Gaussians and then adjusts their position. The first stage, grid-based sampling, directly targets the problem of primitive redundancy to generate a more uniform spatial distribution.

To achieve this, we first discretize the continuous 3D space by partitioning it into a regular grid. Given the set of initial Gaussians $\mathcal{G}$, we map each mean position $\boldsymbol{\mu}_i \in \mathbb{R}^3$ to a integer grid coordinate $\mathbf{v}_i \in \mathbb{Z}^3$. This is done by scaling the mean position with a pre-defined grid size s_g and taking the floor of the result. To efficiently group all Gaussians that fall within the same voxel, we convert each 3D integer coordinate $\mathbf{v}_i$ into a unique 1D hash key k_i. This combined voxelization and hashing process is formulated as

$$\mathbf{v}_i = \lfloor \boldsymbol{\mu}_i / s_g \rfloor, \quad k_i = H(\mathbf{v}_i). \tag{8}$$

This hashing step allows us to group primitives in linear time by sorting the 1D keys, thereby avoiding computationally expensive 3D neighborhood searches and minimizing latency.

After grouping the Gaussians by their hash keys, we obtain a set of non-empty voxel groups $\mathcal{V}$, where each voxel v_i in $\mathcal{V}$ contains all Gaussians that share the same associated key. Within each occupied voxel containing multiple Gaussians, we then randomly select a single representative Gaussian $\mathcal{G}_i'$ and prune all others. The final, sampled set of Gaussians, $\mathcal{G}_s = \{\mathbf{G}_i'\}_{\forall i}$, is then formed by collecting the single representative from each voxel. This process effectively down-samples over-represented regions, such as those near the camera, while leaving the original sparse representation of distant areas intact, thereby mitigating the initial density bias and improving efficiency.

E. Positional Refinement

While grid-based sampling effectively addresses primitive redundancy, the complementary challenge is to enrich the sparse, under-represented regions of the scene. To this end, the second stage of our approach, positional refinement, performs a learnable process to refine the positions of the sampled primitives.

We treat the sampled set of Gaussians and their corresponding features as a sparse 3D point, which is processed by a refinement network, f_r. The refinement network consists of 3D sparse convolutional layer, takes as input the features $\mathbf{F}_i$ assigned to each Gaussian $\mathbf{G}_i$ in the sampled Gaussian set $\mathcal{G}_s$ and predicts a set of weights, $\mathbf{w}_i \in \mathbb{R}^K$, for a predefined basis $\mathbf{B} \in \mathbb{R}^{K\times 3}$. This basis is composed of vectors aligned with the axes, constraining the adjustments.

This allows our model to learn a robust and expressive positional offset, $\Delta\boldsymbol{\mu}_i$, as a linear combination of the basis vectors. The final refined position, μ_i', is then computed by adding this offset to the original position, μ_i.

$$\mathbf{w}_i = f_r(\mathbf{F}_i), \quad \Delta\boldsymbol{\mu}_i = \mathbf{B}^T\sigma(\mathbf{w}_i), \quad \boldsymbol{\mu}_i' = \boldsymbol{\mu}_i + \Delta\boldsymbol{\mu}_i, \tag{9}$$

where σ denotes the sigmoid function.

This two-stage approach ensures a well-distributed set of primitives that capture essential scene details while significantly reducing the total number of primitives required, yielding the final set of Gaussians, $\hat{\mathcal{G}}$.

