EasyOcc: 3D Pseudo-Label Supervision for Fully Self-Supervised Semantic Occupancy Prediction Models

Seamie Hayes, Ganesh Sistu, Ciaran Eising

***Abstract*— Self-supervised models have recently achieved notable advancements, particularly in the domain of semantic occupancy prediction. These models utilize sophisticated loss computation strategies to compensate for the absence of ground-truth labels. For instance, techniques such as novel view synthesis, cross-view rendering, and depth estimation have been explored to address the issue of semantic and depth ambiguity. However, such techniques typically incur high computational costs and memory usage during the training stage, especially in the case of novel view synthesis. To mitigate these issues, we propose 3D pseudo-ground-truth labels generated by the foundation models Grounded-SAM and Metric3Dv2, and harness temporal information for label densification. Our 3D pseudo-labels can be easily integrated into existing models, which yields substantial performance improvements, with mIoU increasing by 45%, from 9.73 to 14.09, when implemented into the OccNeRF model. This stands in contrast to earlier advancements in the field, which are often not readily transferable to other architectures. Additionally, we propose a streamlined model, EasyOcc, achieving 13.86 mIoU. This model conducts learning solely from our labels, avoiding complex rendering strategies mentioned previously. Furthermore, our method enables models to attain state-of-the-art performance when evaluated on the full scene without applying the camera mask, with EasyOcc achieving 7.71 mIoU, outperforming the previous best model by 31%. These findings highlight the critical importance of foundation models, temporal context, and the choice of loss computation space in self-supervised learning for comprehensive scene understanding.**

I. Introduction

Recent progress in machine learning has resulted in highly capable models for tackling complex tasks, particularly Vision Language Models (VLM) such as Grounding DINO [1], and Visual Foundation Models (VFM) including Metric3Dv2 [2] and Segment Anything (SAM) [3]. Trained on large datasets with hundreds of millions of parameters, these models enable capabilities such as zero-shot depth estimation and semantic segmentation via Metric3Dv2 and Grounded-SAM [1], [3], [4] respectively. Each model holds promise for autonomous perception, especially in self-supervised semantic occupancy prediction, which involves assigning semantic labels to voxels in a discretized 3D grid without manually annotated ground-truth labels for supervision. The lack of ground-truth labels leads to the issue of semantic and depth ambiguity, which foundation models help address. Existing methods leverage foundation models to generate 2D pseudo-labels for supervision: SelfOcc employs OpenSeeD [5] for semantic supervision, while both OccNeRF and GaussianOcc utilize Grounded-SAM. GaussTR [6] integrates CLIP [7] for feature supervision and Metric3Dv2 for depth supervision. More recently, the generation of 3D pseudo-labels has been studied [8], [9], and our work aims to advance this research further by integrating 3D pseudo-labels into existing models and striving for holistic scene representation.

Despite the deployment of such strong foundation models, a key challenge that persists in this domain is the complexity of training strategies. SelfOcc uses novel view synthesis and a multi-layer perceptron (MLP) to predict signed distance functions for occupancy and semantics. OccNeRF applies NeRF-based volume rendering [10] for depth and semantics, while GaussianOcc uses a two-stage process: estimating transformations between subsequent sample coordinate frames, and then applying Gaussian Splatting [11] for rendering. These methods reflect a trend toward extracting maximal information from camera images using increasingly complex techniques. For these models, all loss is computed in 2D camera space, as pseudo-ground-truth labels are generated from the camera images. Representing these labels in 3D eliminates the need for the costly 2D rendering of the scene, allowing direct scene representation and loss computation in 3D.

We propose a method that uses Grounded-SAM and Metric3Dv2 to generate 3D pseudo-labels for direct loss computation in 3D space, which easily allows the model to understand semantics and spatial geometry in one loss function, pseudo-loss. To increase label density, we aggregate temporal samples, limited to static objects to avoid duplicating dynamic objects, a known issue in self-supervised [12]–[14] and weakly supervised models [15], [16]. Prior work [17]–[19] has shown the value of temporal information for improving performance. Aggregating temporal information is typically performed at inference time via the aggregation features, thereby introducing additional computational overhead during model deployment. Our method avoids this issue. In summary, 3D pseudo-labels provide several important advantages. Firstly, they remove the requirement for novel view synthesis and depth estimation during training, thereby

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

Seamie Hayes and Ciarán Eising are with the Department of Electronic and Computer Engineering, the Research Ireland Centre for Research Training in Foundations in Data Science, and the Data Driven Computer Engineering (D²iCE) Research Centre, all hosted in the University of Limerick, Limerick, V94 T9PX Ireland.

Ganesh Sistu is with the Department of Electronic and Computer Engineering, and the Data Driven Computer Engineering (D²iCE) Research Centre, University of Limerick, Limerick, V94 T9PX, Ireland.

Corresponding author: Seamie Hayes (e-mail: seamie.hayes@ul.ie)

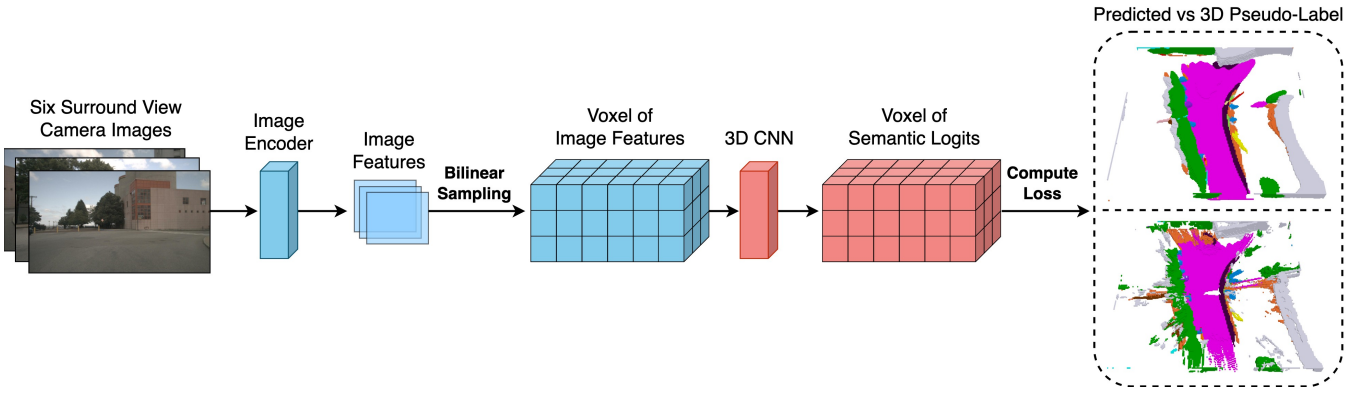

Fig. 1: **Overview of the EasyOcc Model Architecture:** Image features are first extracted from the six surrounding camera views using a 2D image encoder. These features are then projected into a 3D feature volume via bilinear sampling. The resulting voxel grid is processed by a 3D CNN to generate the final semantic occupancy prediction. Loss is computed by comparing the predicted output (**top**) with the corresponding 3D pseudo-labels (**bottom**).

lowering computational demands. Secondly, they enable efficient aggregation of temporal information, which plays a crucial role in spatial understanding. Lastly, they allow models to develop a more comprehensive understanding of the scene, leading to improved performance not only in regions visible to the camera but, more importantly, in areas occluded from the current viewpoint.

Furthermore, our 3D pseudo-labels can be easily added to existing models as an auxiliary loss to boost performance. We explore the integration of these labels in three previous models: SelfOcc, OccNeRF, and GaussianOcc. As recent advances often develop in isolation, we present a complementary method to enhance model compatibility and generalization. Additionally, we introduce EasyOcc, a streamlined model that solely uses our generated 3D pseudo-labels for loss computation, demonstrating that complex rendering techniques are not necessary for noteworthy model performance. Unlike earlier self-supervised methods, our approach aligns more with supervised frameworks [20], [21], with a simpler architecture, which is illustrated in Figure 1. EasyOcc requires no LiDAR supervision or deployment of foundation models at inference.

In summary, our main contributions are as follows:

- **3D Pseudo-Labels:** We introduce an approach that leverages Grounded-SAM and Metric3Dv2 to generate 3D pseudo-labels for loss computation directly in 3D space.
- **Seamless Integration:** Our labels can be effortlessly integrated into existing models via an auxiliary loss function, yielding improvements of 43% in mIoU.
- **Segmentation of Dynamic Classes:** Accurate segmentation of dynamic classes is critical for autonomous perception. In the case of SelfOcc, incorporating our 3D pseudo-labels improves prediction segmentation performance by 627%.
- **Holistic Scene Representation:** The proposed 3D pseudo-labels enable a more comprehensive representation of the scene, resulting in a 219% increase in mIoU for OccNeRF when evaluated across the entire voxel grid.

This paper is structured as follows: Section II reviews prior work on semantic occupancy prediction models and pseudo-labels. Section III details the generation of our 3D pseudo-labels and provides an analysis of their quality. Following this, Section IV details the EasyOcc model and 3D pseudo-label integration into existing architectures. Section V compares models incorporating our 3D pseudo-labels with state-of-the-art (SOTA) models, highlighting quantitative and qualitative improvements through ablation studies. Finally, Section VI presents concluding remarks.

II. Literature Review

This section is structured as follows: Subsection II-A reviews semantic occupancy prediction models and analyzes the three models selected for modification, along with other SOTA models we shall compare against. Subsection II-B discusses pseudo-labels used in this space.

A. Semantic Occupancy Prediction

In autonomous perception, Bird's Eye View (BEV) methods have historically been dominant due to their simple yet effective scene representation [22]–[25]. Recently, semantic occupancy prediction has gained attention, driven by benchmark datasets [26]–[29] with accurate annotations, generated from manually labeled nuScenes LiDAR data [30]. This shift led to the creation of notable supervised semantic occupancy prediction models, with improvements from techniques such as Gaussian Splatting, multi-modal fusion, and object deduplication [11], [15], [16], [20], [21]. Following this, self-supervised counterparts of these models emerged, particularly due to their flexibility in training strategy. In this study, we modify three self-supervised models: SelfOcc, OccNeRF, and GaussianOcc, with each model's pipeline discussed in this section, along with other SOTA models that we will be comparing against.

