

Real-Time 3D Occupancy Prediction via Geometric-Semantic Disentanglement

Yulin He¹ Wei Chen¹ Tianci Xun¹ Yusong Tan¹

¹National University of Defense Technology

Abstract

Occupancy prediction plays a pivotal role in autonomous driving (AD) due to the fine-grained geometric perception and general object recognition capabilities. However, existing methods often incur high computational costs, which contradicts the real-time demands of AD. To this end, we first evaluate the speed and memory usage of most public available methods, aiming to redirect the focus from solely prioritizing accuracy to also considering efficiency. We then identify a core challenge in achieving both fast and accurate performance: **the strong coupling between geometry and semantic**. To address this issue, 1) we propose a Geometric-Semantic Dual-Branch Network (GSDBN) with a hybrid BEV-Voxel representation. In the BEV branch, a BEV-level temporal fusion module and a U-Net encoder is introduced to extract dense semantic features. In the voxel branch, a large-kernel re-parameterized 3D convolution is proposed to refine sparse 3D geometry and reduce computation. Moreover, we propose a novel BEV-Voxel lifting module that projects BEV features into voxel space for feature fusion of the two branches. In addition to the network design, 2) we also propose a Geometric-Semantic Decoupled Learning (GSDL) strategy. This strategy initially learns semantics with accurate geometry using ground-truth depth, and then gradually mixes predicted depth to adapt the model to the predicted geometry. Extensive experiments on the widely-used Occ3D-nuScenes benchmark demonstrate the superiority of our method, which achieves a 39.4 mIoU with 20.0 FPS. This result is $\sim 3\times$ faster and +1.9 mIoU higher compared to FB-OCC, the winner of CVPR2023 3D Occupancy Prediction Challenge. Our code will be made open-source.

1. Introduction

Vision-based occupancy prediction [1] leverages surround-view camera images of ego vehicle to estimate object occupancy and semantics within a voxel space [13, 21, 23, 25, 32, 34, 35]. Compared to 3D object detection [15, 22, 37], it offers finer-grained 3D scene perception and produces a LiDAR-free alternative. Besides, by determining object

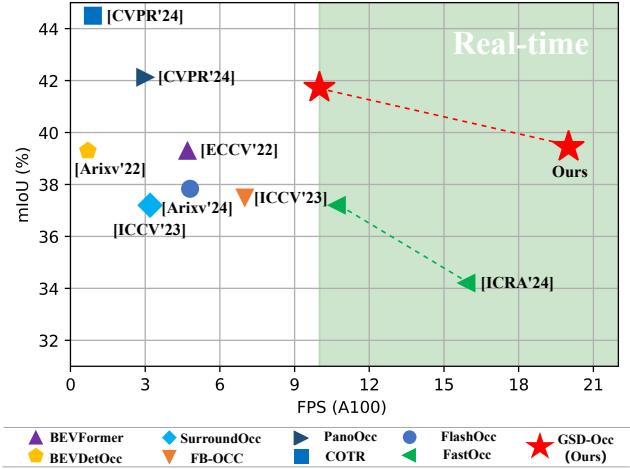


Figure 1. The inference speed (FPS) and accuracy (mIoU) of occupancy prediction methods on the Occ3D-nuScenes [32] benchmark. GSD-Occ has a clear advantage of accuracy in real-time.

presence within grid cells, occupancy prediction models can identify general objects, effectively handling out-of-vocabulary and unusual obstacles.

Despite these strengths, existing methods [4, 17, 19, 20, 39] often suffer from low inference speed (e.g., 1 ~ 3 FPS on Nvidia A100 [11, 25, 34]) and high memory usage (e.g., > 10,000 MB [25, 34]) due to the high computational cost of 3D voxel features. These limitations hinder their application in AD vehicles equipped with on-board GPUs. To redirect the focus from solely prioritizing accuracy to also considering deployment friendliness, we conduct an extensive evaluation of the speed and memory usage for most public available methods.

Through an extensive review and evaluation of existing methods, we identify a core challenge in achieving both fast and accurate performance: **the strong coupling between geometry and semantic**. As shown in Fig. 2, the geometric prediction (depth) serves as the input of the 2D-to-3D feature projection and impacts the downstream semantic classification. Therefore, inaccurate prediction depth can destroy the discriminative power of features and increases optimization difficulty. To address this issue, we propose to decouple

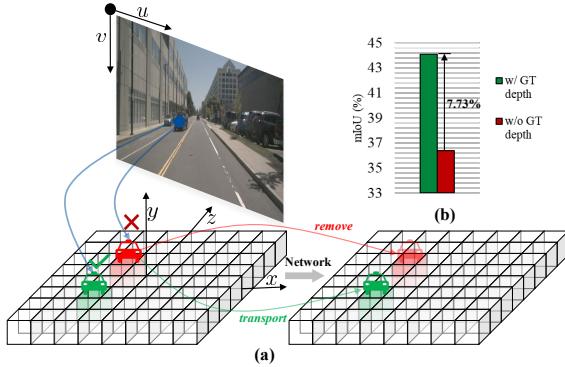


Figure 2. Illustration of the geometric-semantic coupling problem. (a) Incorrect prediction depth can result in inaccurate 2D-to-3D feature projection, which requires refinement and correction by the subsequent network. (b) illustrates the performance gap between using prediction depth and ground-truth depth, which further underscores the importance of addressing this issue.

geometric and semantic learning from both network design and learning strategy two perspectives.