TABLE I: **3D semantic occupancy prediction results on nuScenes benchmark.**

Method	IoU	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
MonoScene [16]	23.96	7.31	4.03	0.35	8.00	8.04	2.90	0.28	1.16	0.67	4.01	4.35	27.72	5.20	15.13	11.29	9.03	14.86
Atlas [38]	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 [28]	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 [29]	11.51	11.66	16.14	7.17	22.63	17.13	8.83	11.39	10.46	8.23	9.43	17.02	8.07	13.64	13.85	10.34	4.90	7.37
TPVFormer* [29]	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
OccFormer [21]	31.39	19.03	18.65	10.41	23.92	30.29	10.31	14.19	13.59	10.13	12.49	20.77	38.78	19.79	24.19	22.21	13.48	21.35
SurroundOcc [39]	31.49	20.30	**20.59**	11.68	28.06	**30.86**	10.70	15.14	**14.09**	**12.06**	14.38	22.26	37.29	23.70	24.49	22.77	14.89	21.86
GaussianFormer [26]	29.83	19.10	19.52	11.26	26.11	29.78	10.47	13.83	12.58	8.67	12.74	21.57	39.63	23.28	24.46	22.99	9.59	19.12
GaussianFormer-2 [27]	30.56	20.02	20.15	12.99	27.61	30.23	11.19	15.31	12.64	9.63	13.31	22.26	39.68	23.47	25.62	23.20	12.25	20.73
Ours	**34.06**	**21.74**	19.95	**13.63**	**28.69**	29.52	**12.69**	**16.02**	13.77	10.64	**15.75**	**23.02**	**41.74**	**26.26**	**27.52**	**26.44**	**16.51**	**25.75**

* means supervised by dense occupancy annotations as opposed to the original LiDAR segmentation labels.
The best and second-best performances are represented by **bold** and underline respectively.

F. 3D Occupancy Rendering

The final stage of our methodology is to render the learned set of refined Gaussian primitives, $\hat{\mathcal{G}}$, into the final dense semantic occupancy grid, $\mathbf{O}$. We follow the probabilistic superposition approach from GaussianFormer-2 [27], interpreting each primitive as a local probability distribution over the scene.

First, the total occupancy probability $\alpha(\mathbf{x})$ at any 3D point is calculated by aggregating the influence of all nearby Gaussians. The influence of each primitive is determined by its opacity and a Gaussian kernel, $\phi(\mathbf{x}; \hat{\mathbf{G}}_i)$, which decays with distance from its center. Assuming each contribution is an independent event, the overall probability of occupancy at point $\mathbf{x}$:

$$\alpha(\mathbf{x}) = 1 - \prod_{i=1}^{P} \big(1 - a_i \cdot \phi(\mathbf{x}; \hat{\mathbf{G}}_i)\big). \tag{10}$$

Next, the semantic prediction for an occupied point is a weighted average of each Gaussian's softmaxed semantic features $\tilde{\mathbf{c}}_i$, scaled by its opacity a_i. The weight for each Gaussian is determined by its posterior probability, denoted as $p(\mathbf{x}|\mathbf{G}_i)$, which measures its probabilistic influence at that location. The expected semantic vector $\mathbf{e}(\mathbf{x}; \hat{\mathcal{G}})$ is formulated as:

$$\mathbf{e}(\mathbf{x}; \hat{\mathcal{G}}) = \frac{\sum_{i=1}^{P} p(\mathbf{x}|\hat{\mathbf{G}}_i) a_i \tilde{\mathbf{c}}_i}{\sum_{j=1}^{P} p(\mathbf{x}|\hat{\mathbf{G}}_j) a_j}. \tag{11}$$

The final semantic occupancy vector for a given voxel, $\hat{\mathbf{o}}(\mathbf{x})$, is formed by combining the geometric and semantic predictions. The probability for the empty class is defined as $1 - \alpha(\mathbf{x})$, while the probabilities for all semantic classes are the semantic predictions $\mathbf{e}(\mathbf{x}; \hat{\mathcal{G}})$ weighted by the total occupancy probability $\alpha(\mathbf{x})$.

$$\hat{\mathbf{o}}(\mathbf{x}) = [1 - \alpha(\mathbf{x}); \alpha(\mathbf{x}) \cdot \mathbf{e}(\mathbf{x}; \hat{\mathcal{G}})]. \tag{12}$$

G. Training Objective

Our entire VG3T framework is trained as end-to-end using the total loss, L_{total}, which is a weighted sum of two primary components: a loss for the final semantic occupancy prediction, L_{occ}, and a loss for supervising the intermediate depth prediction head, L_{depth}.

$$L_{total} = \lambda_{occ} L_{occ} + \lambda_{depth} L_{depth}, \tag{13}$$

where λ_{occ} and λ_{depth} are scalar weights that balance the two losses.