SelfOcc employs an RN-50 image encoder followed by a 3D encoder to generate a 3D scene representation, inspired

by BEVFormer [24]. An MLP then predicts signed distance field (SDF) values, color, and semantic features from the 3D volume, for rendering depth, color, and semantics [12]. Rendered depth supports multi-frame photometric consistency, rendered color is compared to the camera image, while semantics are compared against 2D pseudo-labels from OpenSeeD. Semantics and occupancy are both computed via the SDF, with both contributing to the final scene representation. In our implementation, pseudo-loss is applied exclusively to the semantic voxel, with the occupancy voxel excluded from both the pseudo-loss computation and the final scene representation, as detailed in Subsection IV-B. Our loss computation technique aligns with that of supervised models, as both compare a predicted voxel grid against voxel labels in 3D space.

OccNeRF follows a similar pipeline, using an RN-101 for feature extraction and bilinear sampling to project image features into 3D voxel space. These features are refined using a 3D CNN [13]. Bilinear sampling outperforms depth-based splatting methods seen in Lift-Splat-Shoot [23], though multi-scale deformable attention provides superior performance at higher computational cost [22]. Following the 3D CNN, NeRF rendering is employed to render both depth and semantic information, with depth supporting multi-frame photometric consistency, while semantics are compared against pseudo-labels from Grounded-SAM [1], [3], [4]. Similar to SelfOcc, pseudo-loss is computed on the predicted 3D voxel of semantic logits.

GaussianOcc builds on OccNeRF with Gaussian rasterization [11] for rendering both semantics and depth [14]. For use in multi-frame consistency, it estimates pose transformations using a 6D pose network instead of ground-truth poses, which is more effective, given the nuScenes dataset's lack of z-axis translation in ego-vehicle transformations [30]. Pseudo-loss is computed in the same manner as in OccNeRF.

GaussTR represents the scene using 3D Gaussians, which are refined through self-attention and image cross-attention mechanisms [6]. The model utilizes CLIP and Metric3Dv2 for pseudo-supervision. Similarly, **TT-OccCamera** also models the scene with 3D Gaussians, however, it is a test-time compute method, which eliminates the need for pre-training. **GaussianFlowOcc** refines Gaussians using image cross-attention, induced self-attention, and induced temporal attention in relation to previous Gaussians [31]. Additionally, it incorporates temporal 2D pseudo-labels to enhance performance. The model further leverages Grounded-SAM and Metric3Dv2 for pseudo-supervision.

AutoOcc addresses semantic occupancy prediction using a VLM-only framework [9]. It begins by computing attention maps from surround-view images and candidate object categories [32]–[34]. These attention maps are then input into VLMs to extract semantic and depth features [35]–[37], which are subsequently projected into 3D space. The label refinement process involves flow estimation and smoothing of instance-level predictions using 3D Gaussian Splatting. AutoOcc presents their work as both a prediction paradigm but also a pseudo-labelling method. In contrast, our approach investigates the integration of our 3D pseudo-labels into additional models, enabling distillation into significantly smaller architectures. AutoOcc does not explore this direction.

Our proposed model, **EasyOcc**, shares a similar pipeline with GaussianOcc. However, it omits several components: there is no pose estimation, novel view synthesis using Gaussian Splatting, or multi-frame photometric consistency. Instead, EasyOcc uses a single loss function, pseudo-loss, comparing voxel predictions directly with our generated 3D pseudo-labels. Despite its simplified design, EasyOcc outperforms other models that employ complex techniques in the mIoU metric, as detailed in the results Section V.

We did not pursue implementing 3D pseudo-labels in GaussTR, as its Gaussian-based scene representation would require significant adjustments to maintain practical training times due to the inefficiencies of the Gaussian-to-voxel splatting module. TT-OccCamera, GaussianFlowOcc, and AutoOcc are currently not open-source and hence omitted for modification.

B. Pseudo-Labels

The use of 2D pseudo-labels in self-supervised semantic occupancy models has been extensively studied, particularly through the application of VLMs [4], [7], [38] and VFMs [2], [35]. These models have demonstrated significant utility across various domains, including medical applications [39] and other autonomous perception tasks [40]. They play a crucial role in addressing the challenge of missing ground-truth labels, particularly in resolving semantic and depth ambiguities. VLMs help mitigate semantic ambiguity by leveraging both spatial and linguistic cues to produce pixel-level semantic maps. Depth ambiguity, while partially addressed using multi-frame photometric consistency, is significantly reduced through the use of metric depth VFMs [2], [35], [41], which provide accurate pixel-level depth maps, shown to increase model performance substantially [6], [31]. Semantic maps and depth maps serve as supervision signals against renders of the semantic voxel grid. Nonetheless, a noticeable performance gap persists between supervised and self-supervised models, indicating a need for a more nuanced integration of foundation models, potentially by focusing on the dimensionality of the labels.

More recently, the use of 3D pseudo-labels has gained attention, which aligns with the space in which final predictions are made. A notable example is AGO [8], which combines Grounded-SAM [4] with LiDAR point cloud data, utilizing multi-frame aggregation, point cloud ray casting, and semantic voting to generate richer labels for training. However, this approach necessitates equipping the vehicle with a LiDAR sensor, which introduces significant cost and complexity, including the need for careful synchronization with the camera system. Additionally, the output generated by the aforementioned AutoOcc can be considered a form of 3D pseudo-labels.

Our approach to 3D pseudo-label generation does not rely on LiDAR data and instead combines Metric3Dv2 [2] with

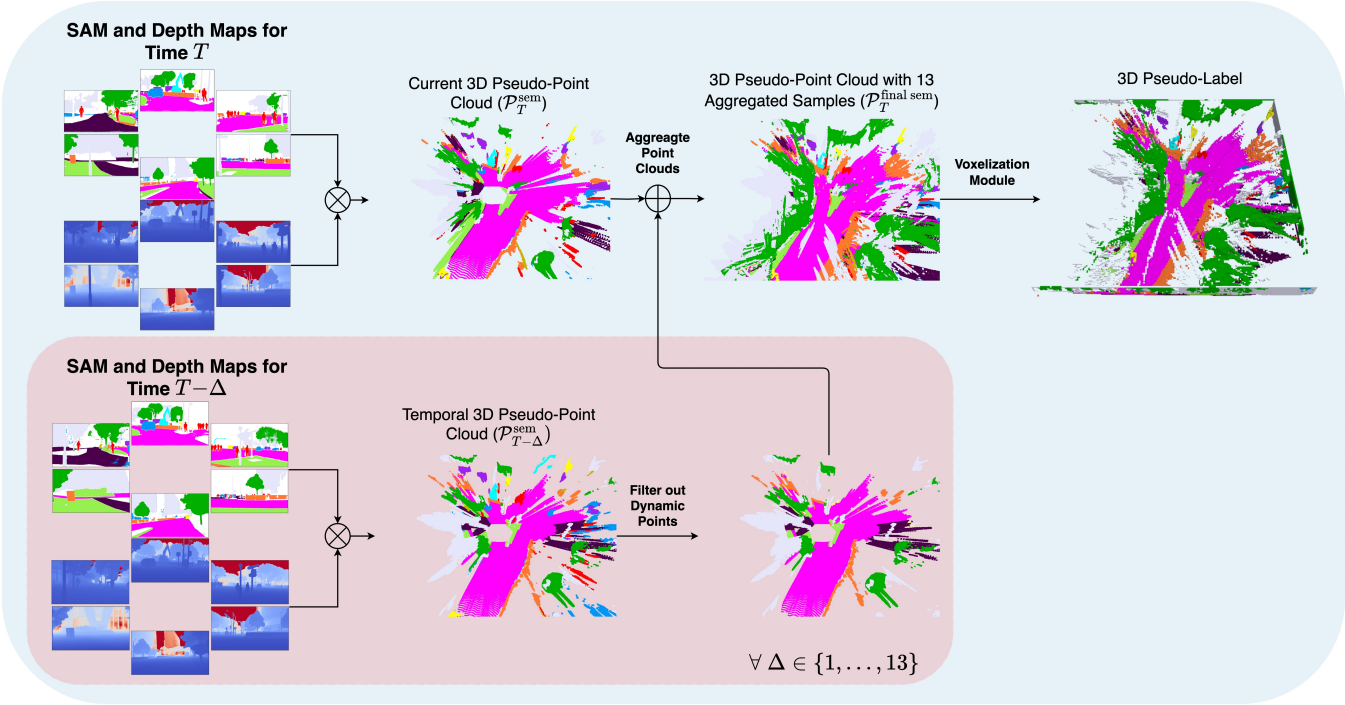

Fig. 2: **Our Method of Generating 3D Pseudo-Labels:** Grounded-SAM labels are first projected into 3D space using Metric3Dv2 depth maps and camera pose information to produce a semantic point cloud. To densify the point cloud, we aggregate 13 temporal samples while filtering out dynamic objects to avoid duplication. The resulting densified point cloud is then passed to a voxelization module to generate 3D pseudo-voxel labels.

Grounded-SAM [1], [3], [4] to produce 3D pseudo-ground-truth labels. Metric3Dv2 is selected for its reliable depth estimation across different camera views, while Grounded-SAM is utilized for its high-quality semantic segmentation. The generated labels undergo further refinement through techniques such as outlier removal, occupancy thresholding, and temporal aggregation. These 3D pseudo-labels enable models to learn more effective scene representations, particularly when evaluated over the full voxel grid, a setting that has not been thoroughly investigated in prior work.

III. Methodology I: 3D Pseudo-Labels

This section begins with the generation process for our 3D pseudo-labels in Subsection III-A. Following this, subsection III-B evaluates the quality of our 3D pseudo-labels when aggregating multiple samples and altering the occupancy threshold value. A comparison against ground truth data is also provided.

A. Generation of 3D Pseudo-Labels

This section presents a key contribution of this paper: the generation of 3D pseudo-labels from semantic maps of Grounded-SAM and depth maps from Metric3Dv2. This method enables loss computation directly in 3D voxel space, eliminating the need for view synthesis and aligning our approach with supervised training pipelines. The process is illustrated in Figure 2 and divided into three steps: semantic point cloud generation (Subsubsection III-A.1), densification (Subsubsection III-A.2), and voxelization (Subsubsection III-A.3).