As for the network design, existing methods primarily rely on *heavy* 3D networks [11, 25] to simultaneously refine geometric structure and learn semantic knowledge. However, the high computational cost of 3D networks is unaffordable for real-time methods. Recently, several works [10, 38] collapse 3D voxel features into BEV features to improve efficiency, but they often fail to achieve satisfactory accuracy (*e.g.*, FastOcc [10] in Fig. 1), as the BEV representation loses height information [25]. Therefore, it is both natural and promising to adopt a hybrid BEV-Voxel representation, which combines the strengths of computational efficiency in BEV representation and geometric integrity in voxel representation. To this end, we propose a Geometric-Semantic Dual-Branch Network (GSDBN) guided by two principles: ***sparse geometry*** and ***dense semantics***. In the BEV branch, we employ BEV-level temporal fusion and a 2D semantic encoder with U-Net [30] structure to extract dense semantic features. In the voxel branch, we propose a 3D geometric encoder with a re-parameterized 3D large-kernel convolution, which refines the sparse geometric structure with enhanced receptive field and reduces computation through the re-parameterization technique. To fuse the features of two branches, we propose a BEV-Voxel lifting module, which projects BEV-level semantic features into the voxel space along the height dimension, thus effectively recovering the lost height information.

As for the learning strategy, followed by Lift-Splat-Shoot (LSS) [27], almost all existing methods [10, 11, 25, 38] directly utilize the prediction depth for 2D-to-3D view transformation. However, they overlook that the prediction depth is not always accurate, especially at the early stage of training, which exacerbates the coupling problem

and leads to unstable optimization. Inspired by language models [2, 28, 29], which provide sequential ground-truth tokens to predict the next token, we replace the prediction depth with the ground-truth depth for 2D-to-3D view transformation during training. However, this strategy performs poorly when using the prediction depth for testing, as the model is not adapted to the prediction depth and cannot correct errors in the predicted geometry. To this end, we introduce a Geometric-Semantic Decoupled Learning (GSDL) strategy. Initially, we use ground-truth depth for 2D-to-3D view transformation to maintain accurate geometric structure, allowing for isolated semantic learning. Gradually, we mix the ground-truth depth with the prediction depth, which enables the model to learn to refine the predicted geometry. By decoupling the learning of geometric refinement and semantic knowledge, we effectively reduce the optimization difficulty and achieve further accuracy improvements without incurring additional deployment costs.

Our contributions can be summarized as follows:

- We conduct an extensive evaluation of speed and memory usage for most public available methods, which aims to redirect the focus from solely prioritizing accuracy to also considering deployment friendliness.
- We propose a dual-branch network with a hybrid BEV-Voxel representation, which separates the learning of sparse geometry and dense semantics, ensuring both computational efficiency and geometric integrity.
- We propose a novel learning strategy to decouple the learning of geometric refinement and semantic knowledge, which achieves consistent accuracy improvements across various pre-training models and methods.
- We propose GSD-Occ, a Geometric-Semantic Disentangled Occupancy predictor, which establishes a new state-of-the-art with 39.4 mIoU and 20.0 FPS for real-time occupancy prediction.

2. Related works

Vision-based BEV Perception. Bird’s-Eye View (BEV) perception [16] has recently seen significant advancements, developing as a crucial component in autonomous driving (AD) due to its computational efficiency and rich visual semantics. By leveraging 2D-to-3D view transformation to project camera image features into the 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 [11, 12, 17, 18, 27]. In contrast, the latter first initializes a BEV space and then implicitly models depth information by querying image features using a spatial cross-attention [14, 20, 33, 36, 40]. Although BEV perception excels in 3D object detection, it still struggle with corner-case and out-of-vocabulary objects, which are

crucial for ensuring the safety of autonomous driving. To address this issue, 3D occupancy prediction has been proposed, quickly emerging as a promising solution in AD [1].

3D Occupancy Prediction. 3D occupancy prediction reconstructs the 3D space using continuous voxel grids, which offers an enhanced geometry information and capability in detecting general objects. A straightforward idea is to replace the BEV representation of 3D object detection methods with the voxel representation, and then append a segmentation head [12, 20, 32]. However, voxel representations incur substantial computational and memory costs compared to BEV. To address this, TPVFormer [13] divided the 3D space into three-view planes for feature extraction, followed by interpolation to recover voxel representations. SurroundOcc [35] and CTF-Occ [32] utilized multi-scale encoders to gradually enhance voxel representations. FB-OCC [21] adapt a hybrid of forward and backward view transformation to complete sparse voxel features. COTR [25] proposes a compact voxel representation through downsampling, yet its feature enhancement network is so heavy that slows down the runtime significantly. PannoOcc [34] introduced a novel panoramic segmentation task based on occupancy representation and adapt sparse 3D convolutions to decrease computation. Despite progress in accuracy, existing methods often suffer from speed and memory limitations. Therefore, we establish a benchmark that incorporates speed and memory usage to provide a more comprehensive and fair assessment of methods.

Deployment-Friendly Occupancy Prediction. Recently, several works have focused on the deployment friendliness of occupancy prediction. For example, FlashOcc [38] directly uses a BEV representation to predict geometry and semantic, thereby reducing computational costs. Similarly, FastOcc [10] employed a BEV representation but enhanced it using a residual structure that integrates voxel features obtained from view transformation. SparseOcc [23] employed a pure sparse transformer-based network to reduce computation. However, these methods typically evaluate the speed or memory usage of only a limited set of methods. To establish a comprehensive and fair evaluation benchmark, this work evaluates most public available methods using a same experimental environment. Moreover, while existing methods significantly improve efficiency, they often fail to achieve satisfactory accuracy in real-time. This work addresses this limitation by decoupling the learning of geometry and semantic, thereby achieving both real-time and accurate performance.

