Achieving Speed-Accuracy Balance in Vision-based 3D Occupancy Prediction via Geometric-Semantic Disentanglement

Yulin He*, Wei Chen†*, Siqi Wang†, Tianci Xun, Yusong Tan

School of Computer, National University of Defense Technology, Changsha, China
{heyulin, chenwei, wangsiqi10c, xuntianci22, ystan}@nudt.edu.cn

Abstract

Occupancy prediction plays a pivotal role in autonomous driving (AD) due to its capabilities of fine-grained 3D perception and general object recognition. However, existing methods often incur high computational costs, which conflict with AD's real-time demand. To this end, we redirect the focus from accuracy only to both accuracy and efficiency. By conducting a head-to-head comparison of existing methods, we find it challenging to balance accuracy and efficiency. We identify a core issue for this challenge: ***the strong coupling between geometry and semantics***. Specifically, the predicted geometric structure (*e.g.*, depth) guides the projection of 2D image features into 3D voxel space, which significantly affects feature discriminability and subsequent semantic learning. To address this issue, we focus on two key aspects: ***model design*** and ***learning strategies***. 1) For model design, we propose a dual-branch network that disentangles the representation of geometry and semantics. The voxel branch utilizes a novel re-parameterized large-kernel 3D convolution to refine geometric structure efficiently, while the BEV branch employs temporal fusion and BEV encoding for efficient semantic learning. 2) For learning strategies, we propose to separate geometric learning from semantic learning by the mixup of ground-truth and predicted depths. Our method achieves 39.4% mIoU at 20 FPS on Occ3D-nuScenes, showcasing a state-of-the-art balance between accuracy and efficiency.

Code — https://github.com/harrylin-hyl/GSD-OCC

Introduction

Vision-based occupancy prediction (Tesla 2021) aims to utilize surround-view camera images of ego vehicle to estimate object occupancy and semantics within a 3D voxel space (Cao and de Charette 2022; Tian et al. 2023; Li et al. 2023c). Compared to 3D object detection (Liang et al. 2022; Yin, Zhou, and Krahenbuhl 2021; Lang et al. 2019), it offers finer-grained 3D scene perception and can identify general objects by learning object occupancy, effectively handling out-of-vocabulary and unusual obstacles.

Despite these strengths, existing methods (Wang et al. 2024; Ma et al. 2024) often suffer from low inference speed and high memory usage, *e.g.*, $1 \sim 3$ FPS and $> 10{,}000$ MB on Nvidia A100. These limitations hinder their application in AD vehicles equipped with on-board GPUs. To this end, we aim to redirect the focus from accuracy only to a balanced emphasis on both accuracy and efficiency. We assess the speed and memory usage of publicly available methods to offer a more deployment-friendly comparison.

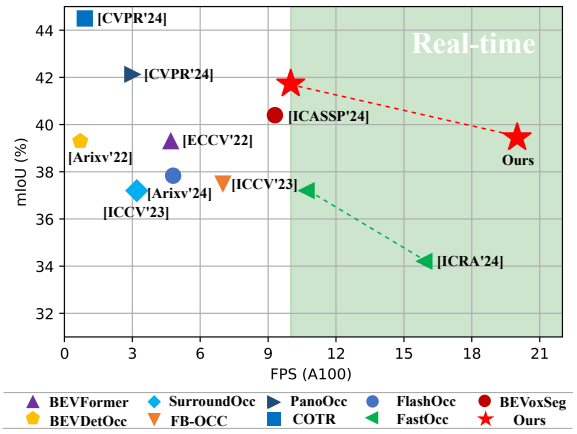

Figure 1: **The inference speed (FPS) and accuracy (mIoU) of methods on the Occ3D-nuScenes (Tian et al. 2023) benchmark.** We follow (Hou et al. 2024) to define real-time in occupancy prediction as 10 FPS.

Through this evaluation, we find it challenging to balance accuracy and efficiency, as shown in Fig. 1. A core issue underlying this challenge is ***the strong coupling between geometry and semantics***. As illustrated in Fig. 2, the predicted geometric structure (*e.g.*, depth) impacts the feature projection of 2D image features into 3D voxel space. Inaccurate feature projection can compromise feature discriminability and negatively affect downstream semantic learning, making geometric and semantic learning inherently tightly coupled. One straightforward and rough solution is to adopt a heavy network, but this comes at the expense of high computational cost and significantly reduced efficiency. To address this coupling issue effectively and efficiently, we propose a geometric-semantic disentanglement solution that focuses on both ***model design*** and ***learning strategies***.

*These authors contributed equally.
†Corresponding authors.

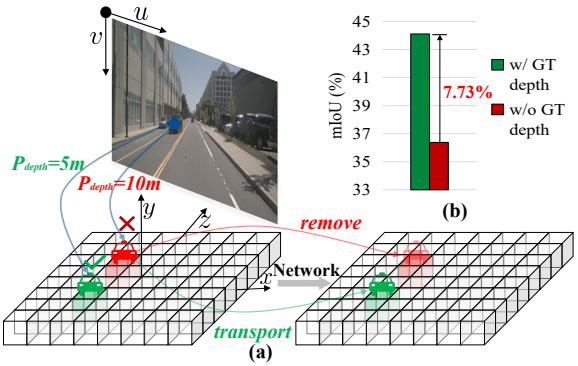

Figure 2: **Illustration of the geometric-semantic coupling problem.** (a) Incorrect predicted depth can result in inaccurate 2D-to-3D view transformation, which requires refinement and correction by the subsequent network. (b) shows the accuracy gap between using predicted depth and ground-truth depth, addressing the importance of this issue.