Occupancy Loss. Our occupancy loss is a combination of a per-voxel cross-entropy loss (L_{ce}) and the Lovász-Softmax loss (L_{lov}). The cross-entropy loss handles the per-class classification, while the Lovász-Softmax loss is particularly effective for optimizing IoU, a crucial metric for this task. The total occupancy loss is the sum of these two terms:

$$L_{occ} = L_{ce} + L_{lov}. \tag{14}$$

Depth Loss. To supervise the depth prediction head, we employ an aleatoric uncertainty-aware loss, adapting the formulation used in DUSt3R [32] and VGGT [35]. The depth loss is defined as: $\mathcal{L}_{\text{depth}} = \sum_{i=1}^{N} (\|\Sigma_i^D \odot (\hat{D}_i - D_i)\| + \|\Sigma_i^D \odot (\nabla \hat{D}_i - \nabla D_i)\| - \alpha \log \Sigma_i^D)$, with per-pixel depth D_i and uncertainty map Σ_i^D.

IV. Experiments

A. Implementation Details

We implement our VG3T framework using PyTorch. The model's backbone, including the DINOv2 [36] patch embedding layers and the main VGGT [35] transformer blocks, is initialized from the official pre-trained VGGT weights and fine-tuned. And all other components, are trained from scratch. The model is trained for 20 epochs on the nuScenes [40] dataset with a total batch size of 4, distributed across NVIDIA L40S GPUs. For optimization, we use the AdamW optimizer with a weight decay of 0.01 [41]. To ensure stable fine-tuning, a differential learning rate is applied with a cosine annealing schedule: the pre-trained VGGT backbone is optimized with a learning rate of 1e-5, while our newly introduced components are trained with a learning rate of 1e-4. During both training and inference, input images are resized to a resolution of 294×518. We apply Grid-Based sampling with a 0.5 m cell size to remove redundant primitives.

