OpenOccupancy: A Large Scale Benchmark for Surrounding Semantic Occupancy Perception

Xiaofeng Wang[1,3*] Zheng Zhu[2*†] Wenbo Xu[2*] Yunpeng Zhang[2]
Yi Wei[4] Xu Chi[2] Yun Ye[2] Dalong Du[2] Jiwen Lu[4] Xingang Wang[1†]
[1]Institute of Automation, Chinese Academy of Sciences [2]PhiGent Robotics
[3]University of Chinese Academy of Sciences [4]Tsinghua University

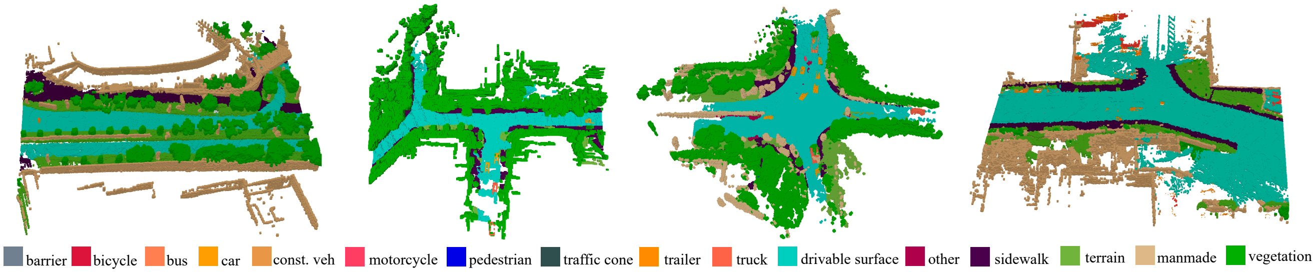

Figure 1: The nuScenes-Occupancy provides dense semantic occupancy labels for all key frames in the nuScenes [3] dataset. Here we showcase the annotated ground truth with the volumetric size of $(40 \times 512 \times 512)$ and grid size of 0.2 m.

Abstract

*Semantic occupancy perception is essential for autonomous driving, as automated vehicles require a fine-grained perception of the 3D urban structures. However, existing relevant benchmarks lack diversity in urban scenes, and they only evaluate front-view predictions. Towards a comprehensive benchmarking of surrounding perception algorithms, we propose OpenOccupancy, which is the first surrounding semantic occupancy perception benchmark. In the OpenOccupancy benchmark, we extend the large-scale nuScenes dataset with dense semantic occupancy annotations. Previous annotations rely on LiDAR points superimposition, where some occupancy labels are missed due to sparse LiDAR channels. To mitigate the problem, we introduce the **A**ugmenting **A**nd **P**urifying (AAP) pipeline to ∼2× densify the annotations, where ∼4000 human hours are involved in the labeling process. Besides, camera-based, LiDAR-based and multi-modal baselines are established for the OpenOccupancy benchmark. Furthermore, considering the complexity of surrounding occupancy perception lies in the computational burden of high-resolution 3D predictions, we propose the Cascade Occupancy Network (CONet) to refine the coarse prediction, which relatively enhances the performance by ∼30% than the baseline. We hope the OpenOccupancy benchmark [‡] will boost the development of surrounding occupancy perception algorithms.*

1. Introduction

Accurately perceiving 3D structures of different objects and regions in urban scenes is a fundamental requirement for safe driving, thus there are growing interests in semantic occupancy perception [1, 35, 10, 36, 41, 16, 8]. Unlike 3D detection [13, 6, 34, 3, 38] and LiDAR segmentation [1, 38, 11] that are designed for foreground objects or sparse scanned points, the occupancy task targets at assigning semantic labels to every spatially-occupied region within the perceptive range. Therefore, semantic occupancy perception is a promising and challenging research direction in autonomous-driving perception.