3. Method

3.1. Problem Formulation

Given a sequence of images $I_{i,t} \in \mathbb{R}^{H_i \times W_i \times 3}$ from N_c surround-view cameras over T frames, where $i \in$

$\{1, \dots, N_c\}$ and $t \in \{1, \dots, T\}$. The camera intrinsic parameters $\{K_i\}$ and extrinsic parameters $\{[R_i | t_i]\}$ 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 [32]), where $[\frac{X_e - X_s}{X}, \frac{Y_e - Y_s}{Y}, \frac{Z_e - Z_s}{Z}]$ is the size of each voxel. Each voxel contains *occupancy state* (“occupied” or “empty”) and specific *semantic information* (“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.

3.2. Overall Architecture

The overview of the Geometric-Semantic Disentangled Occupancy predictor (GSD-Occ) is shown in Fig. 3, which includes an image encoder to extract image features, a 2D-to-3D view transformation to project image features into 3D space, a geometric-semantic dual-branch network (Sec. 3.3) to efficiently maintain geometric integrity and extract rich semantics, and a geometric-semantic decoupled learning strategy (Sec. 3.4) to further enhance the ability of geometric refinement and semantic learning.

Image Encoder. Given a set of surround-view camera 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 [9]) to extract image features $F = \{F_i \in \mathbb{R}^{C_F \times H_F \times W_F}\}_{i=1}^{N_c}$, where $[H_i, W_i, 3]$ and $[C_F, H_F, W_F]$ are the shapes of images and features, respectively. N_c is the number of cameras on the ego-vehicle.

2D-to-3D View Transformation. 2D-to-3D view transformation aims to convert 2D image features F to voxel representation. Given the limited learning capacity of real-time models, we adopt an explicit view transformation module [27] supervised by depth. Specifically, the image features F are first fed into the DepthNet [18] to generate a predicted 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. With F and the D as input, a pseudo point cloud feature $P \in \mathbb{R}^{N_c D_{bin} H_F W_F \times C_p}$ can be obtained through outer product $F \otimes D$. Finally, voxel-pooling is applied to the P to obtain the voxel features $V \in \mathbb{R}^{C \times \frac{X}{2} \times \frac{Y}{2} \times \frac{Z}{2}}$, with $2 \times$ downsampling to reduce computational complexity.

3.3. Geometric-Semantic Dual-Branch Network

The key idea behind Geometric-Semantic Dual-Branch Network (GSDBN) module is to employ a hybrid BEV-Voxel representation, where sparse voxel features serve as “skeleton” to maintain 3D geometric information and computation-efficient BEV features are used as “flesh” to complete voxel features with semantic information. We first elaborate the two principles for the design of GSDBN *i.e.*, **sparse geometry** and **dense semantic**. (1) Sparse geome-