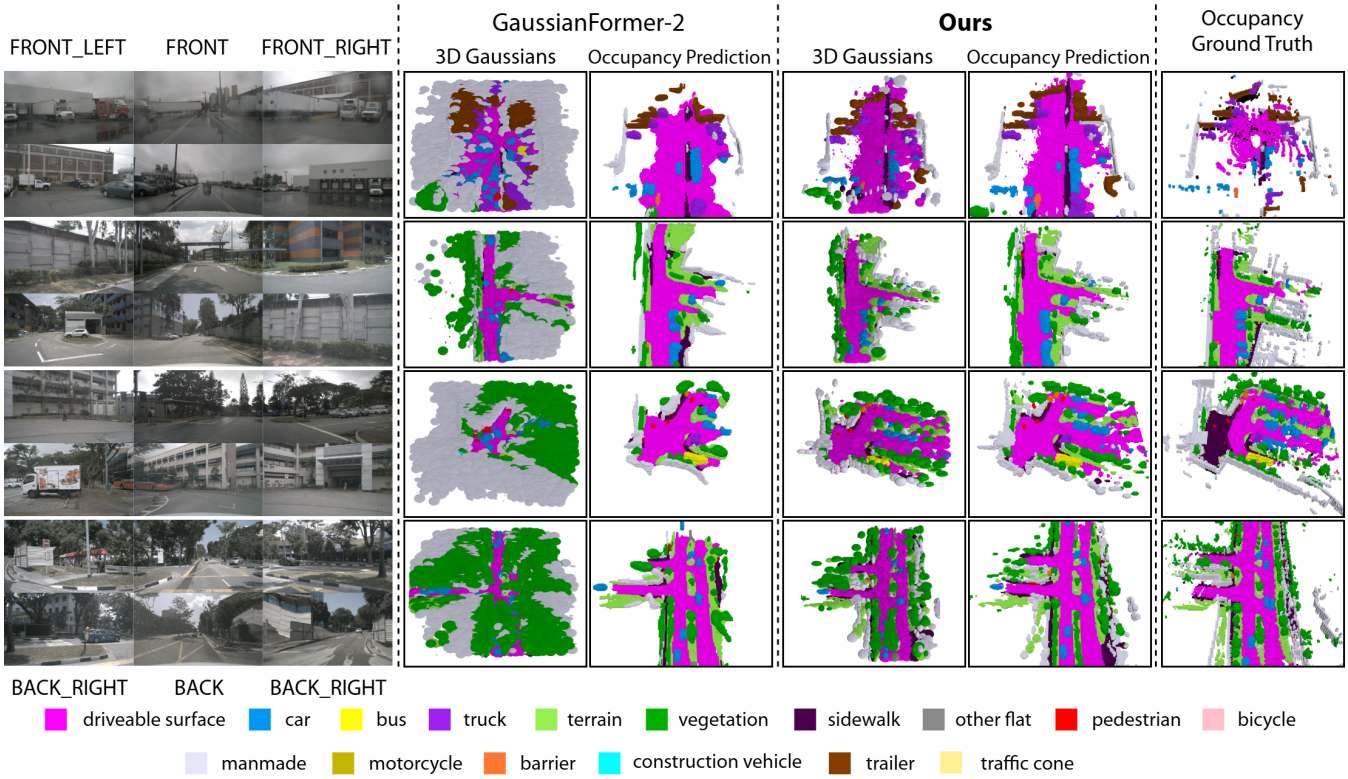

Fig. 3: Qualitative results of our method and GaussianFormer-2 on SurroundOcc dataset.

TABLE II: Initializing Methods Comparison with GaussianFormer.

Method	Initialization	Position	
		Perc. (%) ↑	Dist. (m) ↓
GaussianFormer [26]	Random	16.41	3.07
GaussianFormer-2 [27]	Single-View	28.85	1.24
VG3T (ours)	Multi-View	**51.22**	**0.97**

TABLE III: Efficiency Comparison with GaussianFormer.

Method	Number of Gaussians ↓	Latency (ms) ↓	Memory (MB) ↓	mIoU	IoU	RayIoU
GaussianFormer [26]	144000	372	6229	19.10	29.83	25.25
GaussianFormer-2 [27]	25600	513	3063	20.02	30.56	29.34
VG3T (ours)	**13661***	**223**	**1975**	**21.74**	**34.06**	**32.66**

* denotes average number of Gaussians per scene.

B. Evaluation Benchmarks

All experiments are conducted on the nuScenes [40] dataset, a large-scale benchmark for autonomous driving. This dataset consists of 1,000 urban driving sequences, with a standard training, validation, and testing split of 700/150/150. We use the dense semantic occupancy annotations from the SurroundOcc [39] benchmark for supervision and evaluation. The ground truth discretizes the scene into a $200 \times 200 \times 16$ voxel grid, covering a spatial volume of $[-50m, 50m]$ on the X/Y axes and $[-5m, 3m]$ on the Z axis. This results in a voxel resolution of $0.5m$. Each voxel is labeled as one of 18 classes: 16 semantic categories, an empty class, and an unknown class.

We evaluate our method using two primary metrics, mean Intersection-over-Union (mIoU) and the standard Intersection-over-Union (IoU). To provide a more comprehensive assessment of scene completion capabilities, we also report Ray-Based IoU (RayIoU) [25] as a supplementary metric. RayIoU specifically evaluates the quality of predictions along rays cast from the camera's viewpoint, offering a robust measure of geometric completion.

C. Experimental Results

We evaluate VG3T on the nuScenes [40] validation set and compare its performance against state-of-the-art methods in (Table I). Our proposed model sets a new state-of-the-art. This represents a significant improvement of 1.4%p mIoU over the previous best-performing methods like SurroundOcc [39] and improvement of 1.7%p over GaussianFormer-2 [27]. VG3T demonstrates a clear superiority in modeling static geometry, achieving the highest scores across all background classes. Furthermore, our model shows exceptional performance on small and dynamic object classes, such as bicycle, motorcycle, bus, and trailer, that are traditionally challenging for vision-based methods.

D. Ablation Studies

We conduct a detailed comparison with the leading Gaussian-based methods in 3D occupancy prediction, GaussianFormer [26] and GaussianFormer-2 [27], as shown in Tables II and III.