1) Semantic Point Cloud Generation: This stage will detail the generation of a semantic point cloud for an arbitrary sample. Semantic maps are sourced from the OccNeRF repository, which are generated using Grounded-SAM, while depth maps are generated with the Giant variant of Metric3Dv2 for optimal training performance [2]. For each camera, $i \in \{1, \ldots, 6\}$, in a sample, given the corresponding semantic map $S_i \in \mathbb{R}^{H \times W}$, depth map $D_i \in \mathbb{R}^{H \times W}$, camera intrinsic matrix $K_i \in \mathbb{R}^{3\times3}$, and camera-to-global transformation $\mathbf{T}^{\text{global}}_{\text{camera},\,i} \in \mathbb{R}^{4\times4}$, each arbitraty pixel $(u, v) \in [0, W) \times [0, H)$ is projected into the dehomogenised 3D global coordinates in Equation (1):

$$\mathbf{P}^{(u,v)}_{\text{global},\,i} = \mathbf{T}^{\text{global}}_{\text{camera},\,i} \begin{bmatrix} D_i(u,v) \cdot K_i^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \\ 1 \end{bmatrix} \tag{1}$$

Following this, each projected pixel, $\mathbf{P}^{(u,v)}_{\text{global},i}$, is then decorated with its corresponding semantic pixel $S_i(u, v)$, to yield a semantic point cloud, $\mathcal{P}_i^{\text{sem}}$, seen in Equation (2), where $\mathcal{L} = \{0, 1, \ldots, 17\}$ denotes the semantic label space.

$$\mathcal{P}_i^{\text{sem}} = \left\{ \left(\mathbf{P}^{(u,v)}_{\text{global},\,i},\ S_i(u,v) \right) \right\}, \quad \mathcal{P}_i^{\text{sem}} \subset \mathbb{R}^3 \times \mathcal{L} \tag{2}$$

Finally, we aggregate the semantic point cloud for each camera, $\mathcal{P}_i^{\text{sem}}$, into a unified semantic point cloud, $\mathcal{P}^{\text{sem}}$,

expressed in Equation (3).

$$\mathcal{P}^{\text{sem}} = \bigcup_{i=1}^{N} \{\mathcal{P}_i^{\text{sem}}\} \tag{3}$$

To improve spatial accuracy, outlier removal is performed on $\mathcal{P}^{\text{sem}}$, yielding a consolidated semantic point cloud in the global coordinate frame. Outlier removal is implemented in the Open3D library [42], and is commonly used for preprocessing in 3D data analysis.

2) Semantic Point Cloud Densification: Following the previous step, point clouds for each sample, $\mathcal{P}_T^{\text{sem}}$, are densified using temporal semantic point clouds, $\mathcal{P}_{T-\Delta}^{\text{sem}}$, for all $\Delta \in \{1, \ldots, 13\}$. First, we remove dynamic points (e.g., vehicles and pedestrians) in $\mathcal{P}_{T-\Delta}^{\text{sem}}$, to prevent object duplication, while retaining static points, such as the sidewalk and drivable surface. Following this, we transform the unions of $\mathcal{P}_T^{\text{sem}}$ and $\mathcal{P}_{T-\Delta}^{\text{sem}}$ from global coordinates to ego-vehicle coordinates of time T with $\mathbf{T}_{\text{global}}^{\text{ego}} \in \mathbb{R}^{4\times4}$, which yields the densified semantic point cloud, $\mathcal{P}_T^{\text{final sem}}$, in Equation (4):

$$\mathcal{P}_T^{\text{final sem}} = \mathbf{T}_{\text{global}}^{\text{ego}} \circ \left(\mathcal{P}_T^{\text{sem}} \cup \bigcup_{\Delta=1}^{13} \mathcal{P}_{T-\Delta}^{\text{sem}} \right) \tag{4}$$

This process ensures that the final voxelisation of $\mathcal{P}_T^{\text{final sem}}$ yields labels more closely resembling ground-truth data. The effectiveness of this step is demonstrated in the following Subsection III-B.1 and in the ablation study on the EasyOcc model in Subsection V-E.2.

3) Semantic Point Cloud Voxelization: Once $\mathcal{P}_T^{\text{sem}}$ is obtained, it is voxelized according to the voxel bounds defined by the Occ3D ground truth: $[-40\text{m}, -40\text{m}, -1\text{m}, 40\text{m}, 40\text{m}, 5.4\text{m}]$, using a voxel resolution of 0.4m^3, expressed in the ego-frame coordinate system of the current sample [27]. Given the high density of $\mathcal{P}_T^{\text{sem}}$ due to the aggregation of many temporal samples, a voxel is considered occupied only if it contains a minimum of ten points; otherwise, it is treated as empty. This threshold helps mitigate the influence of stray points that could otherwise result in erroneous voxelization. For voxels classified as occupied, the semantic label is assigned based on the majority class among the contained points. The final output is a 3D pseudo-label designed for use in self-supervised semantic occupancy prediction models, specifically adapted for the Occ3D dataset.

B. 3D Pseudo-Label Quality

In this section, we compare our generated 3D pseudo-labels with the ground-truth labels from Occ3D, both quantitatively and qualitatively.

1) Quantitative Analysis: Here, we examine the effect that aggregating temporal samples and thresholding occupancy has on likeness between our labels and the ground-truth labels in regards to the mIoU metric.

In Figure 3, we compare 3D pseudo-labels generated using varying numbers of aggregated temporal samples against the Occ3D ground-truth labels. The result follows a logarithmic trend, indicating saturation, where aggregating more temporal samples provides diminishing returns in mIoU. The optimal number of temporal samples is found to be 13, at which point we achieve the highest mIoU score of 13.58. These findings show that incorporating temporal samples improves the similarity of the pseudo-labels to the Occ3D ground truth. However, as the number of temporal samples increases, fewer instances can fully utilize them, for example, the 13$^{\text{th}}$ sample has only 12 predecessors. The maximum number of temporal samples is capped at 13 due to memory constraints.

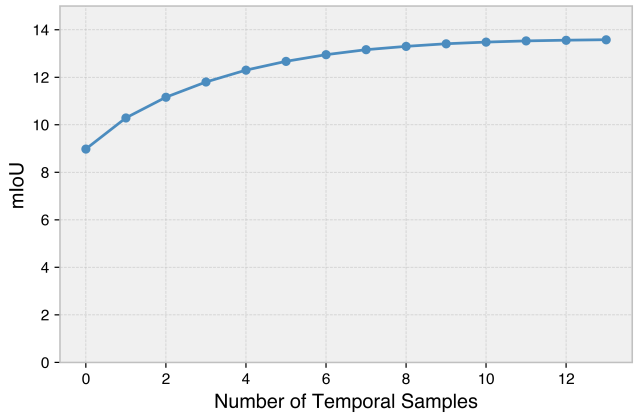

Fig. 3: **Temporal Sample Aggregation Analysis:** 3D pseudo-labels directly compared to ground-truth labels for various numbers of aggregated samples. A camera mask is applied to align with the model evaluation pipeline.

In Figure 4, we compare 3D pseudo-labels generated using different occupancy threshold values, ranging from 1 to 25. The highest performance is observed at a threshold of 3, indicating that even noisy points contribute useful information. For our experiments, we selected a threshold of 10 to balance slightly faster generation time with comparable accuracy.

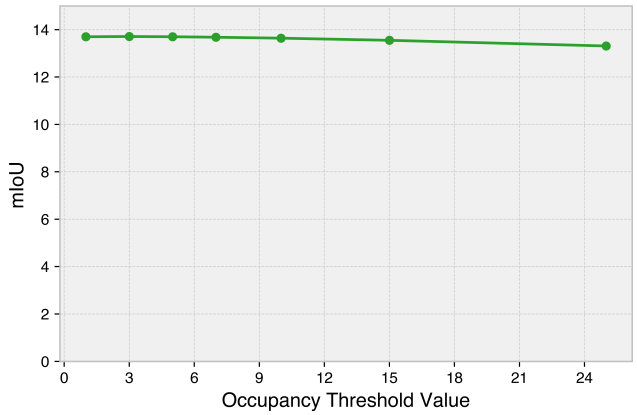

Fig. 4: **Occupancy Threshold Analysis:** 3D pseudo-labels compared to ground-truth labels for various threshold values in the generation processes. A camera mask is applied.

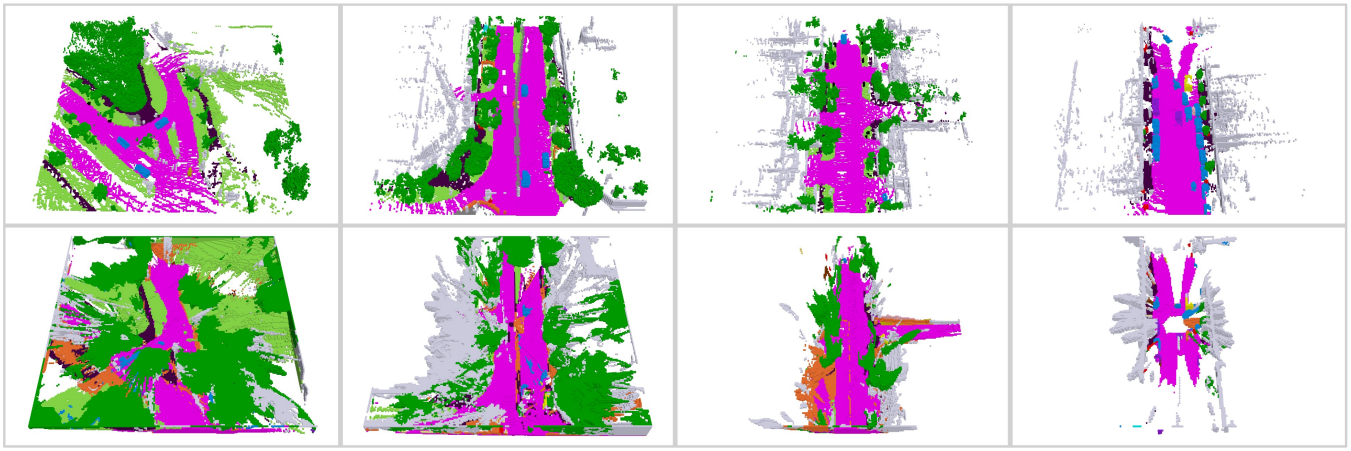

Fig. 5: **Occ3D Ground Truth (top) and Our 3D Pseudo-Labels(bottom):** Visual comparison of four samples from the Occ3D dataset with their corresponding 3D pseudo-labels generated by our method.

2) Qualitative Analysis: In Figure 5, we compare four 3D pseudo-label training samples with their corresponding ground-truth labels. The pseudo-labels closely match the ground truth, accurately identifying key scene elements such as roads, vegetation, and buildings. Despite mitigation efforts such as outlier removal and occupancy thresholding, our pseudo-labels remain cluttered, primarily due to noise in the depth maps. However, as will be discussed in Subsection V-F, the model's predicted outputs often appear smoother and more continuous than the ground-truth labels. In the rightmost sample, scene densification is lacking due to the sample being early in the sequence, resulting in limited aggregation of temporal data.