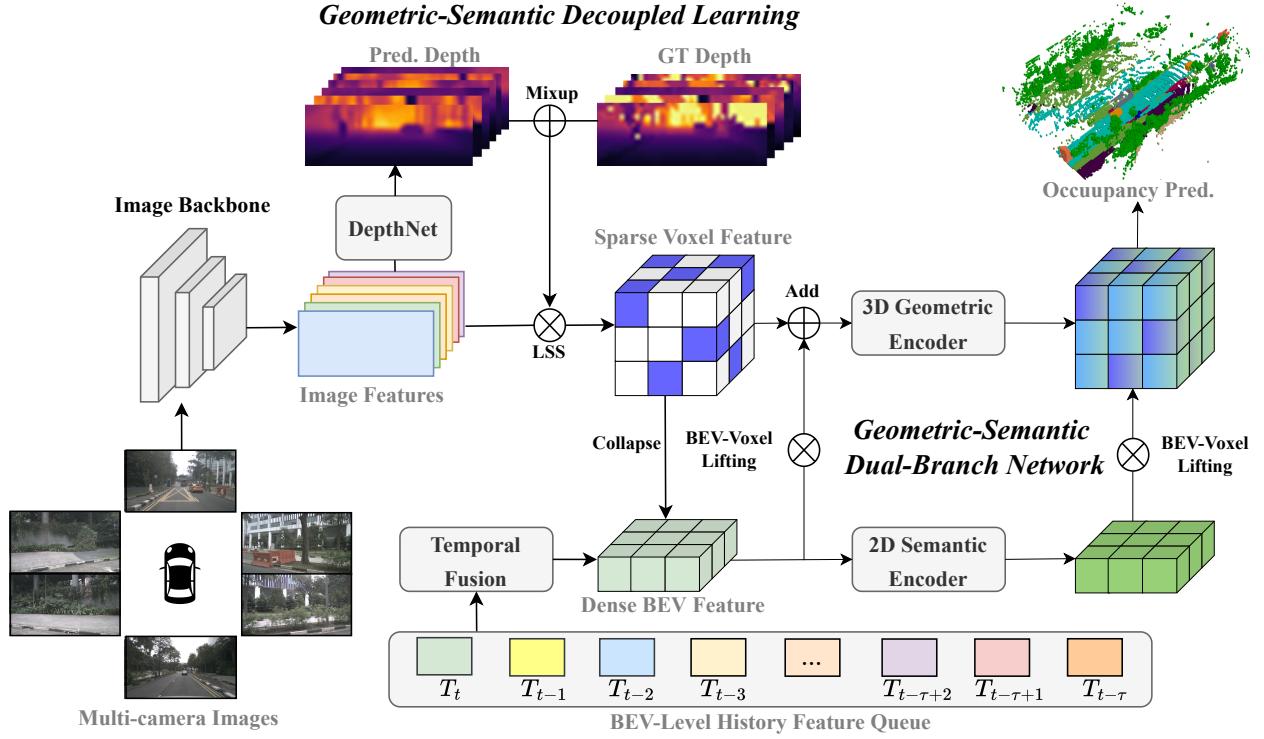


Figure 3. The overview of GSD-Occ. Multi-camera images are first fed into an image backbone network to get image features, and DepthNet [18] is used to predict a depth distribution. The Lift-Splat-Shoot (LSS) [27] module is then employed to explicitly transform 2D image features into 3D voxel features. Subsequently, the geometric-semantic dual-branch network exploits a hybrid BEV-Voxel representation to efficiently maintain geometric structure while extracting rich semantics. The geometric-semantic decoupled learning strategy injects ground-truth depth into LSS to separate the learning of geometric correction and semantic knowledge, thereby further improving accuracy.

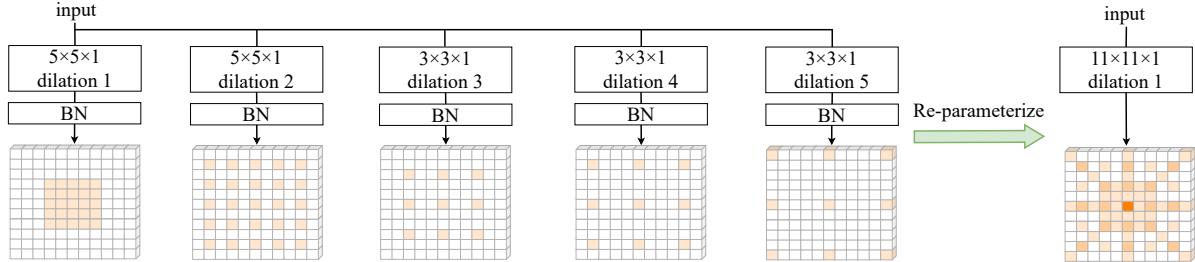


Figure 4. Illustration of the large-kernel 3D convolutional re-parameterization technique in 3D geometric encoder. It uses parallel dilated small-kernel 3D convolutions to enhance a non-dilated large-kernel 3D convolution. This example shows $[K_H, K_W, K_Z] = [11, 11, 1]$.

try in 3D occupancy grids reflects the discretization of the physical world, which leads to the sparsity of voxel features, with over 35% of values being zero after the 2D-to-3D view transformation. (2) Dense semantic, on the other hand, is necessary to maintain the model’s classification ability, as excessive zero values can severely degrade performance. Then, we detail GSDBN based on the two key principles.

3.3.1 Semantic BEV Branch

BEV-Level Temporal Fusion. To reduce computation and memory costs, we propose using BEV features in-

stead of voxel features employed in [21] for temporal fusion. Besides, we introduce a history feature queue as in [26] to avoid time-consuming and redundant feature re-computation in [11, 25, 38]. 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 the historical BEV features. To fuse the BEV features of the historical τ frames with the current frame, we first warp them to the current timestamp T and then feed them into 2D convolutions to obtain the temporal BEV features $B_t \in \mathbb{R}^{C \times \frac{X}{2} \times \frac{Y}{2}}$. The sparsity of voxel features enable BEV features to retain rich informa-

tion, resulting in an acceptable accuracy degradation (0.69 mIoU) and a notable decrease in inference time (0.025 s).

2D Semantic Encoder. We employ a light-weight 2D UNet-like [31] 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 features $B_s \in \mathbb{R}^{C' \times \frac{X}{2} \times \frac{Y}{2}}$.

3.3.2 Geometric Voxel Branch

3D Geometric Encoder. Inspired by [7, 8], we extend re-parameterization technique to 3D occupancy prediction by designing a large-kernel re-parameterized 3D convolution for geometric encoding. By this way, we can enhance the receptive field of voxel features to refine the geometric structure, while the re-parameterization technique significantly reduces inference time.

During training, we employ a non-dilated small-kernel and multiple dilated small-kernel 3D convolutions along with batchnorm (BN) layers. This combination helps capture small-scale patterns and enhance the receptive field. During inference, these parallel small-kernel 3D convolutions can be converted into a large-kernel convolution to improve efficiency.

As illustrated in Fig 4, we show a case of a 3D convolutional kernel with size $[K_X, K_Y, K_Z]$ equals to $[11, 11, 1]$. 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 equivalently converted into a non-dilated one with a sparse larger kernel [8]. 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 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 converted into a convolution with a bias vector:

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

The weight and bias of the final large kernel can be obtained by summing W'' and b'' across multiple parallel convolutions:

$$\hat{W} = \sum_{i=1}^{C_s} \text{zero_padding}(W''_i), \quad \hat{b} = \sum_{i=1}^{C_s} (b''_i), \quad (3)$$

where C_s is the number of small-kernel convolutions and zero-padding is the zero-padding function that pads W'' to

the size of large kernel $[K_X, K_Y, K_Z]$. Finally, the geometric voxel features $V_g \in \mathbb{R}^{C' \times \frac{X}{2} \times \frac{Y}{2} \times \frac{Z}{2}}$ are obtained by performing the 3D convolution with the weight \hat{W} and bias \hat{b} of the large kernel.