Initializing Methods Comparison. Table II demonstrates the effectiveness of our multi-view initialization strategy. We

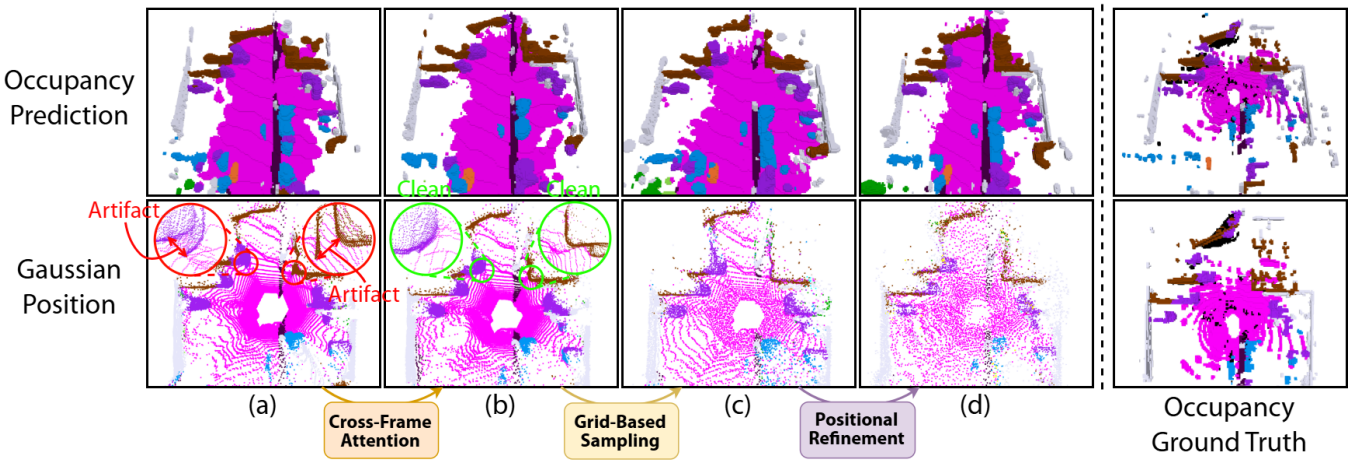

Fig. 4: Qualitative results of model component design.

TABLE IV: Ablation Study on effectiveness of each proposed component.

Multi-View Initialization	Grid Sampling	Refinement	Number of Gaussians ↓	Latency (ms) ↓	mIoU	IoU	RayIoU
×	×	×	64129*	456	21.38	33.42	32.10
✓	×	×	64074*	453	21.48	33.83	32.44
✓	✓	×	14082*	437	21.20	33.65	31.99
✓	✓	✓	**13661***	**443**	**21.74**	**34.06**	**32.66**

* denotes average number of Gaussians per scene.

evaluate initialization quality using two metrics, the percentage of Gaussians positioned in occupied space (Perc.) and the average distance from each Gaussian to its nearest occupied voxel center (Dist.). Our multi-view initialization achieves 77% improvement in positioning accuracy and 22% reduction in distance compared to the baseline GaussianFormer-2 [27]. This better initialization directly translates to better occupancy prediction performance.

Efficiency Comparison. Table III compares VG3T with GaussianFormer [26] and GaussianFormer-2 [27] in terms of accuracy and efficiency. Compared with GaussianFormer, VG3T improves mIoU 2.64%p and IoU 4.23%p, while reducing the number of Gaussians about 90% fewer and using substantially less memory. Compared with GaussianFormer-2, VG3T achieves higher mIoU 1.72%p and higher IoU 3.50%p, while having low latency and more memory efficieny. VG3T also improves RayIoU, indicating stronger ray-level geometric consistency. Overall, the results show that VG3T achieves better accuracy with a significantly more compact and efficient Gaussian representation.

Model Component Design. We conduct an ablation study to validate the effectiveness of our key components, with results in Table IV and Figure 4. Starting from a minimum baseline that processes views independently, Multi-View Initialization improves geometric consistency and boosts both mIoU and RayIoU. Adding Grid-Based Sampling reduces the number of Gaussians by 78%, yielding a more compact representation with only a minor performance drop. Finally, Positional Refinement recovers this loss by correcting the positions of the remaining primitives, producing a more uniform and geometrically aligned distribution. Together, these components achieve the best performance with the most efficient representation.