Despite growing interests in semantic occupancy perception, most of the relevant benchmarks [35, 10, 36, 41, 16, 8] are devised for indoor scenes. SemanticKITTI [1] extends the occupancy perception to driving scenarios, but its dataset is relatively small in scale and limited in diversity, which hinders the generalization and evaluation of the developed occupancy perception algorithms. Besides, SemanticKITTI only evaluates the front-view occupancy

[*] These authors contributed equally to this work.
[†] Corresponding authors. zhengzhu@ieee.org, xingang.wang@ia.ac.cn
[‡] https://github.com/JeffWang987/OpenOccupancy

	Type	Surround	Modality	Vol. Size	#Scenes	#Frames	Annotation
NYUv2 [35]	Indoor	✗	C&D	$(144 \times 240 \times 240)$	1.4K	1.4K	Human
ScanNet [8]	Indoor	✗	C&D	$(31 \times 62 \times 62)$	1.5K	1.5K	Human
SceneNN [16]	Indoor	✗	C&D	-	100	-	Human
SUNCG [36]	Synthtic	✗	C&D	$(144 \times 240 \times 240)$	46K	140K	Synthtic
SynthCity [14]	Synthtic	✗	L	-	9	-	Synthtic
SemanticPOSS [30]	Outdoor	✓	L	-	-	3K	Human
SemanticKITTI [1]	Outdoor	✗	C&L	$(32 \times 256 \times 256)$	22	44K	Auto&Human
nuScenes-Occupancy	Outdoor	✓	C&L	$(40 \times 512 \times 512)$	850	200K[1]	Auto&Human

Table 1: Comparison between nuScenes-Occupancy and other dense LiDAR/occupancy perception datasets. *Surround*=✓ represents datasets that use surround-view inputs. *C, D, L* denote camera, depth and LiDAR. *Vol. Size* is the volumetric size. [1] Note that nuScenes-Occupancy has 34K key frames, where 6 images are in each frame (*i.e.*, 200K image frames).

predictions, while the surrounding perception is more critical for safe driving. To address these problems, we propose OpenOccupancy, which is the first surrounding semantic occupancy perception benchmark. In the OpenOccupancy benchmark, we introduce nuScenes-Occupancy, which extends the large-scale nuScenes [3] dataset with dense semantic occupancy annotation. As shown in Tab. 1, the number of annotated scenes and frames (of nuScenes-Occupancy) are ∼40× and ∼5× more than that of [1]. Notably, it is almost impractical to directly annotate large-scale occupancy labels by human labor. Therefore, the **A**ugmenting **A**nd **P**urifying (AAP) pipeline is introduced to efficiently annotate and densify the occupancy labels. Specifically, we initialize annotation by multi-frame LiDAR points superimposition, where the per-point semantic labels are from [11]. Considering the sparsity of the initial annotation (*i.e.*, some occupancy labels are missed due to occlusion or limited LiDAR channels), we augment it with pseudo occupancy labels, which are constructed by the pretrained baseline (see Sec. 3.4). To further reduce noise and artifacts, human endeavors are leveraged to purify the augmented annotation. Based on the AAP pipeline, we generate ∼2× dense occupancy labels than the initial annotation. Visualizations of the dense annotation are shown in Fig. 1.

To facilitate future research, we establish camera-based, LiDAR-based and multi-modal baselines for the OpenOccupancy benchmark. Experiment results show that the camera-based method achieves better performance on small objects (*e.g.*, *bicycle, pedestrian, motorcycle*), while the LiDAR-based approach shows superior performance on large structured regions (*e.g.*, *drivable surface, sidewalk*). Notably, the multi-modal baseline adaptively fuses intermediate features from both modalities, relatively improving the overall performance (of camera-based and LiDAR-based methods) by 46% and 34%. Considering the computational burden of the surrounding occupancy perception, the proposed baselines can only generate low-resolution predictions. Towards an efficient occupancy perception, we propose the Cascade Occupancy Network (CONet) that builds a coarse-to-fine pipeline upon the proposed baseline, relatively improving the performance by ∼30%.