In terms of model design, existing methods primarily rely on heavy 3D networks (Huang and Huang 2022; Ma et al. 2024) to simultaneously refine geometric structure and learn semantics. However, the high computational cost of 3D networks is unaffordable for real-time methods. To improve efficiency, recent methods (Yu et al. 2023; Hou et al. 2024; Liu et al. 2024a) collapse 3D voxel features into BEV features, but typically fail to achieve satisfactory accuracy with a high speed, as shown in Fig. 1. To balance both accuracy and efficiency, we propose a dual-branch network that disentangles the representation of geometry and semantics. In the voxel branch, we propose a re-parameterized 3D large-kernel convolution, which refines geometric structure with an enhanced receptive field and reduces computation through re-parameterization (Ding et al. 2021, 2024). In the BEV branch, we adopt 2D BEV-level rather than 3D voxel-level temporal fusion and semantic encoding, which notably reduces computation with an acceptable accuracy reduction.

In terms of learning strategies, almost all Lift-Splat-Shoot (LSS) (Philion and Fidler 2020) based methods (Huang and Huang 2022; Yu et al. 2023; Hou et al. 2024) directly utilize the predicted depth for 2D-to-3D view transformation. However, the predicted depth is not always accurate, particularly at the early stage of training, which can exacerbate the coupling problem. To address this issue, we propose a geometric-semantic disentangled learning strategy. Specifically, we use ground-truth depth at the beginning of training to maintain an accurate geometric structure, which bypasses the negative impact of incorrect predicted depth and enables disentangled semantic learning. As the predicted depth improves during training, we gradually mix ground-truth depth with predicted depth, which enables the model to learn how to refine the predicted geometric structure. This strategy effectively reduces optimization difficulty and improves accuracy without additional computations.

Our contributions can be summarized as follows:

- We redirect the focus from accuracy only to efficiency as well, and offer a more deployment-friendly comparison by assessing speed and memory usage.
- We introduce the re-parameterization technique into occupancy prediction and propose a re-parameterized 3D large-kernel convolution to refine geometric structure with minimal computational overhead.
- We propose a novel learning strategy to disentangle the learning of geometric structure and semantic knowledge, which achieves consistent accuracy improvements across various pre-training models and methods.
- We propose a speed-accuracy balanced 3D Occupancy prediction method (GSD-Occ) via Geometric-Semantic Disentanglement, establishing a new real-time state-of-the-art in terms of mIoU and RayIoU.

Related Work

Vision-based BEV Perception. Bird's-Eye View (BEV) perception (Li et al. 2022a) has recently seen significant advancements, developing as a crucial component in autonomous driving (AD). By leveraging 2D-to-3D view transformation to project camera image features into BEV representation, multiple tasks can be integrated into a unified framework. View transformation can be broadly categorized into two types: forward projection and backward projection. The former employs explicit depth estimation to project image features into 3D space (Philion and Fidler 2020; Huang et al. 2021; Li et al. 2023b,a; Huang and Huang 2022). While, the latter initializes a BEV feature and then implicitly models depth information by querying image features using a spatial cross-attention (Zhu et al. 2020; Wang et al. 2022; Li et al. 2022b; Yang et al. 2023; Jiang et al. 2023). Although BEV perception excels in 3D object detection, it still faces challenges with corner-case and out-of-vocabulary objects, which are crucial for ensuring the safety of AD. To address this challenge, 3D occupancy prediction has been proposed, quickly emerging as a promising solution (Tesla 2021).

3D Occupancy Prediction. 3D occupancy prediction reconstructs the 3D occupancy space using continuous voxel grids. A straightforward idea is to replace the BEV representation in 3D object detection with the voxel representation, and then append a segmentation head (Huang et al. 2021; Li et al. 2022b; Tian et al. 2023). However, the 3D voxel representation incurs substantial computational costs. To address this issue, TPVFormer (Huang et al. 2023) divided the 3D space into three-view planes and recovered the voxel representation by interpolation. FB-OCC (Li et al. 2023c) adopted a hybrid of BEV-level forward and voxel-level backward view transformation. COTR (Ma et al. 2024) proposed a compact voxel representation by downsampling. PannoOcc (Wang et al. 2024) introduced a novel panoramic occupancy segmentation task and adopted sparse 3D convolutions to decrease computation. However, these works still suffer from low speed and high memory usage, which are not reported in their papers. To offer a more deployment-friendly comparison, we assess speed and memory usage across a broad range of publicly available methods.