BEV-Voxel Lifting Module. To fuse the output of BEV and voxel branches, we propose a BEV-Voxel lifting (BVL) module that projects BEV features into voxel space. This design is inspired by LSS [27], but it projects BEV features along the height dimension instead of image features along the depth dimension. 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 example, using B_s as input, a context branch generates height-aware features $B'_s \in \mathbb{R}^{C' \times \frac{X}{2} \times \frac{Y}{2}}$, 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 features $V_s \in \mathbb{R}^{C' \times \frac{X}{2} \times \frac{Y}{2} \times \frac{Z}{2}}$ are then obtained through the outer product $B_s \otimes H'$. Finally, the geometric-semantic decoupled features $V_{g\&s} \in \mathbb{R}^{C' \times X \times Y \times Z}$ are 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: $V_{g\&s} = \text{upsample}(V_g + V_s)$.

3.4. Geometric-Semantic Decoupled Learning

In Sec. 3.3, the GSDBN module effectively mitigates the coupling problem between geometry and semantic through a dual-branch network design. In this section, we further think about this issue from a learning perspective. We focus on a key component for 2D-to-3D view transformation, *i.e.*, the LSS module, which projects image features into voxel space by predicting a depth distribution. However, as the prediction depth is not always accurate, especially at the early stage of training, which would exacerbate the coupling problem and lead to unstable optimization.

An intuitive idea is to directly replace the prediction depth with the ground-truth depth during training in LSS, while using the prediction depth in inference. This strategy is inspired by language models [2, 28, 29], where sequential ground-truth tokens are provided to predict the next token during training, but complete sentences are predicted in inference. However, this strategy performs poorly because the model does not learn how to refine the predicted geometry.

To address this issue, we propose a geometric-semantic decoupled 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 ground-truth geometry. Subsequently, we gradually mix the ground-truth depth \hat{D} with the prediction depth D during training to adapt the model to the predicted geometry. The mixup depth D^m can be obtained by conducting the arith-

metic mean, using a factor $\alpha \in [0, 1]$:

$$\begin{aligned} D^m &= \{D_i^m\}_{i=1}^{N_c}, \\ D_i^m &= D_i \alpha + \hat{D}_i(1 - \alpha). \end{aligned} \quad (4)$$

The value of α is determined by a projection function, which is monotonically increasing with respect to the number of training iterations. 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 in this work without careful selection. We then employ a sigmoid function to smooth the training process:

$$\alpha = \frac{1}{1 + e^{rx}} \quad (5)$$

where r is a parameter that controls the steepness of the mixup. As $\alpha \rightarrow 1$ by the end of training, the model gains the ability to refine predicted geometry and no longer requires ground-truth depth in inference.

4. Experiments

4.1. Experimental Setup

We evaluate our model using the Occ3D-nuScenes [32] benchmark, which is based on nuScenes [3] dataset and was 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 in 3D voxel space 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. Following previous works [10, 21, 32, 38], we use the mean intersection over union (mIoU) across all classes to evaluate accuracy.

4.2. Implementation Details

Adhering to common practices [21, 23, 25], we adopt ResNet-50 [9] as the image backbone. We maintain a memory queue of length 15 to store historical features and fuse temporal information with 16 frames. For the large-kernel re-parameterized 3D convolution in the geometric encoder, we set the size of convolution kernel to [11, 11, 1]. The steepness parameter r is set to 5 in geometric-semantic decoupled learning. During training, we use a batch size of 32 on 8 Nvidia A100 GPUs. Unless otherwise specified, all models are trained for 24 epochs using the AdamW optimizer [24] 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 [5].

4.3. 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 real-time inference speed and low memory usage while achieving accuracy comparable to or better than non-real-time methods, such as BEVFormer [20], BEVDet4D [11], SurroundOcc [35], and FlashOCC [38]. When compared with FB-Occ [21], the winner of CVPR 2023 occupancy challenge, GSD-Occ is $\sim 3\times$ faster and shows a 1.9% mIoU improvement. Compared to other real-time occupancy prediction methods, GSD-Occ achieves a notable 5.2% higher mIoU with even faster speed than FastOCC [10]. These results highlight the effectiveness of geometric-semantic disentanglement in our method. When we increase the input image size of GSD-Occ to $2\times$, the mIoU further improved by 2.3% without bells and whistles. The inference speed decreases by $2\times$, which indicates a nearly linear relationship between input size and inference speed. This property enables GSD-Occ to efficiently handle high-resolution images. Compared to more recent methods, GSD-Occ* achieves only 0.4% lower mIoU than PannoOcc [34], but it is $\sim 3\times$ faster and uses only $\sim 50\%$ of the memory. Although COTR [25] achieves 2.8% higher mIoU than GSD-Occ*, it is significantly slower ($>10\times$). Additionally, we also report the RayIoU metric proposed by [23] in Tab. 2. GSD-Occ achieves 4.9 % higher mIoU with faster speed and lower memory usage when compared with the recent SOTA method, SparseOcc [23].

We further provide qualitative results in Fig. 5. Despite significantly reducing computation, our method can also effectively perceive geometric details (even with few clues in Row 2) and accurate semantics (Row 3). Additionally, our method also performs well under night conditions (Row 4).