The main contributions are summarized as follows: (1) We propose OpenOccupancy, which is the first benchmark designed for surrounding occupancy perception in driving scenarios. (2) The AAP pipeline is proposed to efficiently annotate and densify semantic occupancy labels of the nuScenes dataset, and the resulted nuScenes-Occupancy is the first dataset for surrounding semantic occupancy segmentation. (3) We establish camera-based, LiDAR-based and multi-modal baselines in the OpenOccupancy benchmark. Besides, the CONet is introduced to alleviate the computational burden of high-resolution occupancy predictions, which relatively improves the baseline by ∼30%. (4) Based on the OpenOccupancy benchmark, we conduct comprehensive experiments on the proposed baselines, CONet, and modern occupancy perception approaches.

2. Related Work

Semantic occupancy perception benchmarks. Semantic occupancy perception originates from SUNCG [36], where the algorithms are required to output occupancy and semantic labels for all voxels in the camera-view frustum. In recent years, semantic occupancy perception draws growing attention and is thoroughly reviewed in [33]. To facilitate the development of occupancy perception, various relevant benchmarks have been released [1, 35, 10, 36, 41, 16, 8, 30, 14]. Among these benchmarks, SUNCG [36], NYUv2 [35], NYUCAD [10], SUN3D [41], SceneNN [16], ScanNet [8] focus on the indoor stationary scenarios. Unlike the prevalence of indoor datasets, few benchmarks [14, 1, 30, 11] are devised for outdoor scenes. SynthCity [14], SemanticPOSS [30], Panoptic nuScenes [3] only provide semantic labels for sparse/synthetic point clouds. SemanticKITTI [1] is most relevant to the proposed OpenOccupancy benchmark, as it annotates real-world occupancy in driving scenarios. However, SemanticKITTI lacks diversity in urban scenes, which hinders the gener-

Algorithm 1 Augmenting And Purifying (AAP)

```
Input:
    $P = \{P_i\}_{i=1}^{N} \in \mathbb{R}^{M\times 3}$ are multi-frame LiDAR points.
    $T = \{T_i\}_{i=1}^{N} \in \mathbb{R}^{N\times 4\times 4}$ are extrinsic parameters.
    $B = \{B_i\}_{i=1}^{N}$ are bounding boxes in each frame.
    $S = \{S_i\}_{i=1}^{N} \in \mathbb{R}^{M}$ are semantic labels of $P$.
    $I = \{I_i\}_{i=1}^{N} \in \mathbb{R}^{N\times 6\times H_i\times W_i\times 3}$ are multi-frame images.
Output:
    Multi-frame occupancy ground truth $V_{\text{final}} = \{V_i\}_{i=1}^{N}$.
$V_{\text{init}} = \mathcal{F}_{\text{vox}}(\mathcal{F}_{\text{sup}}(P, L, T, B))$   $V_{\text{init}} \in \mathbb{R}^{N\times D\times H\times W}$
$\mathcal{F}_{\text{m}} = \text{TRAIN}(\mathcal{F}_{\text{m}}(P, I), V_{\text{init}})$
$V_{\text{pseudo}} = \mathcal{F}_{\text{m}}(P, I)$   $V_{\text{pseudo}} \in \mathbb{R}^{N\times D\times H\times W}$
$V_{\text{aug}} = \mathcal{F}_{\text{aug}}(V_{\text{pseudo}}, V_{\text{init}})$   $V_{\text{aug}} \in \mathbb{R}^{N\times D\times H\times W}$
$V_{\text{final}} = \mathcal{F}_{\text{purify}}(V_{\text{aug}})$   $V_{\text{final}} \in \mathbb{R}^{N\times D\times H\times W}$
```

alization of occupancy perception algorithms. Besides, it only evaluates front-view occupancy predictions.