E. Qualitative Results

We provide qualitative visualizations in Figure 3 to illustrate the effectiveness of our approach and compare it against the baseline, GaussianFormer-2 [27]. The results demonstrate VG3T's ability to generate comprehensive and geometrically accurate occupancy predictions. GaussianFormer-2's single-view distribution-based method allocates a significant portion of its primitives to unoccupied space, which reduces its capacity to model fine geometric details accurately, as shown in (Table II). In contrast, our multi-view direct initialization concentrates the model's entire representational capacity on geometrically relevant regions from the very beginning. Consequently, VG3T achieves superior geometric coverage and detail while leveraging fewer Gaussians than GaussianFormer-2. VG3T is able to capture the sharp, planar surfaces of buildings and other structures with significantly less noise and greater detail compared to the often over-smoothed or incomplete predictions from GaussianFormer-2.

V. CONCLUSIONS

This work tackles two fundamental challenges in 3D semantic occupancy prediction, the lack of multi-view correlation from monocular 2D image features leading to fragmented 3D representations and the density bias inherent in pixel-aligned 3D Gaussian initialization. Our model, VG3T, was designed as a unified solution to both. By employing a multi-view transformer backbone, VG3T effectively exploits cross-view information to initialize all 3D Gaussians in a single, feed-forward pass, enabling end-to-end training. To address the density bias, we introduced a Grid-Based Sampling and Positional Refinement module that manages primitive density. As a result, VG3T, achieves new state-of-the-art performance on the nuScenes benchmark for 3D semantic occupancy prediction.

ACKNOWLEDGMENT

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2025-02219317, AI Star Fellowship, Kookmin University). This work also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2026-25497410).

REFERENCES

[1] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, and et al., "Goal-Oriented Autonomous Driving," arXiv preprint arXiv:2212.10156, 2022.

[2] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li, "Bevdepth: Acquisition of reliable depth for multi-view 3d object detection," 2022. [Online]. Available: https://arxiv.org/abs/2206.10092

[3] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-View 3D Object Detection Network for Autonomous Driving," in *CVPR*, 2017, pp. 1907–1915.

[4] J. Philion and S. Fidler, "Lift, Splat, Shoot: Encoding Images from Arbitrary Camera Rigs by Implicitly Unprojecting to 3D," in *ECCV*, 2020.

[5] S. Woo, M. Park, Y. Lee, S. Lee, and E. Kim, "Location-aware transformer network for bird's eye view semantic segmentation," *IEEE Transactions on Intelligent Vehicles*, 2024.

[6] J. Kim, J. Kim, J. Yoo, D. Kim, and N. Kwak, "Vehicle image generation going well with the surroundings," in *International Conference on Neural Information Processing*. Springer, 2021, pp. 63–74.

[7] X. Timoneda, M. Herb, F. Duerr, and D. Goehring, "Selfoccflow: Towards end-to-end self-supervised 3d occupancy flow prediction," *IEEE Robotics and Automation Letters*, vol. 11, no. 4, p. 4331–4338, Apr. 2026. [Online]. Available: http://dx.doi.org/10.1109/LRA.2026.3665447

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," 2017. [Online]. Available: https://arxiv.org/abs/1706.02413

[9] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017. [Online]. Available: https://arxiv.org/abs/1612.00593

[10] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," 2019. [Online]. Available: https://arxiv.org/abs/1812.05784

[11] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *CVPR*, 2018, pp. 4490–4499.

[12] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang, "BEVFusion: A Simple and Robust Lidar-Camera Fusion Framework," *NIPS*, vol. 35, pp. 10 421–10 434, 2022.

[13] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation," in *ICRA*, 2023, pp. 2774–2781.

[14] D. Ye, Z. Zhou, W. Chen, Y. Xie, Y. Wang, P. Wang, and H. Foroosh, "LidarMultiNet: Towards a Unified Multi-Task Network for Lidar Perception," in *AAAI*, 2023, pp. 3231–3240.

[15] X. Timoneda, M. Herb, F. Duerr, D. Goehring, and F. Yu, "Multimodal nerf self-supervision for lidar semantic segmentation," 2024. [Online]. Available: https://arxiv.org/abs/2411.02969

[16] A.-Q. Cao and R. de Charette, "Monoscene: Monocular 3D Semantic Scene Completion," in *CVPR*, 2022, pp. 3991–4001.

[17] H. Jiang, T. Cheng, N. Gao, H. Zhang, W. Liu, and X. Wang, "Symphonize 3D Semantic Scene Completion with Contextual Instance Queries," arXiv preprint arXiv:2306.15670, 2023.