IV. Methodology II: Models

This section starts with Subsection IV-A, which outlines the loss function used for our 3D pseudo-labels, pseudo-loss. Subsection IV-B details the modifications made to existing models for integrating this loss function, while Subsection IV-C introduces our model, EasyOcc, which solely utilizes this loss function. The section concludes with Subsection IV-D, which explains the camera mask and describes the purpose of evaluating the models with and without it.

A. Pseudo Loss

Now that we have described the 3D pseudo-label generation process, we shall outline the loss function for using the labels during training. The loss function, pseudo-loss, will be used in the chosen modified models and EasyOcc. This loss consists of two distinct terms, as shown in (5), where λ is a constant initialized at the start of training (an ablation on the value of λ is discussed in Table VI). The formulation is adapted from GaussianOcc, where it was initially used as an optional component for training with ground-truth labels [14].

$$\mathcal{L}_{\text{Pseudo}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{Geomety}} \tag{5}$$

As shown in Equation (6), geometry loss is composed of three separate losses: geometric scale loss, semantic scale loss, and Lovász softmax loss [43].

$$\mathcal{L}_{\text{Geometry}} = \mathcal{L}_{\text{geom\_scal}} + \mathcal{L}_{\text{sem\_scal}} + \mathcal{L}_{\text{Lovász}} \tag{6}$$

All losses, including cross-entropy, are standard in semantic occupancy prediction, as they effectively penalize misclassifications and support class re-weighting to address dataset imbalance.

B. Modifications to Existing Models

Incorporating pseudo-loss into the three selected architectures requires considering how the loss function interacts with existing losses and also specific implementation details, as outlined below.

As described in Section II, SelfOcc predicts semantic occupancy using a two-step process: binary occupancy prediction (occ), followed by a semantic voxel prediction (sem) which together produce the final output. This structure introduces two key considerations: (1) a voxel may be classified as occupied by occ but empty by sem, resulting in it being considered unoccupied, and (2) a voxel may be classified as unoccupied by occ but occupied by sem, leading to it remaining unoccupied.

Through preliminary testing, we observe that excluding occ from pseudo-loss computation and from the final scene representation improved performance in both IoU and mIoU metrics. This is perhaps explained by the considerations discussed above. Hence, the final scene representation is the semantic voxel, sem. The final loss function is defined in Equation (7). The additional losses present aid in SDF stability, multi-frame photometric consistency, RGB rendering, and 2D semantic loss.

$$\mathcal{L}_{\text{SelfOcc}} = \mathcal{L}_{\text{regularisation}} + \mathcal{L}_{\text{reprojection}} + \mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{sem}} + \mathcal{L}_{\text{Pseudo}} \tag{7}$$

In OccNeRF, pseudo-loss is implemented alongside additional loss components to form the final loss function, as shown in Equation (8). The pseudo-loss serves as a complement to the three existing losses in the original OccNeRF model: $\mathcal{L}_{\text{regularisation}}$, $\mathcal{L}_{\text{reprojection}}$, $\mathcal{L}_{\text{sem}}$. These losses

regulate voxel occupancy stability, multi-frame photometric consistency, and 2D semantic loss, respectively.

$$\mathcal{L}_{\text{OccNeRF}} = \mathcal{L}_{\text{regularisation}} + \mathcal{L}_{\text{reprojection}} + \mathcal{L}_{\text{sem}} + \mathcal{L}_{\text{Pseudo}} \quad (8)$$

Given the similarity between GaussianOcc and OccNeRF, our pseudo-loss implementation follows the same approach in GaussianOcc, with one key difference: we omit $\mathcal{L}_{\text{sem}}$ due to sporadic NaN gradients in the convolutional layers of the image encoder during training. The cause of this issue remains unknown. The resulting loss function is defined in Equation (9).

$$\mathcal{L}_{\text{GaussianOcc}} = \mathcal{L}_{\text{regularisation}} + \mathcal{L}_{\text{reprojection}} + \mathcal{L}_{\text{Pseudo}} \quad (9)$$

C. EasyOcc

Integrating pseudo-loss in EasyOcc is straightforward, as the framework relies exclusively on this signal for learning. The continuous and dense scene representation enables effective learning in conjunction with our 3D pseudo-labels.

The model is a simplified variant of GaussianOcc, with its architectural flow illustrated in Figure 1. Multi-view camera images are processed through a ResNet-101 image encoder to extract high-level features, which have shown robust performance across BEV models [22], [44] and semantic occupancy prediction frameworks [13], [14], [20]. Parameter-free bilinear sampling projects these features into 3D space, which are then passed through a 3D CNN to enhance spatial reasoning and produce semantic logits.

Up to this point, the pipeline mirrors OccNeRF and GaussianOcc. However, our approach to loss computation diverges by directly computing loss against the generated 3D pseudo-labels, eliminating the need for depth estimation, multi-frame consistency, and view synthesis, thus reducing training complexity and duration.

D. Camera Mask

Comprehensive scene modeling requires an understanding of the full surrounding environment. In prior occupancy datasets [26], [29], evaluation is conducted across the entire voxel grid, compelling models to accurately predict both occupancy and semantics for the entire scene.

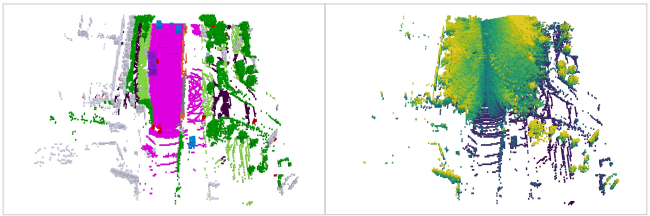

Fig. 6: **Camera Mask Visualization:** Semantic occupancy grid (**left**), and corresponding boolean camera mask (**right**). Camera mask includes empty voxels.

In contrast, models assessed on the Occ3D benchmark are evaluated solely on voxels visible from the current camera view. These visible voxels, referred to as the camera mask, are determined by ray casting from each surround-view camera in the current sample to the voxel grid [27]. This evaluation strategy limits overall scene comprehension and reduces penalties for issues such as overprediction and object duplication. Given the availability of temporal cues, the ability to infer occluded areas, and the use of priors, such as typical object dimensions, it is important to also evaluate models over the full voxel grid. Figure 6 illustrates a sample's semantic occupancy grid alongside its boolean camera mask. While the entire scene contains 640,000 voxels, models evaluated under Occ3D are assessed on only 67,761 of these.

As will be demonstrated in Subsection V-D, our 3D pseudo-labels enhance performance not only under the standard camera mask evaluation but also significantly boost results when evaluated over the full voxel grid. While previous works primarily report performance using the camera mask, we extend this by providing full-voxel grid evaluations for all models, highlighting that our 3D pseudo-labels contribute to a more holistic understanding of the scene.

V. Results

In this section, we present the main results of this paper. We begin with a description of the dataset and evaluation metrics in Subsection V-A, and a brief explanation of model configurations in Subsection V-B. In Subsections V-C and V-D, we examine the effectiveness of the deployment of our 3D pseudo-labels, evaluated with and without the camera-mask, respectively. This is followed by an ablation study for the EasyOcc in Subsection V-E. Subsection V-F analyzes the qualitative results based on the main results table. The section concludes with a summary and discussion of all findings in Subsection V-G.

A. Dataset and Evaluation Metrics

All models are trained and evaluated on the Occ3D dataset, which consists of 600 training scenes and 150 validation scenes from the nuScenes dataset. The voxel space is bounded by $[-40\text{m}, -40\text{m}, -1\text{m}, 40\text{m}, 40\text{m}, 5.4\text{m}]$, with a voxel size of 0.4m^3.

Models are evaluated using the Intersection over Union (IoU) and mean Intersection over Union (mIoU) metrics. IoU, defined in Equation (10), reflects the model's ability to capture overall spatial structure through occupancy. The mIoU metric, shown in Equation (11), computes the average IoU across all semantic classes, excluding the empty class (class index 17).

$$\text{IoU} = \frac{TP}{TP + FP + FN} \quad (10)$$

$$\text{mIoU} = \frac{1}{C-1} \sum_{\substack{c=0 \\ c \neq 17}}^{C} \frac{TP_c}{TP_c + FP_c + FN_c} \quad (11)$$

TP: True Positive, FP: False Positive, FN: False Negative

B. Model Configurations

In Table I, we compare all model configurations. Rendering time and depth estimation add overhead, with Gaussian

TABLE I: **Model Configurations:** Rendering time refers to the time required to render semantics, depth, or features for a single sample during training. Training time denotes the time per training epoch; the plus symbol (+) indicates the additional time incurred due to the inclusion of our pseudo-loss. * For OccNeRF, the parameter count includes the NeRF rendering module, which is used during training only. ** For GaussianOcc, the parameter count includes the pose estimation module, which is used during training only. (-) indicates the data was not listed in the paper or repository. TT-OccCamera and AutoOcc are not listed as they require no training.

Model	Backbone	Model Parameters	Image Size	Epochs	Rendering Time	Training Time
SelfOcc [12]	RN-50	35.4M	800×384	24	32ms	2hr 12m *(+22m)*
OccNeRF [13]	RN-101	179.1M*	672×336	24	1061ms	5hr 8m *(+31m)*
GaussianOcc [14]	RN-101	64.7M**	640×384	24	23ms	1hr 32m *(+11m)*
GaussTR [6]	DINOv2 Base	108.3M	896×504	24	20ms	1hr 9m
GaussianFlowOcc [6]	RN-50	-	704×256	18	-	-
EasyOcc (Ours)	RN-101	40.9M	640×384	24	0ms	1hr 25m

Splatting being the most efficient due to its rasterization-based rendering [11]. EasyOcc avoids these rendering methods during training, resulting in reduced training time. While incorporating pseudo-loss increases epoch training time, the impact varies by model, with OccNeRF experiencing the largest increase of (+31m) due to its overall slower training.

TABLE II: **EasyOcc Inference Time:** Time required for each component of the inference pipeline.

Process	Execution Time
Image Encoding	27ms
Bilinear Sampling	6ms
3D CNN	7ms
Grid Sampling	145ms
Total	185ms

In Table II, we report the inference time (in milliseconds) for each component of the EasyOcc model. Grid Sampling refers to downsampling the contracted coordinate voxel grid to align with the dimensions of the Occ3D ground-truth labels, a technique introduced in OccNeRF [13]. Preliminary experiments showed that retaining the contracted coordinate system, rather than modeling the scene in the Occ3D output space, improved performance.

Self-supervised models that utilize LiDAR during training or inference are excluded from our comparison [8], [18], [45], [46] as our method is a camera-only pipeline, consistent with the models used for comparison in this study. Training and inference are performed on four NVIDIA A100-SXM4-40GB GPUs.