Semantic occupancy perception approaches. Most existing occupancy perception methods rely on geometric inputs, including occupancy grids [43, 32, 12, 40], LiDAR points [31, 48], RGBD images [21, 22, 23, 24, 27], and Truncated Signed Distance Function (TSDF) [4, 7, 37, 9, 39, 46, 47]. MonoScene [5] is the first camera-based occupancy perception method in the literature, which can deduce occupancy semantics from a single image. Despite the significant development of occupancy perception approaches, most of them focus on front-view indoor scenarios. Recently, TPVFormer [18] proposes a *tri-perspective view* representation to generate surrounding occupancy prediction, yet its occupancy output is relatively sparse, as TPVFormer is designed for LiDAR segmentation.

3. The OpenOccupancy Benchmark

In this section, the concept of surrounding semantic occupancy perception is first introduced. Then we introduce nuScenes-Occupancy, which extends the nuScenes dataset [3] with dense semantic occupancy annotations based on the AAP pipeline. Subsequently, the evaluation protocol is presented to comprehensively assess the surrounding occupancy perception algorithms. Finally, we propose camera-based, LiDAR-based and multi-modal baselines for the OpenOccupancy Benchmark.

3.1. Surrounding Semantic Occupancy Perception

Referring to [36], surrounding semantic occupancy perception is a task for generating a complete 3D representation of volumetric occupancy and semantic labels for a scene. Different from the monocular paradigm [36] that focuses on the front-view perception, the surrounding occupancy perception algorithms target at producing semantic occupancy in the surround-view driving scenarios. Specifically, given 360-degree inputs X_i (*e.g.*, LiDAR sweeps or surround-view images), the perception algorithms are required to predict the surrounding occupancy labels $\mathcal{F}(X_i) \in \mathbb{R}^{D\times H\times W}$, where D, H, W is the volumetric size of the entire scene. It is noted that the surround-view inputs cover ∼5× perceptive range more than that of front-view sensors. Therefore, the core challenge of the surrounding occupancy perception lies in efficiently constructing high-resolution occupancy.

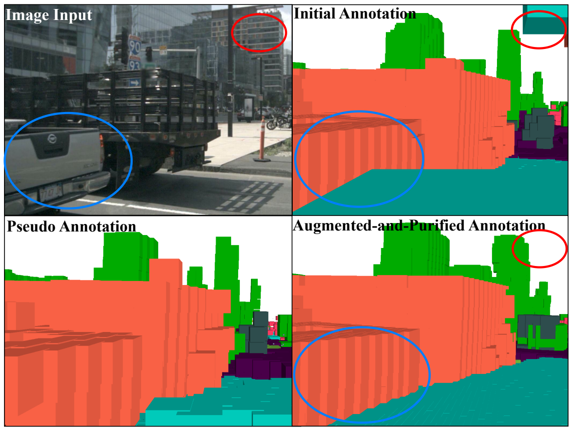

Figure 2: Comparison between the initial, pseudo and the augmented-and-purified annotation, where regions highlighted by red and blue circle indicate that the augmented annotation is more dense and accurate.

3.2. nuScenes-Occupancy

SemanticKITTI [1] is the first dataset for outdoor occupancy perception, but it lacks diversity in driving scenes and only evaluates front-view predictions. Towards a large-scale surrounding occupancy perception dataset, we introduce the nuScenes-Occupancy that extends the nuScenes [3] dataset with dense semantic occupancy annotation. Although sparse LiDAR semantic labels are provided in [11], it is almost unfeasible to directly annotate dense occupancy labels through human effort. Therefore, the AAP pipeline is introduced to efficiently annotate and densify the occupancy labels.