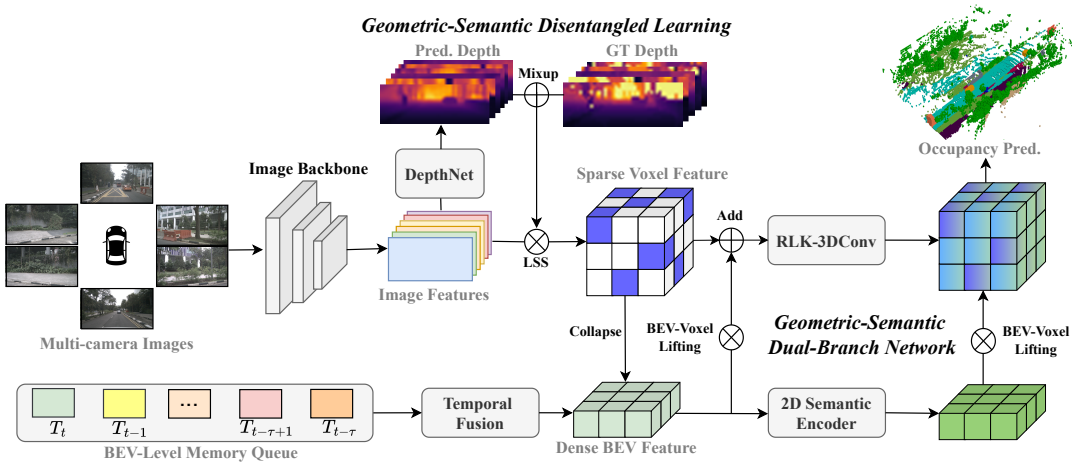

Figure 3: **The overview of GSD-Occ.** Multi-camera images are first fed into a backbone (He et al. 2016) to get image features, and DepthNet (Li et al. 2023b) is used to predict a depth distribution. The LSS (Philion and Fidler 2020) module is then employed to explicitly project 2D image features into 3D voxel space. Subsequently, Geometric-Semantic Dual-Branch Network efficiently refines geometric structure and extracts rich semantic knowledge. Geometric-Semantic Disentangled Learning strategy injects ground-truth depth into LSS to disentangle the learning of geometric structure and semantic knowledge.

Deployment-Friendly Occupancy Prediction. Only a few recent occupancy prediction works focused on deployment friendliness. SparseOcc (Liu et al. 2024b) employed a pure sparse transformer-based network. FlashOcc (Yu et al. 2023) directly used BEV features to predict geometric structure and semantic classes. FastOcc (Hou et al. 2024) and BEVoxSeg (Liu et al. 2024a) employed a hybrid BEV-Voxel representation by simply concatenating the lifted BEV features with voxel features. Different from them, we explore the potential of hybrid BEV-Voxel representation further by: (1) proposing a re-parameterized large-kernel 3D convolution to efficiently refine geometric structure, (2) adopting BEV-level rather than voxel-level temporal fusion to improve efficiency, and (3) adding multiple BEV-Voxel feature fusion paths. Moreover, we introduce a plug-and-play disentanglement strategy that can improve accuracy without incurring additional computational costs.

Method

Problem Formulation

Given a sequence of images $I_{i,t} \in \mathbb{R}^{H_i \times W_i \times 3}$ captured from N_c surround-view cameras over T frames, where $i \in \{1, ..., N_c\}$ and $t \in \{1, ..., T\}$. The camera intrinsic parameters $\{K_i\}$ and extrinsic parameters $\{[R_i \mid t_i]\}$ in each frame are also known. Vision-based 3D occupancy prediction aims to estimate the state of 3D voxels within the range $[X_s, Y_s, Z_s, X_e, Y_e, Z_e]$ around the ego vehicle. The shape of the 3D voxels is $[X, Y, Z]$ (*e.g.*, [200,200,16] in (Tian et al. 2023)), where $[\frac{X_e - X_s}{X}, \frac{Y_e - Y_s}{Y}, \frac{Z_e - Z_s}{Z}]$ is the size of a single voxel. Each voxel contains *occupancy state* ("occupied" or "empty") and specific *semantic* ("category" or "unknown"). Benefit from the learning of occupancy, 3D occupancy prediction can develop a general object representation to handle out-of-vocabulary and unusual obstacles.

Overall Architecture

The overview of GSD-Occ is shown in Fig. 3, which includes an image backbone to extract image features, a 2D-to-3D view transformation to project image features into 3D space, a geometric-semantic dual-branch network (GSDBN) to maintain geometric integrity and enhance efficiency, and a geometric-semantic disentangled learning (GSDL) strategy to reduce optimization difficulty and improve accuracy.

Image Encoder. Given a set of surround-view images at T-th frame, denoted as $I_T = \{I_{i,T} \in \mathbb{R}^{H_i \times W_i \times 3}\}_{i=1}^{N_c}$, we employ a pre-trained backbone network (*e.g.*, ResNet-50 (He et al. 2016)) to extract image features $F = \{F_i \in \mathbb{R}^{C_F \times H_F \times W_F}\}_{i=1}^{N_c}$, where $[C_F, H_F, W_F]$ is the feature size.

2D-to-3D View Transformation. Given the limited learning capacity of real-time models, we adopt an explicit view transformation module (Philion and Fidler 2020) supervised by depth. Specifically, the image feature F is first fed into the DepthNet (Li et al. 2023b) to generate a depth distribution $D = \{D_i \in \mathbb{R}^{D_{bin} \times H_F \times W_F}\}_{i=1}^{N_c}$, where D_{bin} is the number of depth bins. Meanwhile, the F is processed by a context module to get the context feature $\hat{F}$. With $\hat{F}$ and the D as input, a pseudo point cloud feature $P \in \mathbb{R}^{N_c D_{bin} H_F W_F \times C}$ can be obtained through outer product $\hat{F} \otimes D$. Then, voxel-pooling is applied to the P to obtain the voxel feature $V \in \mathbb{R}^{C \times \frac{X}{2} \times \frac{Y}{2} \times \frac{Z}{2}}$, with $2\times$ downsampling to reduce computational complexity.

Geometric-Semantic Dual-Branch Network

Geometric-Semantic Dual-Branch Network (GSDBN) employs a hybrid BEV-Voxel representation, where sparse voxel features server as *"skeleton"* to maintain 3D geometric structure and computation-efficient BEV features are