C. Main Results: Camera Mask

In this section, we evaluate the performance of seven baseline models against the modified versions of the three selected baseline models and EasyOcc. We compare the models across four key evaluation categories: inference time (FPS), Intersection over Union (IoU), mean IoU (mIoU), and class-wise IoU for each semantic category. Results are provided in Table III.

1) Inference Time: GaussianFlowOcc achieves the highest inference speed at 10.2 FPS, attributed to its use of induced attention, which significantly reduces computational overhead. SelfOcc ranks second with 7.4 FPS, benefiting from the lack of a contracted coordinate system. The inclusion of pseudo-loss has no effect on inference time, as it influences only the training phase. EasyOcc matches the inference speed of both OccNeRF and GaussianOcc, all of which operate at 5.4 FPS.

2) Intersection over Union (IoU): AutoOcc secures the highest performance with an IoU of 83.01, significantly surpassing all other models. GaussianFlowOcc ranks second, achieving an IoU of 46.91. Notably, the addition of pseudo-loss led to a marked decline in IoU, for example, it reduces SelfOcc's score from 44.05 to 34.50, a 22% drop. This performance drop may be attributed to object duplication, as the model lacks the ability to reason about occluded regions. Consequently, it tends to predict occupancy beyond visible surfaces, resulting in an overly dense scene representation. This behavior is visualized in the qualitative analysis presented in Subsection V-F, and further substantiated by evaluations without the camera mask, assessing every voxel in the space, which shall be discussed in Subsection V-D.

3) Mean Intersection over Union (mIoU): A notable performance gap exists between the original voxel-based models (SelfOcc, OccNeRF, and GaussianOcc) and the Gaussian-based models: GaussTR, TT-OccCamera, GaussianFlowOcc, and AutoOcc. AutoOcc once again leads with an mIoU of 20.92, largely due to the integration of a VLM [32] and a VFM [37] during inference, which provides high-quality semantic predictions.

The incorporation of pseudo-loss enables substantial improvements to existing models, surpassing even the Gaussian-based methods: GaussTR and TT-OccCamera. When pseudo-loss is applied to OccNeRF, it achieves a 15% improvement over GaussTR and a 45% gain compared to the original OccNeRF model. Similar performance boosts are observed for both SelfOcc and GaussianOcc. Notably, SelfOcc gains the ability to predict previously unsupported classes, such as construction vehicles, thanks to semantic supervision from Grounded-SAM.

Our model, EasyOcc, reaches an mIoU of 13.86, placing it on par with the other three models that utilize rendering-based supervision along with pseudo-loss.

4) IoU per Semantic Class: AutoOcc, GaussianFlowOcc, and GaussTR achieve the highest performance across all

TABLE III: **State of the Art Comparison with the Camera Mask:** FPS denotes frames per second, representing the time taken to process a single sample. IoU and mIoU refer to Intersection over Union and mean Intersection over Union, respectively. The best result in each category is highlighted in **bold**, while the second-best is underlined. * OccNeRF implements 2D semantic loss, whereas GaussianOcc does not.

Model	FPS	IoU	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. surf.	sidewalk	terrain	manmade	vegetation
SelfOcc [12]	7.4	44.05	9.66	0.20	0.52	6.71	10.42	0.00	0.10	2.12	0.00	0.00	7.72	56.13	26.92	25.68	13.38	4.62
OccNeRF [13]	5.4	46.43	9.73	0.68	1.81	6.59	6.59	3.68	0.34	2.93	3.19	2.90	6.64	52.80	23.99	24.97	18.62	9.69
GaussianOcc [14]	5.4	42.91	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
TT-OccCamera [18]	0.7	-	1184	0.00	5.90	8.94	12.58	2.75	9.67	4.71	4.04	0.00	8.77	55.65	26.49	30.20	15.13	16.57
GaussTR [6]	0.3	44.54	12.27	6.50	8.54	**21.77**	**24.27**	**6.26**	15.48	7.94	1.86	**6.10**	**17.16**	36.98	17.21	7.16	21.18	9.99
GaussianFlowOcc [31]	**10.2**	46.91	15.07	**7.23**	9.33	17.55	17.94	4.50	9.32	8.51	10.66	2.00	11.80	**63.89**	**31.11**	**35.12**	14.64	12.59
AutoOcc [9]	-	**83.01**	**17.87**	2.70	**10.45**	7.81	20.42	5.79	**17.58**	**18.50**	**24.25**	4.23	12.88	55.54	24.23	27.14	**35.62**	**36.61**
SelfOcc [12] + Ours	7.4	34.50	13.43	1.66	5.43	14.51	22.22	2.60	6.39	15.38	8.90	1.00	12.77	54.97	26.81	21.55	11.34	9.29
OccNeRF [13] + Ours *	5.4	38.46	14.09	1.85	8.18	16.66	22.12	1.01	7.74	14.74	12.84	0.98	13.76	55.91	27.96	22.73	15.77	17.20
GaussianOcc [14] + Ours *	5.4	38.78	13.89	1.74	5.85	16.22	22.34	2.30	8.39	15.71	10.18	0.99	13.36	55.29	27.36	23.41	15.96	16.97
EasyOcc (Ours) RN-101	5.4	38.86	13.86	1.90	6.67	15.09	21.68	2.70	8.06	15.28	11.08	1.36	12.81	55.78	27.85	22.06	16.07	17.30

TABLE IV: **Model Comparison evaluated without the Camera Mask:** Models are retrained from Table III without the camera mask. FPS denotes frames per second, representing the time taken to process a single sample. IoU and mIoU refer to Intersection over Union and mean Intersection over Union, respectively. The best result in each category is highlighted in **bold**, while the second-best is underlined. * OccNeRF implements 2D semantic loss, whereas GaussianOcc does not.

Model	FPS	IoU	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. surf.	sidewalk	terrain	manmade	vegetation
SelfOcc [12]	**7.4**	6.96	2.78	0.43	0.45	7.45	5.58	0.00	0.07	1.96	0.00	0.00	5.45	5.65	6.55	4.73	2.84	3.33
OccNeRF [13]	5.4	5.95	2.57	0.03	0.16	2.42	1.28	0.73	0.03	0.37	0.13	0.45	0.92	21.08	5.11	2.84	3.88	4.33
GaussianOcc [14]	5.4	5.33	5.90	0.59	3.27	11.94	7.99	0.49	1.43	5.04	5.19	0.32	7.28	24.34	13.05	10.55	1.42	7.47
GaussTR [6]	0.3	11.58	4.37	**1.49**	1.96	**12.22**	9.13	**1.67**	**6.58**	3.30	1.09	**1.56**	8.62	9.33	3.72	2.12	5.00	6.43
SelfOcc [12] + Ours	**7.4**	**18.22**	7.91	0.80	2.65	10.33	**15.30**	1.45	3.25	10.37	4.03	0.56	8.99	27.27	15.32	12.06	6.47	7.78
OccNeRF [13] + Ours *	5.4	16.88	**8.21**	0.84	**5.16**	11.62	14.71	0.56	4.94	**11.72**	**7.04**	0.66	**9.34**	27.37	**16.22**	**12.37**	6.71	10.24
GaussianOcc [14] + Ours *	5.4	17.13	7.77	0.80	2.98	10.47	14.81	0.95	4.34	11.32	4.63	0.57	8.74	**27.60**	15.88	12.03	**6.84**	10.22
EasyOcc (Ours) RN-101	5.4	16.98	7.71	0.86	3.22	9.57	14.46	1.14	4.31	11.07	5.08	0.74	8.32	27.48	15.89	11.83	6.83	**10.28**

semantic classes. AutoOcc leads in 6 out of 15 classes, particularly excelling in smaller object categories such as bicycle, motorcycle, and traffic cone, likely due to the use of foundation models during inference. GaussianFlowOcc ranks first in 4 classes, primarily large-scale categories such as drivable surface, sidewalk, and terrain, which may be attributed to the strong emphasis on temporal information during training. GaussTR attains top performance in 5 classes, notably dynamic objects such as bus and car, which is an interesting outcome given the absence of a dedicated flow module or temporal modeling, which GaussianFlowOcc and AutoOcc implement.

None of the voxel-based models, SelfOcc, OccNeRF, GaussianOcc, EasyOcc, or their pseudo-loss-enhanced variants achieves top performance in any class. Nevertheless, with the integration of pseudo-loss, these models deliver competitive results, securing second place in several classes. Significant improvements in IoU are observed across nearly all categories. For GaussianOcc, IoU increases in 14 out of 15 classes, with the barrier class as the sole exception. Similar trends are seen in SelfOcc and OccNeRF. Notably, SelfOcc demonstrates a dramatic 624% increase in IoU for the pedestrian class, emphasizing the importance of 3D labels for accurately detecting vulnerable object categories. EasyOcc exhibits strong detection capabilities, achieving performance comparable to that of the modified models enhanced with pseudo-loss.

D. Main Results: No Camera Mask

In this section, we re-examine the previously reported results, this time excluding the use of the camera mask, as motivated in earlier Subsection IV-D. The models are evaluated in the updated results Table IV based solely on IoU, mIoU, and per-class IoU. FPS is omitted as it remains the same as the previous. TT-OccCamera, GaussianFlowOcc, and AutoOcc are excluded from this analysis due to the lack of publicly available implementations.

1) Intersection over Union (IoU): Compared to evaluation with the camera mask, we see a stark drop in performance, most notably for the original models, where GaussTR observes a 74% decrease. EasyOcc observes much less of a decrease of 56%, achieving a higher overall IoU of 16.98 compared to GaussTR's 11.58. All models incorporating pseudo-loss experience massive increases in IoU, with SelfOcc achieving the highest IoU of 18.22, a 162% increase over the original SelfOcc's IoU of 6.96. This is in contrast to the camera mask models, where pseudo-loss worsened IoU metrics, displaying that our labels do indeed enable a better

scene representation.

2) Mean Intersection over Union (mIoU): The trend continues into mIoU, where again the original models suffer much greater on average than our models, when compared to evaluation with the camera mask. One outlier is GaussianOcc, which observes a 41% decrease, whereas GaussianOcc with pseudo-loss observes a 44% decrease. However, our models still achieve the best performance, with OccNeRF including our 3D pseudo-labels achieving 8.21 mIoU. EasyOcc achieves an mIoU of 7.71, despite using many fewer loss computation techniques than OccNeRF.

We note that both SelfOcc and OccNeRF take the biggest hit in performance when evaluating on the entire grid. This likely originates from their overprediction and volume rendering, which causes object duplication, heavily penalizing the mIoU metric regarding false positives (FP), as seen in Equation (11). This is more evident when examining IoU per semantic class, and is further evident in the qualitative analysis Subsection V-F.1.