The overall AAP pipeline is shown in Alg. 1. We first initialize annotation by LiDAR points superimposition $V_{\text{init}} = \mathcal{F}_{\text{vox}}(\mathcal{F}_{\text{sup}}(P, L, T, B))$ [1], where static points (*e.g.*, *sidewalk*) are transformed to the unified world coordinate using extrinsics T. For movable objects (*e.g.*, the moving *car*), we transform point clouds to coordinates of their bounding boxes B (each object in different frames can be associated via the instance token [29]). Subsequently, the static and dynamic points are concatenated and voxelized ($\mathcal{F}_{\text{vox}}$) to produce the initial occupancy annotation V_{init}, where the semantic labels S are form [11]. Note that some occupancy labels are missed due to occlusion or sparse LiDAR channels. Inspired by self-training [42], we complement the initial annotation with pseudo occupancy labels. Specifically, the initial annotation is utilized to train the proposed multi-modal baseline $\mathcal{F}_{\text{m}}$ (see Sec. 3.4), and pseudo occupancy labels V_{pseudo} are produced by the pretrained model. Then we augment initial labels with pseudo labels to construct dense annotations $V_{\text{aug}} = \mathcal{F}_{\text{aug}}(V_{\text{pseudo}}, V_{\text{init}})$. To resolve conflicts in the two annotations, we only augment empty voxels in V_{init}:

$$V_{\text{aug}}(x,y,z) = \begin{cases} V_{\text{init}}(x,y,z) & V_{\text{init}}(x,y,z) \text{ is occupied} \\ V_{\text{pseudo}}(x,y,z) & \text{else.} \end{cases} \tag{1}$$

Regarding artifacts caused by pseudo labels, human endeavors are further leveraged to purify the augmented labels and establish final annotation $V_{\text{final}} = \mathcal{F}_{\text{purify}}(V_{\text{aug}})$. For efficiency, labeling software is devised for human annotators, where the 3D semantic occupancy is projected to multi-view images, and annotators can efficiently determine the occupancy boundary through both 3D global view and 2D camera views (the purifying process involves ∼4000 human hours of labeling effort).

As shown in Fig. 2, the pseudo labels are complementary to the initial annotation, and the augmented-and-purified labels are more dense and precise. Notably, ∼400K occupied voxels are in each frame of the augmented-and-purified annotation, which is ∼2× dense than the initial annotation. In summary, nuScenes-Occupancy has 28130 training frames and 6019 validation frames, where 17 semantic labels (same as [11]) are assigned to occupied voxels in each frame.

3.3. Evaluation Protocol

The evaluation range is set as $[-51.2\text{m}, 51.2\text{m}]$ for X, Y axis, and $[-5\text{m}, 3\text{m}]$ for Z axis. Following [1], the voxel resolution is 0.2m, which results in a volume of $40 \times 512 \times 512$ voxels for occupancy prediction. For evaluation metrics, we utilize Intersection of Union (IoU) [1] as the *geometric metric*, which identifies a voxel as being occupied or empty (*i.e.*, deem all occupied voxels as one category):

$$\text{IoU} = \frac{\text{TP}_{\text{o}}}{\text{TP}_{\text{o}} + \text{FP}_{\text{o}} + \text{FN}_{\text{o}}}, \tag{2}$$

where $\text{TP}_o, \text{FP}_o, \text{FN}_o$ are the number of true positive, false positive and false negative predictions for occupied voxels. Besides, we calculate the mean IoU (mIoU) of each class as the *semantic metric*:

$$\text{mIoU} = \frac{1}{\text{C}_{\text{sem}}} \sum_{\text{c}=1}^{\text{C}_{\text{sem}}} \frac{\text{TP}_{\text{c}}}{\text{TP}_{\text{c}} + \text{FP}_{\text{c}} + \text{FN}_{\text{c}}}, \tag{3}$$

where $\text{TP}_c, \text{FP}_c, \text{FN}_c$ denote the number of true positive, false positive and false negative predictions for class c, and C_{sem} is the total number of classes. Following [11], the *noise* class [11] is ignored in the evaluation.