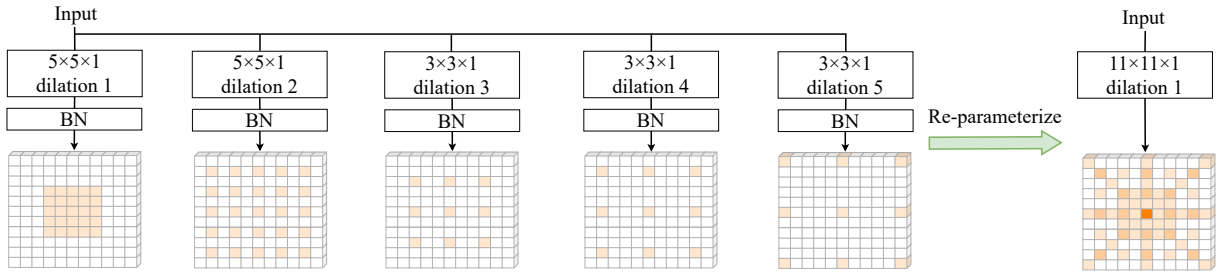

Figure 4: **Illustration of the re-parameterized large-kernel 3D convolution.** It uses parallel dilated small-kernel 3D convolutions to enhance a non-dilated large-kernel 3D convolution. An example with a kernel size of $[11, 11, 1]$ is shown.

used as *"flesh"* to complete voxel features with semantic information. As shown in Fig. 3, GSDBN adopts a dual-branch framework with a BEV-level temporal fusion module, a 2D semantic encoder, a re-parameterized large-kernel 3D convolution (RLK-3DConv), and a BEV-Voxel lifting module.

BEV-Level Temporal Fusion. To reduce computational and memory costs, we propose using BEV features instead of voxel features employed in (Li et al. 2023c; Liu et al. 2024a) for temporal fusion. Besides, we introduce a memory queue, as in (Park et al. 2022), to avoid time-consuming feature re-computation. Specifically, we collapse the voxel feature V along the height dimension to obtain the BEV feature $B \in \mathbb{R}^{C \times \frac{X}{2} \times \frac{Y}{2}}$ and maintain a memory queue of length τ to store historical BEV features. To fuse the BEV features of historical frames with the current frame, we first warp them to the current timestamp using the camera's intrinsic and extrinsic parameters. Then, we feed the aligned BEV features into 2D convolutions to obtain the temporal BEV feature $B_t \in \mathbb{R}^{C \times \frac{X}{2} \times \frac{Y}{2}}$. Due to the sparsity of voxel features after view transformation (Liu et al. 2024b), the collapsed BEV features can also retain rich semantic information, resulting in only a slight accuracy drop of 0.69% mIoU and a notable decrease in inference time by 0.025 s (see Tab. 3).

2D Semantic Encoder. We employ a light-weight 2D UNet-like (2015) encoder to extract features with rich semantic information. Specifically, the temporal BEV feature B_t is downsampled and then upsampled by a factor of 4, with residuals utilized to fuse multi-scale features. This process yields the semantic BEV feature $B_s \in \mathbb{R}^{C' \times \frac{X}{2} \times \frac{Y}{2}}$.

Re-parameterized Large-Kernel 3D Convolution. Inspired by (Ding et al. 2021, 2024), we introduce the re-parameterization technique into occupancy prediction, and propose a novel re-parameterized large-kernel 3D convolution (RLK-3DConv) to efficiently refine geometric structure. During training, we employ multiple parallel dilated small-kernel 3D convolutions along with batchnorm (BN) layers. This combination helps capture small-scale patterns and enhance the receptive filed. During inference, these parallel small-kernel 3D convolutions can be converted into a large-kernel convolution to improve efficiency.

Fig. 4 shows the re-parameterization process using an example with a convolutional kernel size of $[K_X, K_Y, K_Z] = [11, 11, 1]$. Since omitting pixels in the input is equivalent to inserting extra zero entries into the convolution, a dilated convolution with a small kernel can be converted into a non-dilated one with a sparse larger kernel (Ding et al. 2024). For a small 3D convolutional kernel $W \in \mathbb{R}^{k_x \times k_y \times k_z}$ with the dilation rate (r_x, r_y, r_z), this transformation can be elegantly implemented by a transpose 3D convolution:

$$W' = \text{conv\_transpose3d}(W, I, s = (r_x, r_y, r_z)) \quad (1)$$

where $I \in \mathbb{R}^{1 \times 1 \times 1}$ and s means the stride. Then, the sparse kernel W' and the subsequent 3D BN layer (with the parameters of accumulated mean μ, standard deviation σ, the learned scaling factor γ, and the learned bias β) can be integrated into a single convolution with a bias vector:

$$W'' = \frac{\gamma}{\sigma} W', \quad b'' = -\frac{\mu\gamma}{\sigma} + \beta. \quad (2)$$

By summing W'' and b'' across multiple parallel convolutions, we can get the final re-parameterized 3D convolution, with its kernel and bias computed as follows:

$$\hat{W} = \sum_{i=1}^{C_s} \text{zero\_padding}(W_i''), \quad \hat{b} = \sum_{i=1}^{C_s} (b_i''), \quad (3)$$

where C_s is the number of parallel convolutions and zero_padding function zero-pads W'' to the size of $[K_X, K_Y, K_Z]$. By performing the re-parameterized 3D convolution with the kernel $\hat{W}$ and the bias $\hat{b}$, we can obtain the geometric voxel feature $V_g \in \mathbb{R}^{C' \times \frac{X}{2} \times \frac{Y}{2} \times \frac{Z}{2}}$.