3) IoU per Semantic Class: Here, we examine a clear distinction between the models that utilize the 3D pseudo-labels and the ones that do not. SelfOcc, OccNeRF, and GaussianOcc fail to achieve top performance across any class, whereas GaussTR successfully achieved top performance on 5 out of 16 total classes. However, the remaining top-performing models in all other classes use 3D pseudo-labels. For example, when applying our 3D pseudo-labels to OccNeRF, the model outperforms GaussTR in 10 out of 16 semantic classes and achieves the highest performance on six classes across all evaluated models.

Looking further into specific classes, we find that 3D pseudo-labels enable much better detection on all classes. For pedestrian detection, GaussianOcc achieved 5.04 mIoU, and this is increased to 11.32 mIoU when 3D pseudo-labels are used for supervision. For the original models, some classes do achieve respectable performance. For example, GaussianOcc achieves an mIoU of 24.34 for drivable surface, and this increases to just 27.60 mIoU when 3D pseudo-labels are implemented. A similar observation is seen for OccNeRF. For SelfOcc, all semantic classes show improved performance with the integration of our 3D pseudo-labels. In the case of OccNeRF, all classes improve except for construction vehicles, while for GaussianOcc, all classes improve except for bicycle, bus, and traffic cone. Notably, the decline in mIoU for these few classes is marginal.

For EasyOcc, we examine performance comparable with that of the models incorporating pseudo-loss, achieving top performance for one category only, vegetation, due to the dominance of OccNeRF with pseudo-labels.

E. Ablation Study: EasyOcc

In this section, we present a series of ablation studies on the EasyOcc model to assess the impact of key design choices. Models are trained for 12 epochs, balancing thorough evaluation with practical constraints on energy consumption and training time. First, in Subsubsection V-E.1, we examine the effect of varying the image encoder and input image resolution. This is followed by an analysis of temporal sample aggregation in Subsubsection V-E.2. Next, in Subsubsection V-E.3, we evaluate the sensitivity of pseudo-loss to changes in the weighting constant λ. In Subsubsection V-E.4, we investigate the effect of selectively removing components of the pseudo-loss. Finally, in Subsubsection V-E.5, we explore the performance of Lovász-Softmax loss in more detail. All evaluation is performed with the camera mask.

1) Image Encoder and Image Size: In Table V, we investigate the effect of varying the image encoder and input image resolution on the mIoU metric, total model parameters, and inference speed (FPS).

TABLE V: **Ablation: Image Encoder and Input Image Size:** Models are trained for 12 epochs to ensure reasonable training time. The best-performing model in each category for each input resolution is highlighted in **bold**.

Encoder	Image Size	mIoU	Model Parameters	FPS
RN-152	640×384	**13.51**	56.5M	5.1
RN-101		13.18	40.9M	5.4
RN-50		13.08	21.9M	5.6
RN-34		13.11	16.2M	5.8
RN-18		12.73	**10.8M**	**6.0**
RN-152	320×192	**12.16**	56.5M	5.4
RN-101		12.02	40.9M	5.6
RN-50		11.98	21.9M	5.8
RN-34		11.63	16.2M	6.0
RN-18		11.28	**10.8M**	**6.2**

First, we analyze the models using the full input resolution of 640×384. With shallower ResNet backbones, mIoU decreases, model parameters decrease, and FPS improves as expected. The performance difference between RN-18 and RN-152 is 0.78 mIoU points, suggesting that encoder depth may be less critical than one thought. However, a 0.33 mIoU gain is observed when moving from RN-101 to RN-152, indicating that deeper models still offer noticeable benefits.

While shallower models typically exhibit higher inference speeds, this trend is less prominent in our results due to the inference bottleneck introduced by grid sampling, inherited from the contracted coordinate system used in OccNeRF and GaussianOcc. Thus, for a marginal trade-off in inference speed, deeper models like RN-101 or RN-152 may be preferable. However, deeper models also come with a substantial increase in parameter count—transitioning from RN-18 to RN-152 results in a 423% increase in parameters. Therefore, selecting the encoder requires balancing performance, inference speed, and memory constraints.

Next, we evaluate the models with a reduced input resolution of 320×192. This leads to a performance drop across all architectures. For instance, RN-101 suffers a 1.16-point decrease in mIoU, highlighting the importance of high-resolution input. A similar trend is seen when comparing the RN-18 and RN-152 models at this lower resolution, where a 0.88 mIoU drop is observed.

2) Aggregation of Temporal Samples: As shown in the previous Figure 3, the use of temporal samples significantly

improves the similarity of our 3D pseudo-labels to the Occ3D ground-truth annotations. In Figure 7, we extend this analysis by training the EasyOcc model with varying numbers of temporal samples to evaluate whether a similar trend holds in model performance post-training.

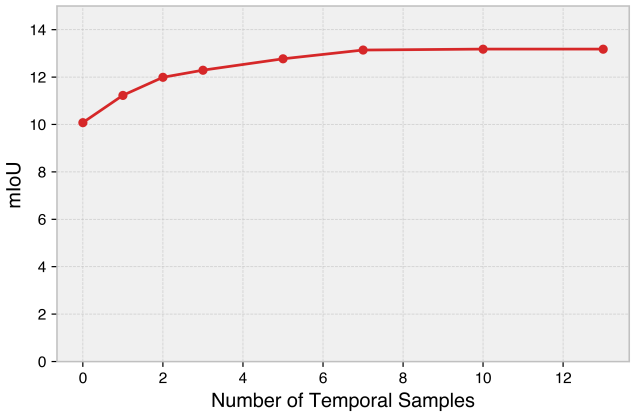

Fig. 7: **Ablation: Temporal Samples:** Models are trained for 12 epochs to maintain manageable training time.

The figure exhibits a logarithmic curve, seen previously when comparing Occ3D ground-truth labels to the 3D pseudo-labels, with the results clearly showing that increasing the number of temporal samples improves the mIoU metric. This improvement is attributed to scene densification, especially in regions farther from the camera's field of view. Notably, we achieve an mIoU of 13.18 with both 10 and 13 temporal samples, further indicating a saturation point beyond which additional samples yield diminishing returns in the mIoU metric.

3) Variation of λ: In Table VI, we experiment with the λ constant in Equation (5) to balance the contributions of the cross-entropy and geometry loss components. The results show that a λ value of 0.1 yields the best mIoU performance. A value of 1 performs slightly worse, trailing by 0.22 points, while a value of 0.01 results in a significant drop of 0.80 points. These findings indicate that geometry loss plays a meaningful role, but balancing the contributions of both cross-entropy and geometry losses is crucial. With $\lambda = 0.1$, the magnitudes of the cross-entropy and geometry losses are approximately equal during testing, underscoring the importance of careful weighting for optimal performance.

TABLE VI: **Ablation: Variation of λ in Pseudo-Loss:** Models are trained for 12 epochs to ensure practical training time. The best-performing configuration is highlighted in **bold**.

Lambda λ	mIoU
0.01	12.38
0.1	**13.18**
1	12.96

4) Choice of Losses: In Table VII, we ablate the pseudo-loss in Equation (5) by removing individual components, such as the cross-entropy term, and elements of the geometry loss in Equation (6), to assess the contribution of each component to the model's learning.

TABLE VII: **Ablation: Dropping Losses in Pseudo-Loss:** Models are trained for 12 epochs to maintain feasible training time. The best-performing configuration is highlighted in **bold**.

Cross En.	Geometry Scale	Semantic Scale	Lovász	mIoU
✓	✓	✓	✓	13.18
✓	✓	✓		**13.29**
	✓	✓	✓	12.21
	✓	✓		12.54
✓			✓	11.02
✓				8.15
			✓	8.71

Overall, the results show that removing components of the loss function reduces the model's learning capability. For example, excluding the cross-entropy loss decreases mIoU by 0.97 points. Similar drops occur when individual components of the geometry loss are removed. Interestingly, excluding the Lovász-Softmax loss results in a slight mIoU improvement of 0.11, which is unexpected since it is designed to optimize the IoU metric. We hypothesize this anomaly arises from including the empty class index (17) in the Lovász loss computation. To validate this, we retrain the model excluding this index, as detailed in the following ablation.

5) Lovàsz Softmax Loss: Following the previous experiment, we retrain the model with the Lovász-Softmax loss that excludes the empty index from the loss computation. The results are presented in Table VIII.

TABLE VIII: **Ablation: Lovász-Softmax Empty Index:** Models are trained for 12 epochs to ensure manageable training time. The best-performing configuration is highlighted in **bold**.

Ignore Empty	mIoU
✗	13.18
✓	**13.55**

Excluding the empty class from the loss computation results in a 0.37 mIoU improvement. Only the previous experiments in the ablation section were conducted with the inclusion of the empty label in the Lovász loss. However, the integrity of the comparative results is maintained, as all models were subject to the same conditions and limitations.

F. Qualitative Results

In this section, we present a qualitative analysis in two parts. First, we evaluate the quality of predicted semantic voxels for all models compared to the ground-truth labels. Following this, we examine the voxel predictions from each of the six camera views, comparing GaussianOcc to its variant trained with our proposed pseudo-loss to examine the benefits of our 3D pseudo-labels. TT-OccCamera, GaussianFlowOcc, and AutoOcc are omitted because they are not currently open-source.

1) Voxel Analysis: We begin the qualitative analysis by inspecting Figure 8, which shows semantic voxel visualizations for each model in the SOTA comparison in Table IV. These visualizations provide insight into the strengths and limitations of each model, starting with a comparison of the original models and their counterparts augmented with our proposed pseudo-loss.

SelfOcc performs well in road segmentation but misses vehicles in the back camera view. With the implementation of pseudo-loss, the model correctly identifies these vehicles. However, with the inclusion of pseudo-loss, we observe a notable increase in misclassifications as the *barrier* class, particularly in the back-left camera view—an issue not present in the original model. This may be attributed to the presence of a metal fence and a construction road sign in the corresponding image.

OccNeRF predicts more elements than SelfOcc, including a car and a bus in the rear view, but introduces noticeable object duplication, which results in heavy penalisation when evaluating without the camera mask. Pseudo-loss reduces this duplication and noticeably improves vegetation identification in the front camera view. A similar improvement is seen in **GaussianOcc**, where pseudo-loss corrects mispredictions, such as classifying road as wall in the rear view.