[18] Y. Li, Z. Yu, C. Choy, C. Xiao, J. M. Alvarez, S. Fidler, C. Feng, and A. Anandkumar, "VoxFormer: Sparse Voxel Transformer for Camera-Based 3D Semantic Scene Completion," in *CVPR*, 2023, pp. 9087–9098.

[19] Z. Li, Z. Yu, D. Austin, M. Fang, S. Lan, J. Kautz, and J. M. Alvarez, "FB-Occ: 3D Occupancy Prediction Based on Forward-Backward View Transformation," arXiv preprint arXiv:2307.01492, 2023.

[20] Y. Huang, W. Zheng, B. Zhang, J. Zhou, and J. Lu, "SelfOcc: Self-Supervised Vision-Based 3D Occupancy Prediction," in *CVPR*, 2024.

[21] Y. Zhang, Z. Zhu, and D. Du, "OccFormer: Dual-Path Transformer for Vision-Based 3D Semantic Occupancy Prediction," arXiv preprint arXiv:2304.05316, 2023.

[22] Z. Yu, C. Shu, J. Deng, K. Lu, Z. Liu, J. Yu, D. Yang, H. Li, and Y. Chen, "FlashOcc: Fast and Memory-Efficient Occupancy Prediction via Channel-to-Height Plugin," arXiv preprint arXiv:2311.12058, 2023.

[23] J. Shin, Y. Kim, S. Hong, and J. Lee, "Learning dual hierarchical representation for 3d surface reconstruction," in *Proceedings of the Asian Conference on Computer Vision*, 2024, pp. 4422–4438.

[24] Y. Lu, X. Zhu, T. Wang, and Y. Ma, "OctreeOcc: Efficient and Multi-Granularity Occupancy Prediction Using Octree Queries," arXiv preprint arXiv:2312.03774, 2023.

[25] P. Tang, Z. Wang, G. Wang, J. Zheng, X. Ren, B. Feng, and C. Ma, "SparseOcc: Rethinking Sparse Latent Representation for Vision-Based Semantic Occupancy Prediction," in *CVPR*, 2024, pp. 15 035–15 044.

[26] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction," *arXiv preprint arXiv:2405.17429*, 2024.

[27] Y. Huang, A. Thammatadatrakoon, W. Zheng, Y. Zhang, D. Du, and J. Lu, "Gaussianformer-2: Probabilistic gaussian superposition for efficient 3d occupancy prediction," *arXiv preprint arXiv:2412.04384*, 2024.

[28] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers," arXiv preprint arXiv:2203.17270, 2022.

[29] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-Perspective View for Vision-Based 3D Semantic Occupancy Prediction," in *CVPR*, 2023, pp. 9223–9232.

[30] Y. Shi, T. Cheng, Q. Zhang, W. Liu, and X. Wang, "Occupancy as Set of Points," in *ECCV*, 2024.

[31] J. Wang, Z. Liu, Q. Meng, L. Yan, K. Wang, J. Yang, W. Liu, Q. Hou, and M. Cheng, "Opus: Occupancy Prediction Using a Sparse Set," in *NIPS*, 2024.

[32] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," 2024. [Online]. Available: https://arxiv.org/abs/2312.14132

[33] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," 2024. [Online]. Available: https://arxiv.org/abs/2406.09756

[34] J. Yang, A. Sax, K. J. Liang, M. Henaff, H. Tang, A. Cao, J. Chai, F. Meier, and M. Feiszli, "Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025.

[35] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," 2025. [Online]. Available: https://arxiv.org/abs/2503.11651

[36] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.

[37] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," 2021. [Online]. Available: https://arxiv.org/abs/2103.13413

[38] Z. Murez, T. V. As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich, "Atlas: End-to-End 3D Scene Reconstruction from Posed Images," in *ECCV*, 2020, pp. 414–431.

[39] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, J. Zhou, and J. Lu, "SurroundOcc: Multi-Camera 3D Occupancy Prediction for Autonomous Driving," in *ICCV*, 2023, pp. 21 729–21 740.

[40] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *CVPR*, 2020.

[41] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," arXiv preprint arXiv:1711.05101, 2017.