BEV-Voxel Lifting Module. To fuse the outputs of the BEV and the voxel branch, we employ a BEV-Voxel lifting (BVL) module that projects BEV features into voxel space. This design is inspired by LSS (Philion and Fidler 2020) and also used in a concurrent work (Liu et al. 2024a), but we add more feature fusion paths. As shown in Fig. 3, the BVL module is applied to the temporal BEV feature B_t and the semantic BEV feature B_s for two rounds of feature fusion. Taking B_s as an example, a context branch generates height-aware feature $B_s' \in \mathbb{R}^{C' \times \frac{X}{2} \times \frac{Y}{2}}$, while a height branch predicts a height distribution $H' \in \mathbb{R}^{\frac{X}{2} \times \frac{Y}{2} \times \frac{Z}{2}}$. Then, the semantic voxel feature $V_s \in \mathbb{R}^{C' \times \frac{X}{2} \times \frac{Y}{2} \times \frac{Z}{2}}$ can be obtained through the outer product $B_s' \otimes H'$. Finally, the geometric-semantic disentangled feature $V_{g\&s} \in \mathbb{R}^{C' \times X \times Y \times Z}$ can be obtained by summing the geometric voxel feature V_g and the semantic voxel feature V_s, followed by upsampling 2 $\times$ using transpose 3D convolutions: $V_{g\&s} = \text{upsample3d}(V_g + V_s)$.

Geometric-Semantic Disentangled Learning

In this section, we further address the geometric-semantic coupling problem from the perspective of learning strategies. We focus on a crucial component for 2D-to-3D view transformation, *i.e.*, the LSS module, which projects image features into 3D voxel space with the input of depth distribution. However, since the predicted depth is not always accurate, especially at the early stage of training, which can exacerbate the coupling issue and degrade accuracy.

Inspired by language models (Radford et al. 2018, 2019; Brown et al. 2020), which provide sequential ground-truth tokens to predict the next token during training, an intuitive idea is to replace the predicted depth with ground-truth depth in LSS. However, this strategy performs poorly when using the predicted depth for testing, achieving only 35.1% mIoU. This result indicates that the model does not learn how to refine the predicted geometric structure.

To ensure both accurate feature projection and the model's ability to refine the predicted geometric structure, we propose a Geometric-Semantic Disentangled Learning (GSDL) strategy. Specifically, we introduce ground-truth depth $\hat{D} = \{\hat{D}_i \in \mathbb{R}^{D_{bin} \times H_F \times W_F}\}_{i=1}^{N_c}$ to LSS at the beginning of training, so that the model can separately focus on learning semantics with accurate geometric structure. As the predicted depth improves during training, we gradually mix the ground-truth depth $\hat{D}$ with the predicted depth D. The mixup depth D^m can be obtained by conducting the arithmetic mean, using a factor $\alpha \in [0, 1]$:

$$D^m = D\alpha + \hat{D}(1 - \alpha). \tag{4}$$

The value of α is determined by a function that increases monotonically with the number of training iterations. Specifically, we first transform the range of iterations from $x \in [0, T_{max}]$ to $x \in [-N_\alpha, N_\alpha]$, where T_{max} is the maximum number of training iterations and N_α is a constant set to 5 without careful selection. We then employ a sigmoid function to smooth this mixup curve:

$$\alpha = \frac{1}{1 + e^{-rx}}, \tag{5}$$

where r is a parameter that controls the steepness of the mixup. As $\alpha \to 1$ by the end of training, the model can gain the ability to refine the predicted geometric structure.

Experiment

Experimental Setup

We evaluate our model using the Occ3D-nuScenes (Tian et al. 2023) benchmark, which is based on nuScenes (Caesar et al. 2020) dataset and constructed for the CVPR2023 3D occupancy prediction challenge. The dataset consists of 1000 videos, split into 700 for training, 150 for validation, and 150 for testing. Each key frame of video contains a 32-beam LiDAR point cloud, six RGB images from surround-view cameras, and dense voxel-wise semantic occupancy annotations. The perception range is [-40m, -40m, -1m, 40m, 40m, 5.4m], with each voxel sized [0.4m, 0.4m, 0.4m]. The voxels contain 18 categories, including 16 known object classes, an unknown object class labeled as "others", and an "empty" class. We use the mean intersection over union (mIoU) and RayIoU (Liu et al. 2024b) across all classes to evaluate the accuracy of methods.

Implementation Details

Adhering to common practices (Li et al. 2023c; Liu et al. 2024b; Ma et al. 2024), we adopt ResNet-50 (He et al. 2016) as the image backbone. We maintain a memory queue of length 15 to store historical features. For RLK-3DConv, we set the size of convolution kernel to [11, 11, 1]. The steepness parameter r is set to 5 in geometric-semantic disentangled learning. During training, we use a batch size of 32 on 8 Nvida A100 GPUs. Unless otherwise specified, all models are trained for 24 epochs using the AdamW optimizer (Loshchilov, Hutter et al. 2017) with a learning rate 1×10^{-4} and a weight decay of 0.05. During inference, we use a batch size of 1 on a single Nvidia A100 GPU. The FPS and memory metrics are tested using the mmdetection3d codebase (Contributors 2020).

Main Results

In Tab. 1 and Fig. 1, we compare GSD-Occ with previous state-of-the-art (SOTA) methods on the validation split of Occ3D-nuScenes. GSD-Occ demonstrates high speed and low memory usage while achieving accuracy comparable to or better than many non-real-time methods, such as BEVFormer (Li et al. 2022b), BEVDet4D (Huang and Huang 2022), SurroundOcc (Wei et al. 2023), and FlashOCC (Yu et al. 2023). When compared with FB-OCC (Li et al. 2023c), the winner of CVPR 2023 occupancy challenge, GSD-Occ is $\sim$ 3$\times$ faster and shows a 1.9% mIoU improvement. Compared to recent SOTA methods in accuracy, GSD-Occ* achieves only 0.4% lower mIoU than PannoOcc (Wang et al. 2024), but it is $\sim$ 3$\times$ faster and uses only $\sim$ 50% of memory usage. Compared to other real-time methods, GSD-Occ achieves a notable 5.2% higher mIoU than FastOCC (Hou et al. 2024) with even higher speed. When the visible mask is unavailable, GSD-Occ also achieves high accuracy, outperforming SparseOcc (Liu et al. 2024b) by 1.2% mIoU. Moreover, we also report the RayIoU metric recently proposed by (Liu et al. 2024b) in Tab. 2. GSD-Occ achieves 4.9% higher RayIoU with higher speed and lower memory usage when compared with the recent SOTA method, SparseOcc. These results highlight the effectiveness of the proposed geometric-semantic disentanglement solution.