Comparing our model, **EasyOcc**, to **GaussTR**, key differences emerge. GaussTR fails to predict vegetation in the front camera view, whereas EasyOcc does. Additionally, the overhang of a building is missing from GaussTR's output but is present in EasyOcc's. GaussTR excels at segmenting dynamic objects, producing more complete representations, while EasyOcc only predicts the visible portions, leading to hollow or partial reconstructions. This aligns with the quantitative results in Table III, where GaussTR performs better in dynamic object segmentation. However, EasyOcc benefits from a voxel-based scene representation, enabling smooth predictions, while GaussTR's Gaussian-based representation produces more fragmented outputs.

The use of our **3D pseudo-labels** offers several advantages. Firstly, our models correctly identify the region beneath the ego vehicle as road due to temporal information during 3D pseudo-label generation. Models without our labels often fail here, as this region is typically absent in camera views. Secondly, it aids in accurately predicting the overhang of a building, highlighted in the back-right camera view, which is missing in the ground-truth labels and other models' predictions. Finally, it is particularly effective in preventing object duplication and reducing scene densification, which significantly enhances performance during evaluation without the camera mask.

2) Image View Analysis: Model predictions and our 3D pseudo-labels are generated solely from camera views. Hence, we shall examine the semantic voxel predictions of GaussianOcc and GaussianOcc trained with 3D pseudo-labels from the perspective of the six surrounding cameras, as shown in Figure 9.

Starting with the **front left** camera, we observe a mislabeling in the ground-truth data, where a pole is annotated incorrectly. GaussianOcc predicts the pole but is penalized due to the incorrect label, while our model predicts vegetation in its place. In the **front** view, our model mistakenly predicts a bus (yellow) and truck (purple) in the distance, likely confused by a distant car in the RGB image, but both models perform well otherwise. In the **front right** view, both models detect a pedestrian, but the position is inaccurate due to partial occlusion by a pole. Neither model correctly labels the fire hydrant or trash can as manmade, though both correctly identify the motorcycle.

In the **back right** view, our model correctly identifies the overhang of a building, which GaussianOcc fails to. Both models detect the pedestrians and a traffic cone, which is absent from the ground-truth annotation. In the **back** view, GaussianOcc fails to predict the road in the lower part of the image, while our model produces a continuous road surface. Additionally, GaussianOcc erroneously predicts a wall behind the bus and vehicle, an object not present in the ground-truth labels or the RGB image, and absent in our model's output. In the **back left** view, our model correctly identifies a traffic light, whereas GaussianOcc only partially detects it and produces several incorrect labels for objects beyond the visible vehicles, including barrier, pedestrian, and construction vehicle, none of which are supported by the ground-truth or RGB image.

G. Discussion

This section provides a summary of the results and analyses presented in this paper.

3D pseudo-labels demonstrated the significant benefits of integration into existing models, especially in relation to the mIoU metric when evaluating with the camera mask. For instance, OccNeRF showed a 45% mIoU improvement due to high-quality supervision from Grounded-SAM and Metric3Dv2. Critical semantic classes, such as bicycle, car, and pedestrian, saw substantial gains, with the IoU for pedestrian increasing by over 627%. Our standalone model, EasyOcc, achieved a respectable performance with an mIoU of 13.86. While these results are promising, they remain behind the Gaussian-based models, GaussianFlowOcc and AutoOcc, which benefit from the incorporation of temporal information and Vision-Language Models (VLMs).

When evaluating the models without the camera mask, it becomes evident that the 3D pseudo-labels contribute to a more holistic representation of the scene, as reflected in improvements to IoU, mIoU, and per-class IoU. Notable gains were observed for SelfOcc, OccNeRF, and GaussianOcc, largely due to the reduction in object duplication and overprediction. However, this trend extends beyond these models; our approach also outperforms GaussTR, which employs a Gaussian-based scene representation. In this case, the improvements are likely due to our model producing more confident predictions with fewer incorrect outputs.

In the ablation study, we evaluated architectural and training design choices. We found only a marginal performance drop with a ResNet-18 backbone compared to ResNet-101, with an mIoU of 12.73 and a slight increase in inference

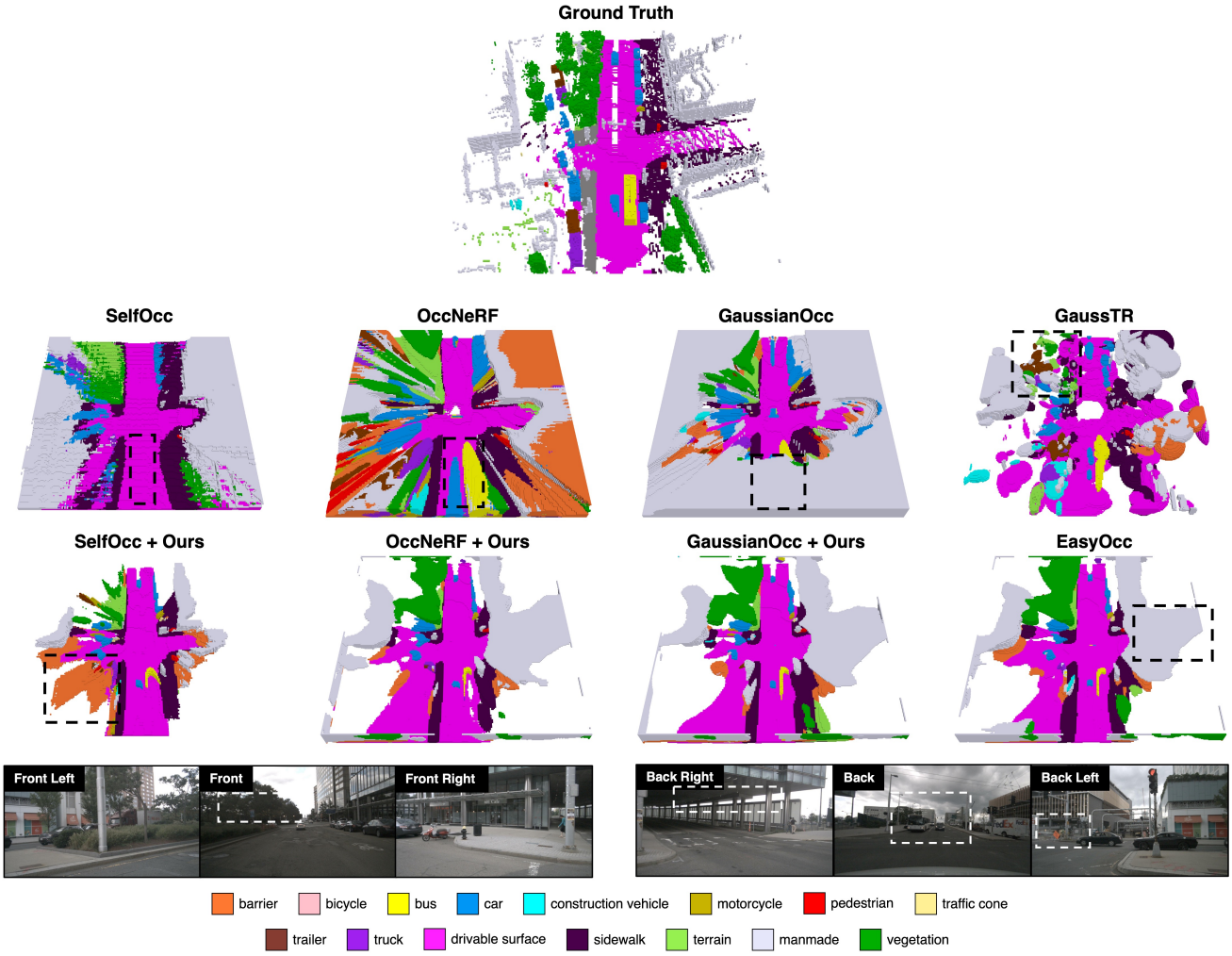

Fig. 8: **Voxel Qualitative Analysis:** Visual analysis of models evaluated in Table III. This analysis is conducted on nuScenes scene-783, sample token `e67f3e81225f426f8e1743af45487762`. Boxes are overlaid to highlight specific areas for qualitative analysis.

speed (5.4 FPS to 6 FPS), highlighting the compensatory power of 3D pseudo-labels. The importance of aggregating temporal samples was also validated, with 10 and 13 temporal samples yielding the best results, suggesting a saturation point beyond which additional samples provide minimal gains. Further ablations on λ in the pseudo-loss, selective removal of loss components, and refinements to Lovász-Softmax loss provided deeper insights into EasyOcc's learning dynamics.

Qualitative analysis reinforced findings of the main results. 3D pseudo-labels produced cleaner, less noisy semantic voxel outputs and made more accurate predictions for occluded or partially visible objects, benefiting from temporal information in label generation.

Some limitations include increased memory usage and spatial resolution constraints when utilizing a voxel representation, which is necessary for our 3D pseudo-labels. To achieve finer voxel-level predictions, new 3D pseudo-labels would need to be generated and retrained at the desired resolution. This contrasts with models using discrete 3D representations, such as Gaussians, which offer more flexible resolution scaling [18].

Overall, we emphasize the importance of performing loss computation directly in 3D space for self-supervised models, especially when combining it with temporal aggregation from prior samples. Importantly, aggregation is only required during training, ensuring efficient deployment. By omitting 2D loss computation, we eliminate computationally expensive rendering operations, reducing training time.

VI. Conclusion

This paper details the use of 3D pseudo-labels for fully self-supervised semantic occupancy prediction, enabling loss computation directly in 3D space, as opposed to the conventional 2D camera space approach. Our method is easily integrable into existing architectures, yielding increased model performance and more holistic scene representation. The effectiveness of these 3D pseudo-labels highlights their potential to significantly enhance self-supervised models.