4.4. Ablations

In this section, we conduct ablation experiments on validation split of Occ3D-nuScenes to delve into the effect of each module.

4.4.1 Ablations on GSDBN

The results are shown in Tab. 3, we can observe that each component of geometric-semantic dual-branch network (GSDBN) contributes to the overall performance. The baseline model, which lacks temporal fusion and both 2D and 3D encoders, achieves fast speed (27.0 FPS) but falls short in accuracy (35.11% mIoU). For temporal fusion, although applying voxel features leads to 0.69 % mIoU improvement when compared with using BEV features, it also introduces a significant inference delay (0.029s), which is costly relative to the accuracy gain. Integrating the GSDBN module into the baseline model results in a 3.79%

Table 1. **3D Occupancy prediction performance on the Occ3D-nuScenes dataset.** The FPS of all methods are evaluated on an Nvidia A100 GPU, except for FastOCC, which is tested on an Nvidia V100 GPU as reported. “-” indicates that the metrics are not reported in the paper, and the code is also not open-source. Visible Mask refers to whether models are trained with visible masks.

Method	Venue	Backbone	Image Size	Visible Mask	mIoU (%)	FPS	Memory (MB)
MonoScene [4]	CVPR’22	ResNet-101	928 × 600	✗	6.1	-	-
OccFormer [39]	ICCV’23	ResNet-50	256 × 704	✗	20.4	4.8	7617
CTF-Occ [32]	arXiv’23	ResNet-101	928 × 600	✗	28.5	-	-
SparseOcc [23]	CVPR’24	ResNet-50	256 × 704	✗	30.6	17.7	6883
GSD-Occ (Ours)	-	ResNet-50	256 × 704	✗	31.8	20.0	4759
BEVFormer [20]	ECCV’22	ResNet-101	900 × 1600	✓	39.3	4.7	6651
BEVDet4D [11]	arXiv’22	ResNet-50	256 × 704	✓	39.2	0.8	6053
SurroundOcc [35]	ICCV’23	ResNet-101	900 × 1600	✓	37.1	3.2	5491
FB-OCC [21]	ICCV’23	ResNet-50	256 × 704	✓	37.5	7.0	5467
FlashOCC [38]	arXiv’24	ResNet-50	256 × 704	✓	37.8	4.8	3143
FastOCC [10]	ICRA’24	ResNet-50	320 × 800	✓	34.2	15.9	-
FastOCC* [10]	ICRA’24	ResNet-101	320 × 800	✓	37.2	10.7	-
PanoOcc [34]	CVPR’24	ResNet-101	900 × 1600	✓	42.1	3.0	11991
COTR [25]	CVPR’24	ResNet-50	256 × 704	✓	44.5	0.9	10453
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 2. **3D Occupancy prediction performance on the Occ3D-nuScenes dataset using the RayIoU metric proposed by [23].** The FPS of all methods are evaluated on an Nvidia A100 GPU.

Method	Venue	Backbone	Image Size	Epoch	RayIoU (%)	FPS	Memory (MB)
BEVFormer [20]	ECCV’22	ResNet-101	900 × 1600	24	32.4	4.7	6651
BEVDet4D [11]	arXiv’22	ResNet-50	256 × 704	90	29.6	0.8	6053
FB-OCC [21]	ICCV’23	ResNet-50	256 × 704	90	33.5	7.0	5467
SparseOcc [23]	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 3. **Ablation study on each component of GSDBN.** “3D” and “2D” denote conducting temporal fusion with voxel or BEV features. “BVL” refers to the BEV-Voxel Lifting module.

GSDBN				mIoU	FPS
2D Encoder	Temporal Fusion	3D Encoder	BVL		
✓				35.11	27.0
✓				36.38	23.1
✓	3D			39.09	13.9
✓	2D			38.40	21.4
✓	2D	✓	✓	38.90	20.0

mIoU improvement, with only a modest increase in computational cost (speed decreases from 27.0 FPS to 20.0 FPS). It demonstrates that GSDBN efficiently and effectively decouples the learning of geometry and semantic by a hybrid BEV-Voxel representation.

4.4.2 Ablations on GSDL

To prove the effectiveness of geometric-semantic decoupled learning (GSDL), we apply it to different pre-training models and methods, as shown in Tab. 4. Without incurring additional computation costs, GSDL achieves consistent accuracy improvement across different pre-training mod-

Table 4. **Effectiveness of GSDL with different pretrained models and methods.** BEVDepth means the model weight in [18].

Method	Pretrained model	GSDL	mIoU
FB-OCC [21]	BEVDepth		37.5
FB-OCC [21]	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)

els (BEVDepth [18] and ImageNet [6]) and methods (FB-OCC [21] and our GSD-Occ). It highlights the generalizability of GSDL, which further decouples the geometry and semantic by a simple yet effective learning strategy.

4.4.3 Additional Ablations

The Effectiveness of BVL. We compare BEV-Voxel lifting (BVL) module with the other existing methods as shown in Tab. 5, it shows that BVL module achieves the best accuracy with the fastest speed, proving its effectiveness.