We further provide qualitative results in Fig. 5. Despite significantly reducing computation, GSD-Occ can also effectively perceive geometric details (Row 1 and Row 2) and accurate semantics (Row 3). Additionally, GSD-Occ also performs well under night conditions (Row 4).

Ablation Studies

Ablations on GSDBN. The results are shown in Tab. 3, we can observe that each component of GSDBN contributes to overall performance. The baseline model, which lacks temporal fusion as well as both 2D and 3D encoder, achieves highest speed but falls short in accuracy. Although applying voxel features for temporal fusion improves mIoU by

Method	Venue	Backbone	Image Size	Visible Mask	mIoU (%)	FPS	Memory (MB)
MonoScene (2022)	CVPR'22	ResNet-101	928 × 600	✘	6.1	-	-
OccFormer (2023)	ICCV'23	ResNet-50	256 × 704	✘	20.4	4.8	7617
CTF-Occ (2023)	arXiv'23	ResNet-101	928 × 600	✘	28.5	-	-
SparseOcc (2024b)	ECCV'24	ResNet-50	256 × 704	✘	30.6	17.7	6883
GSD-Occ (Ours)	-	ResNet-50	256 × 704	✘	**31.8**	**20.0**	**4759**
BEVFormer (2022b)	ECCV'22	ResNet-101	900 × 1600	✔	39.3	4.7	6651
BEVDet4D (2022)	arXiv'22	ResNet-50	256 × 704	✔	39.2	0.8	6053
SurroundOcc (2023)	ICCV'23	ResNet-101	900 × 1600	✔	37.1	3.2	5491
FB-OCC (2023c)	ICCV'23	ResNet-50	256 × 704	✔	37.5	7.0	5467
FlashOCC (2023)	arXiv'24	ResNet-50	256 × 704	✔	37.8	4.8	**3143**
BEVoxSeg (2024a)	ICASSP'24	ResNet-50	-	✔	40.4	**9.3**	-
PanoOcc (2024)	CVPR'24	ResNet-101	900 × 1600	✔	42.1	3.0	11991
COTR (2024)	CVPR'24	ResNet-50	256 × 704	✔	**44.5**	0.9	10453
FastOCC (2024)	ICRA'24	ResNet-50	320 × 800	✔	34.2	15.9	-
FastOCC* (2024)	ICRA'24	ResNet-101	320 × 800	✔	37.2	10.7	-
GSD-Occ (Ours)	-	ResNet-50	256 × 704	✔	39.4	**20.0**	**4759**
GSD-Occ* (Ours)	-	ResNet-50	512 × 1408	✔	**41.7**	10.0	5185

Table 1: **3D Occupancy prediction performance on the Occ3D-nuScenes benchmark.** The FPS of all methods are evaluated on an Nvidia A100 GPU, except for FastOCC, which is reported using an Nvidia V100 GPU in its paper. Visible Mask refers to whether models are trained with visible masks. We follow (Hou et al. 2024) to define real-time in occupancy prediction as 10 FPS, and individually highlight real-time methods by employing gray areas.

Method	Venue	Backbone	Image Size	Epoch	RayIoU (%)	FPS	Memory (MB)
BEVFormer (2022b)	ECCV'22	ResNet-101	900 × 1600	24	32.4	4.7	6651
BEVDet4D (2022)	arXiv'22	ResNet-50	256 × 704	90	29.6	0.8	6053
FB-OCC (2023c)	ICCV'23	ResNet-50	256 × 704	90	33.5	7.0	5467
SparseOcc (2024b)	ECCV'24	ResNet-50	256 × 704	24	34.0	17.7	6883
GSD-Occ (Ours)	-	ResNet-50	256 × 704	24	**38.9**	**20.0**	**4759**

Table 2: **3D Occupancy prediction performance on the Occ3D-nuScenes benchmark using the RayIoU metric proposed by (Liu et al. 2024b).** The FPS of all methods are evaluated on an Nvidia A100 GPU.

2D Encoder	Temporal Fusion	LK-3DConv	RLK-3DConv	mIoU (%)	FPS
✘	✘	✘	✘	35.11	27.0
✔	✘	✘	✘	36.38	23.1
✔	3D	✘	✘	39.09	13.9
✔	2D	✘	✘	38.40	21.4
✔	2D	✔	✘	38.10	20.0
✔	2D	✘	✔	38.90	20.0

Table 3: **Ablation study on each component of GSDBN.** "3D" and "2D" denote employing voxel and BEV features, respectively. LK-3DConv means the large-kernel 3D convolution without re-parameterization.

0.69% compared with using BEV features, it introduces a notable inference delay of 0.025 s, making the accuracy gain costly. Adding RLK-3DConv provides a notable 0.5% mIoU improvement with only a minimal inference delay of 0.003 s. Notably, directly using large-kernel 3D convolution (LK-3DConv) instead of RLK-3DConv degrades the accuracy, which indicates that roughly increasing the receptive field can harm feature representation. In RLK-3DConv, multiple parallel small-kernel convolutions are employed to capture fine-grained patterns, which helps improve accuracy.