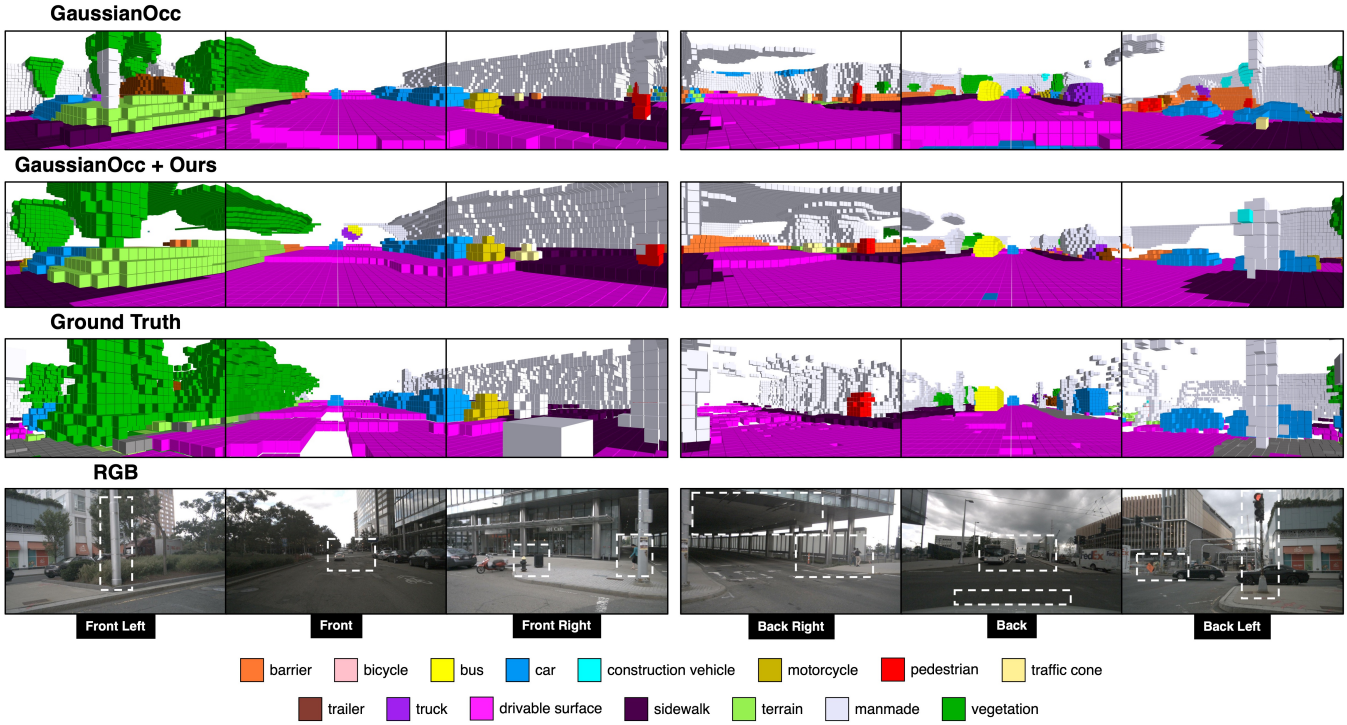

Fig. 9: **Image View Qualitative Analysis:** Examination of semantic voxel quality across the six camera views for GaussianOcc and GaussianOcc trained with 3D pseudo-labels. Conducted on nuScenes scene-783, sample token `e67f3e81225f426f8e1743af45487762`. Boxes are overlaid on the RGB images to highlight specific areas for qualitative analysis.

However, several avenues for future work remain to further refine and evaluate the proposed approach:

1) Incorporating LiDAR data into the 3D pseudo-label generation pipeline to facilitate comparison with LiDAR-supervised models.
2) Conducting a more comprehensive investigation into the integration of 3D pseudo-labels within models that utilise a Gaussian scene representation.
3) Evaluating the robustness of 3D pseudo-labels under challenging driving conditions, such as rain, fog, and low-light environments.

Self-supervised semantic occupancy prediction models have historically lagged behind supervised counterparts. However, recent advancements, including those presented in this work, suggest that self-supervised methods are rapidly gaining traction and may soon close the performance gap. While research trends shift toward discrete Gaussian representations, the widespread adoption of 3D pseudo-labels for voxel scene representation models remains uncertain. Nonetheless, this study shows that incorporating temporal information and carefully considering the domain for loss computation are crucial for achieving optimal performance in semantic occupancy prediction.

References

[1] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XLVII*. Berlin, Heidelberg: Springer-Verlag, 2024, p. 38–55. [Online]. Available: https://doi.org/10.1007/978-3-031-72970-6_3

[2] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen, "Metric3dv2 v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–18, 2024.

[3] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 4015–4026.

[4] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024.

[5] H. Zhang, F. Li, X. Zou, S. Liu, C. Li, J. Yang, and L. Zhang, "A simple framework for open-vocabulary segmentation and detection," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 1020–1031.

[6] H. Jiang, L. Liu, T. Cheng, X. Wang, T. Lin, Z. Su, W. Liu, and X. Wang, "Gausstr: Foundation model-aligned gaussian transformer for self-supervised 3d spatial understanding," *arXiv preprint arXiv:2412.13193*, 2024.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[8] P. Li, S. Ding, Y. Zhou, Q. Zhang, O. Inak, L. Triess, N. Hanselmann, M. Cordts, and A. Zell, "Ago: Adaptive grounding for open world 3d occupancy prediction," 2025. [Online]. Available: https://arxiv.org/abs/2504.10117

[9] X. Zhou, J. Wang, Y. Wang, Y. Wei, N. Dong, and M.-H. Yang, "Autoocc: Automatic open-ended semantic occupancy annotation via

vision-language guided gaussian splatting," 2025. [Online]. Available: https://arxiv.org/abs/2502.04981

[10] 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.

[11] 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.

[12] 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. (CVPR)*, 2024, pp. 19 946–19 956.

[13] 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. (CVPR)*, 2023, pp. 20 722–20 731.

[14] W. Gan, F. Liu, H. Xu, N. Mo, and N. o Yokoya, "Gaussianocc: Fully self-supervised and efficient 3d occupancy estimatio n with gaussian splatting," *CoRR*, vol. abs/2408.11447, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2408.11447

[15] M. Pan, J. Liu, R. Zhang, P. Huang, X. Li, L. Liu, and S. Zhang, "Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision," *arXiv preprint arXiv:2309.09502*, 2023.

[16] Q. Sun, C. Shu, Z. Sifan, Z. Yu, Y. Chen, D. Yang, and Y. Chun, "Gsrender: Deduplicated occupancy prediction via weakly supervised 3d gaussian splatting," 12 2024.

[17] 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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 15 035–15 044.

[18] F. Zhang, H. Yang, Z. Zhang, Z. Huang, and Y. Luo, "Tt-gaussocc: Test-time compute for self-supervised occupancy prediction via spatio-temporal gaussian splatting," 2025.

[19] Z. Liao, P. Wei, S. Chen, H. Wang, and Z. Ren, "Stcocc: Sparse spatial-temporal cascade renovation for 3d occupancy and scene flow prediction," 2025. [Online]. Available: https://arxiv.org/abs/2504.19749

[20] Z. Yang, Y. Dong, and H. Wang, "Daocc: 3d object detection assisted multi-sensor fusion for 3d occupancy prediction," *arXiv preprint arXiv:2409.19972*, 2024.

[21] Z. Ming, J. S. Berrio, M. Shan, and S. Worrall, "Occfusion: Multi-sensor fusion framework for 3d semantic occupancy prediction," *IEEE Trans. Intell. Vehicles*, pp. 1–13, 2024.

[22] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, "Simple-bev: What really matters for multi-sensor bev perception?" in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2759–2765.

[23] J. Philion and S. Fidler, "Lift-splat-shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Computer Vision – ECCV 2020: 16th European Conference*, vol. XIV. Glasgow, UK: Springer-Verlag, 2020, pp. 194–210.

[24] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Eur. Conf. Comput. Vis.*, 2022, pp. 1–18.

[25] 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 *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 2774–2781.

[26] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, J. Zhou, and J. Lu, "Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving," in *2023 IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 21 672–21 683.

[27] X. Tian, T. Jiang, L. Yun, Y. Mao, H. Yang, Y. Wang, Y. Wang, and H. Zhao, "Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, p. 64318–64330.

[28] B. Zhu, Z. Wang, and H. Li, "nucraft: Crafting high resolution 3d semantic occupancy for unified 3d scene understanding," in *Proceedings of the IEEE/CVF Conference on European Conference on Computer Vision*, 2024.

[29] X. Wang, Z. Zhu, W. Xu, Y. Zhang, Y. Wei, X. Chi, Y. Ye, D. Du, J. Lu, and X. Wang, "Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 17 804–17 813.

[30] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 618–11 628.

[31] S. Boeder, F. Gigengack, and B. Risse, "Gaussianflowocc: Sparse and weakly supervised occupancy estimation using gaussian splatting and temporal flow," 2025. [Online]. Available: https://arxiv.org/abs/2502.17288

[32] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, B. Li, P. Luo, T. Lu, Y. Qiao, and J. Dai, "Intern vl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 24 185–24 198.

[33] Z. Lin, Y. Wang, and Z. Tang, "Training-free open-ended object detection and segmentation via attention as prompts," 2024. [Online]. Available: https://arxiv.org/abs/2410.05963

[34] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," 2020. [Online]. Available: https://arxiv.org/abs/2005.00928

[35] L. Piccinelli, Y.-H. Yang, C. Sakaridis, M. Segu, S. Li, L. V. Gool, and F. Yu, "Unidepth: Universal monocular metric depth estimation," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 10 106–10 116.

[36] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong, "Faster segment anything: Towards lightweight sam for mobile applications," 2023. [Online]. Available: https://arxiv.org/abs/2306.14289

[37] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," 2023. [Online]. Available: https://arxiv.org/abs/2306.12156

[38] L. Barsellotti, L. Bianchi, N. Messina, F. Carrara, M. Cornia, L. Baraldi, F. Falchi, and R. Cucchiara, "Talking to dino: Bridging self-supervised vision backbones with language for open-vocabulary segmentation," 2024. [Online]. Available: https://arxiv.org/abs/2411.19331

[39] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Communications*, vol. 15, p. 654, 2024.

[40] S. Hayes, G. Sistu, and C. Eising, "Leveraging frozen foundation models and multimodal fusion for bev segmentation and occupancy prediction," *IEEE Open Journal of Vehicular Technology*, pp. 1–22, 2025.

[41] A. Bochkovskii, A. Delaunoy, H. G. an d Marcel Santos, Y. Zhou, S. R. Richter, and V. Koltun, "Depth pro: Sharp monocular metric depth in less than a second," in *International Conference on Learning Representations*, 2025. [Online]. Available: https://arxiv.org/abs/2410.02073

[42] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[43] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4413–4421.

[44] J. Schramm, N. Vödisch, K. Petek, B. R. Kiran, S. Yogamani, W. Burgard, and A. Valada, "Bevcar: Camera-radar fusion for bev map and object segmentation," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 1435–1442.

[45] S. Sze, D. D. Martini, and L. Kunze, "Minkocc: Towards real-time label-efficient semantic occupancy prediction," 2025. [Online]. Available: https://arxiv.org/abs/2504.02270

[46] J. Zheng, P. Tang, Z. Wang, G. Wang, X. Ren, B. Feng, and C. Ma, "Veon: Vocabulary-enhanced occupancy prediction," 2024.