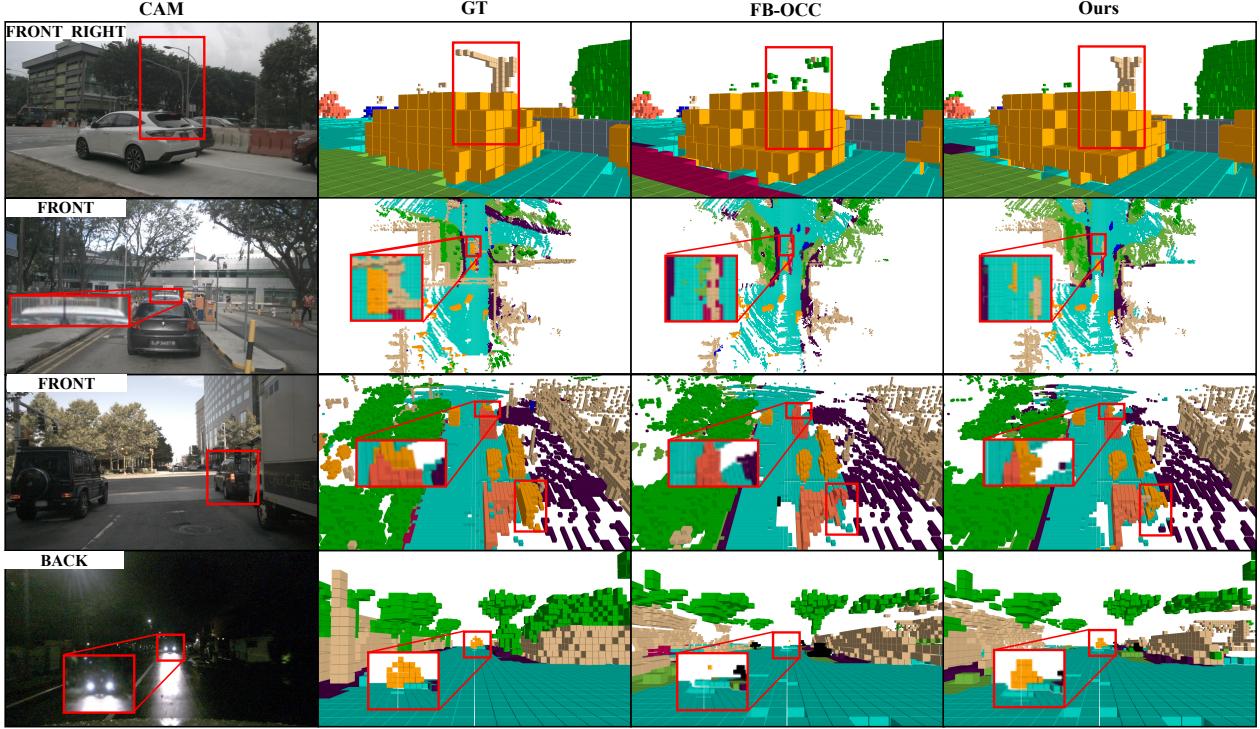


Figure 5. **Qualitative results comparison between FB-OCC and our method.** The results demonstrate that our method is able to construct more detailed geometry (Row 1 and Row 2), more accurate semantics (Row 3), and stronger adaptability in night (Row 4).

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

Lifting Method	mIoU	FPS
Channel-to-Height [38]	38.62	18.7
Repeat + 3D Conv [10]	38.42	18.3
BVL	39.45	20.0

Table 6. **The impact of different number of history frames in temporal fusion.**

Number of history frames	mIoU	FPS
1 (short)	37.28	21.0
7 (moderate)	38.49	20.6
15 (long)	39.45	20.0
31 (very long)	39.32	19.3

Table 7. **The impact of different kernel sizes in 3D encoder.** “-/-” denotes the FPS of before and after re-parameterization.

Kernel size of 3D convolution	mIoU	FPS
7 × 7 × 1	38.67	19.4/20.4
9 × 9 × 1	38.70	19.0/20.2
11 × 11 × 1	38.90	18.6/20.0
13 × 13 × 1	38.65	18.2/19.6
15 × 15 × 1	38.74	17.9/19.4

Are More History Frames Better? As illustrated in Tab. 6, we delve into the impact of various time-series lengths: short (1), moderate (7), long (15), and very long (31). The results indicate that the long temporal fusion achieves the highest accuracy. Since we employ 2D temporal fusion with BEV features, the computational cost remains affordable even as the time-series length increases.

Is a Larger 3D Convolutional Kernel Better? In Table 7, we present the results of different kernel sizes in 3D re-parameterized convolution. Adopting a kernel size of $11 \times 11 \times 1$ achieves the highest accuracy. It indicates that correcting geometric errors requires a relatively large receptive field, but excessively large kernels can be counterproductive. Additionally, thanks to the re-parameterized technique we employed, the inference speed has significantly

improved from 18.6 FPS to 20.0 FPS.

Smooth or steep mixup of predicted and ground-truth depth? As shown in Fig. 6, we plot the curve of Eq. 5 and conduct experiments to explore the impact of various steepness levels in GSDL. When the steepness parameter r is set to 5, we achieve the best accuracy. This suggests that overly smooth mixup curves may hinder the model’s ability to adapt to the predicted depth, while excessively steep curves can complicate the training process.