Method	Pretrained model	GSDL	mIoU (%)
FB-OCC (Li et al. 2023c)	BEVDepth	✘	37.5
FB-OCC (Li et al. 2023c)	BEVDepth	✔	37.82 (+0.32)
GSD-Occ	ImageNet	✘	36.48
GSD-Occ	ImageNet	✔	36.88 (+0.40)
GSD-Occ	BEVDepth	✘	38.90
GSD-Occ	BEVDepth	✔	39.45 (+0.55)

Table 4: **Effectivenss of GSDL.** ImageNet and BEVDepth are the model weights in (Deng et al. 2009; Li et al. 2023b).

Ablations on GSDL. In Tab. 4, we show the results of applying the plug-and-play GSDL strategy to different pretraining models and methods. Without incurring additional computation costs, GSDL achieves consistent accuracy improvement (around 0.3%-0.5% mIoU). It highlights the effectivenss and generalizability of GSDL.

The Effectiveness of BVL. We compare BVL module with the other existing lifting methods as shown in Tab. 5, it shows that BVL module achieves the best performance in both accuracy and speed, proving its effectiveness.

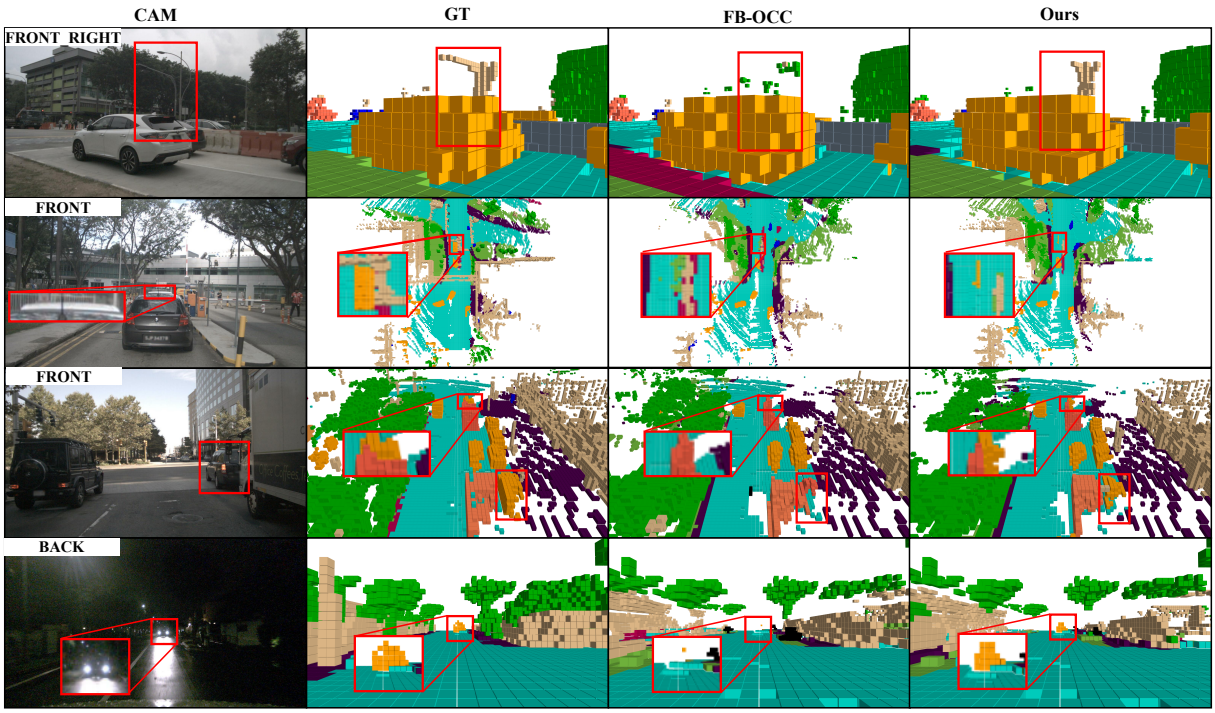

Figure 5: **Qualitative results comparison between FB-OCC and our method.** The results show that our method can construct more detailed geometry (Row 1 and Row 2), more accurate semantics (Row 3), and stronger adaptability in night (Row 4).

Lifting Method	mIoU (%)	FPS
Channel-to-Height (Yu et al. 2023)	38.62	18.7
Repeat + 3D Conv (Hou et al. 2024)	38.42	18.3
BVL	39.45	20.0

Table 5: **The effectivenss of BVL.**

Kernel Size	mIoU (%)	FPS
$7 \times 7 \times 1$	38.67	19.4/20.4
$9 \times 9 \times 1$	38.70	19.0/20.2
$11 \times 11 \times 1$	38.90	18.6/20.0
$13 \times 13 \times 1$	38.65	18.2/19.6
$15 \times 15 \times 1$	38.74	17.9/19.4

Table 6: **The impact of kernel sizes in RLK-3DConv.** "-/-" denotes the FPS of before and after re-parameterization.

Is a Larger 3D Convolutional Kernel Better? In Tab. 6, we present the results of different kernel sizes in RLK-3DConv. Adopting a kernel size of $[11 \times 11 \times 1]$ achieves the highest accuracy, suggesting that refining geometric structure benefits from a relatively large receptive field, but excessively large kernels can be counterproductive. Besides, thanks to the re-parameterization technique, the inference speed has significantly improved from 18.6 to 20.0 FPS.

Smooth or Steep Mixup of Prediction and Ground-Truth Depth? As shown in Tab. 7, we repeatedly run experiments to explore the impact of various steepness values r in Eq. 5. The best accuracy is achieved when r is set to 5, indicating that overly smooth mixup curves may hinder the model's ability to adapt to predicted depth, while excessively steep curves can increase the optimization difficulty.