5. Conclusion

In this paper, we propose GSD-Occ, a employ-friendly real-time 3D occupancy prediction method that achieves accuracy comparable to many non-real-time methods. To achieve this, we identify and address a core challenge: the

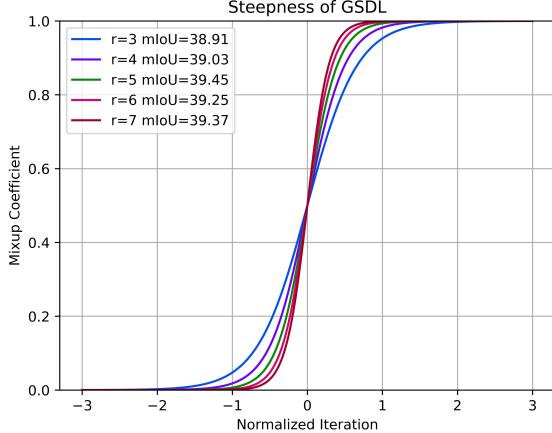


Figure 6. **Ablation study on the steepness (i.e., r in Eq. 5) of GSDL.** “Normalized iteration” refers to the adjusted range of iterations as stated in 3.4. We show the steepness on different r along with their corresponding performance.

strong coupling between geometry and semantic. Specifically, we propose a geometric-semantic dual-branch network with a hybrid BEV-Voxel representation, which maintains both computational efficiency and geometric integrity. Additionally, we propose a geometric-semantic decoupled learning strategy, which separates the learning of geometric correction and semantic knowledge, resulting in consistent accuracy improvements across various pre-training models and methods. To validate the effectiveness of our method, we compare GSD-Occ with recent state-of-the-art (SOTA) methods on the Occ3D-nuScenes benchmark. The results demonstrate that GSD-Occ achieves new SOTA performance in real-time occupancy prediction.

References

- [1] Tesla ai day. <https://www.youtube.com/watch?v=j0z4fwecy4m> (2021). 2021. 1, 3
- [2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2, 5
- [3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liang, Qiang Xu, Anush Krishnan, Yu Pan, Giacarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 6
- [4] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3991–4001, 2022. 1, 7
- [5] MMDetection3D Contributors. MMDetection3D: Open-MMLab next-generation platform for general 3D object detection. <https://github.com/open-mmlab/mmdetection3d>, 2020. 6
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7
- [7] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021. 5
- [8] Xiaohan Ding, Yiyuan Zhang, Yixiao Ge, Sijie Zhao, Lin Song, Xiangyu Yue, and Ying Shan. Unireplnet: 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)*, 2024. 5
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 6
- [10] Jiawei Hou, Xiaoyan Li, Wenhao Guan, Gang Zhang, Di Feng, Yuheng Du, Xiangyang Xue, and Jian Pu. Fastocc: Accelerating 3d occupancy prediction by fusing the 2d bird’s-eye view and perspective view. *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 2, 3, 6, 7, 8
- [11] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 1, 2, 4, 6, 7
- [12] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2, 3
- [13] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9223–9232, 2023. 1, 3
- [14] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1042–1050, 2023. 2
- [15] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 1
- [16] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhui Wang, Lewei Lu, Huijie Wang, Enze Xie, Zhiqi Li, Hanming Deng, Hao Tian, et al. Delving into the devils of bird’s-eye-view perception: A review, evaluation and recipe. *arXiv preprint arXiv:2209.05324*, 2022. 2
- [17] Yiniao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *Pro-*

- ceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1486–1494, 2023. 1, 2
- [18] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1477–1485, 2023. 2, 3, 4, 7
- [19] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9087–9098, 2023. 1
- [20] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision (ECCV)*, pages 1–18. Springer, 2022. 1, 2, 3, 6, 7
- [21] Zhiqi Li, Zhiding Yu, Wenhai Wang, Anima Anandkumar, Tong Lu, and Jose M Alvarez. Fb-bev: Bev representation from forward-backward view transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6919–6928, 2023. 1, 3, 4, 6, 7
- [22] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems (NIPS)*, 35:10421–10434, 2022. 1
- [23] Haisong Liu, Haiguang Wang, Yang Chen, Zetong Yang, Jia Zeng, Li Chen, and Limin Wang. Fully sparse 3d panoptic occupancy prediction. *European conference on computer vision (ECCV)*, 2024. 1, 3, 6, 7
- [24] Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5, 2017. 6
- [25] Qihang Ma, Xin Tan, Yanyun Qu, Lizhuang Ma, Zhizhong Zhang, and Yuan Xie. Cotr: Compact occupancy transformer for vision-based 3d occupancy prediction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 3, 4, 6, 7
- [26] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris M Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. In *The Eleventh International Conference on Learning Representations*, 2022. 4
- [27] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European conference on computer vision (ECCV)*, pages 194–210. Springer, 2020. 2, 3, 4, 5
- [28] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 2, 5
- [29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2, 5
- [30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015. 2
- [31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 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*, pages 234–241. Springer, 2015. 5
- [32] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *arXiv preprint arXiv:2304.14365*, 2023. 1, 3, 6, 7
- [33] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022. 2
- [34] Yuqi Wang, Yuntao Chen, Xingyu Liao, Lue Fan, and Zhaoxiang Zhang. Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 3, 6, 7
- [35] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21729–21740, 2023. 1, 3, 6, 7
- [36] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. 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)*, pages 17830–17839, 2023. 2
- [37] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 1
- [38] Zichen Yu, Changyong Shu, Jiajun Deng, Kangjie Lu, Zongdai Liu, Jiangyong Yu, Dawei Yang, Hui Li, and Yan Chen. Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv preprint arXiv:2311.12058*, 2023. 2, 3, 4, 6, 7, 8
- [39] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9433–9443, 2023. 1, 7
- [40] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2