Steepness r	mIoU (%)
3	39.02 ± 0.08
4	39.05 ± 0.02
5	39.39 ± 0.05
6	39.25 ± 0.02
7	39.22 ± 0.11

Table 7: **Ablation study on the steepness (*i.e.*, r in Eq. 5).**

Conclusion

In this paper, we introduce GSD-Occ, a speed-accuracy balanced 3D occupancy prediction method via geometric-semantic disentanglement. We design a dual-branch network featuring a novel re-parameterized 3D large-kernel convolution and BEV-level temporal fusion, which efficiently disentangles the representation of geometry and semantics. Moreover, we propose to disentangle the learning of geometry and semantic by injecting ground-truth, which further enhances accuracy without adding computational overhead. We validate the effectiveness of GSD-Occ on Occ3D-nuScenes, where it sets a new real-time state-of-the-art performance.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (No. 62476280), the Science and Technology Innovation Program of Hunan Province of China (No. 2024RC3137), the Natural Science Foundation of Hunan Province of China (No. 2022JJ30666), and the National Key Research and Development Program of China (No. 2018YFB0204301).

References

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Cao, A.-Q.; and de Charette, R. 2022. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3991–4001.

Contributors, M. 2020. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; and Sun, J. 2021. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13733–13742.

Ding, X.; Zhang, Y.; Ge, Y.; Zhao, S.; Song, L.; Yue, X.; and Shan, Y. 2024. Unireplknet: A universal perception large-kernel convnet for audio, video, point cloud, time-series and image recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, J.; Li, X.; Guan, W.; Zhang, G.; Feng, D.; Du, Y.; Xue, X.; and Pu, J. 2024. FastOcc: Accelerating 3D Occupancy Prediction by Fusing the 2D Bird's-Eye View and Perspective View. *IEEE International Conference on Robotics and Automation (ICRA)*.

Huang, J.; and Huang, G. 2022. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*.

Huang, J.; Huang, G.; Zhu, Z.; Ye, Y.; and Du, D. 2021. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*.

Huang, Y.; Zheng, W.; Zhang, Y.; Zhou, J.; and Lu, J. 2023. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9223–9232.

Jiang, Y.; Zhang, L.; Miao, Z.; Zhu, X.; Gao, J.; Hu, W.; and Jiang, Y.-G. 2023. Polarformer: Multi-camera 3d object detection with polar transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, 1042–1050.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12697–12705.

Li, H.; Sima, C.; Dai, J.; Wang, W.; Lu, L.; Wang, H.; Xie, E.; Li, Z.; Deng, H.; Tian, H.; et al. 2022a. Delving into the Devils of Bird's-eye-view Perception: A Review, Evaluation and Recipe. *arXiv preprint arXiv:2209.05324*.

Li, Y.; Bao, H.; Ge, Z.; Yang, J.; Sun, J.; and Li, Z. 2023a. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, 1486–1494.

Li, Y.; Ge, Z.; Yu, G.; Yang, J.; Wang, Z.; Shi, Y.; Sun, J.; and Li, Z. 2023b. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, 1477–1485.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; and Dai, J. 2022b. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision (ECCV)*, 1–18. Springer.

Li, Z.; Yu, Z.; Wang, W.; Anandkumar, A.; Lu, T.; and Alvarez, J. M. 2023c. Fb-bev: Bev representation from forward-backward view transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 6919–6928.

Liang, T.; Xie, H.; Yu, K.; Xia, Z.; Lin, Z.; Wang, Y.; Tang, T.; Wang, B.; and Tang, Z. 2022. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems (NIPS)*, 35: 10421–10434.

Liu, H.; Wang, B.; Zhang, L.; Ji, J.; and Zhang, Y. 2024a. BEVoxSeg: BEV-Voxel Representation for Fast and Accurate Camera-Based 3D Segmentation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3345–3349. IEEE.

Liu, H.; Wang, H.; Chen, Y.; Yang, Z.; Zeng, J.; Chen, L.; and Wang, L. 2024b. Fully sparse 3d panoptic occupancy prediction. *European conference on computer vision (ECCV)*.

Loshchilov, I.; Hutter, F.; et al. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5.

Ma, Q.; Tan, X.; Qu, Y.; Ma, L.; Zhang, Z.; and Xie, Y. 2024. COTR: Compact Occupancy TRansformer for Vision-based

3D Occupancy Prediction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Park, J.; Xu, C.; Yang, S.; Keutzer, K.; Kitani, K. M.; Tomizuka, M.; and Zhan, W. 2022. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. In *The Eleventh International Conference on Learning Representations*.

Philion, J.; and Fidler, S. 2020. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European conference on computer vision (ECCV)*, 194–210. Springer.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer.

Tesla. 2021. Tesla AI Day. https://www.youtube.com/watch?v=j0z4FweCy4M.

Tian, X.; Jiang, T.; Yun, L.; Wang, Y.; Wang, Y.; and Zhao, H. 2023. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *arXiv preprint arXiv:2304.14365*.

Wang, Y.; Chen, Y.; Liao, X.; Fan, L.; and Zhang, Z. 2024. Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, 180–191. PMLR.

Wei, Y.; Zhao, L.; Zheng, W.; Zhu, Z.; Zhou, J.; and Lu, J. 2023. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 21729–21740.

Yang, C.; Chen, Y.; Tian, H.; Tao, C.; Zhu, X.; Zhang, Z.; Huang, G.; Li, H.; Qiao, Y.; Lu, L.; et al. 2023. BEVFormer v2: Adapting Modern Image Backbones to Bird's-Eye-View Recognition via Perspective Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17830–17839.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11784–11793.

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

Zhang, Y.; Zhu, Z.; and Du, D. 2023. OccFormer: Dual-path Transformer for Vision-based 3D Semantic Occupancy Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9433–9443.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.