3.4. OpenOccupancy Baselines

The majority of existing occupancy perception methods [21, 22, 23, 24, 27, 36, 4, 7, 37, 9, 39, 5] are designed for front-view perception. To extend these approaches to surrounding occupancy perception, each camera-view input is processed individually, which is inefficient. Besides, inconsistency may exist in the overlap region of two adjacent outputs. To mitigate these problems, we establish baselines that coherently learn surrounding semantic occupancy from 360-degree inputs (*e.g.*, LiDAR sweeps or surround-view images). Specifically, camera-based, LiDAR-based and multi-modal baselines are proposed for the OpenOccupancy benchmark.

LiDAR-based baseline. As shown in the top-left diagram of Fig. 3, parameterized voxelization [49] is first utilized to embed raw LiDAR points to voxelized features. For computational efficiency, 3D sparse convolutions [44] are leveraged to encode features in the voxel space, producing LiDAR voxel features $F^{\mathcal{L}}$ with reduced spatial dimension ($\frac{D}{S} \times \frac{H}{S} \times \frac{W}{S}$, S is the stride). The voxel features are further decoded by 3D convolutions, generating multi-scale voxel features $F_i^{\mathcal{L}} \in \mathbb{R}^{\frac{D}{2^i S} \times \frac{H}{2^i S} \times \frac{W}{2^i S} \times C_i} (i = 0, 1, 2)$. These features are upsampled and concatenated along the channel dimension, resulting in $F_{\text{ms}}^{\mathcal{L}} \in \mathbb{R}^{\frac{D}{S} \times \frac{H}{S} \times \frac{W}{S} \times \sum_{i=0}^{2} C_i}$. Finally, the occupancy head is utilized to reduce feature channels, and a *softmax* function is leveraged to produce semantic probabilities. The output $O^{\mathcal{L}} \in \mathbb{R}^{\frac{D}{S} \times \frac{H}{S} \times \frac{W}{S} \times 18}$ (18: 1 empty label with 17 semantic labels in nuScenes-Occupancy) can be scaled to arbitrary sizes using the *trilinear interpolation*, and class labels can be determined by the *argmax* function along the channel dimension.

Camera-based baseline. As illustrated in the bottom of Fig. 3, the 2D encoder (*e.g.*, ResNet [15] and FPN [26]) is first utilized to extract multi-view features F^{mv}. Subsequently, we apply the *2D to 3D view transform* [28] to project 2D features into 3D ego-car coordinates. Different from [28] that collapses 3D features onto the Bird's Eye View (BEV) plane, the height information is reserved for a fine-grained 3D occupancy prediction. The resulted camera voxel features $F^{\mathcal{C}}$ have the same volumetric size as that of $F^{\mathcal{L}}$. Following the LiDAR-based baseline, we further employ the 3D decoder and occupancy head to output the semantic occupancy $O^{\mathcal{C}} \in \mathbb{R}^{\frac{D}{S} \times \frac{H}{S} \times \frac{W}{S} \times 18}$.

Multi-modal baseline. The LiDAR voxel features $F^{\mathcal{L}}$ and camera voxel features $F^{\mathcal{C}}$ are natural representations for occupancy prediction. In the multi-modal baseline, we propose the adaptive fusion module to dynamically integrate features from $F^{\mathcal{L}}$ and $F^{\mathcal{C}}$:

$$W = \mathcal{G}_{\text{C}}([\mathcal{G}_{\text{C}}(F^{\mathcal{L}}), \mathcal{G}_{\text{C}}(F^{\mathcal{C}})]), \tag{4}$$

$$F^{\mathcal{F}} = \sigma(W) \odot F^{\mathcal{L}} + (1 - \sigma(W)) \odot F^{\mathcal{C}}, \tag{5}$$

where $\mathcal{G}_{\text{C}}$ is the 3D convolution, $[\cdot, \cdot]$ is the concatenation

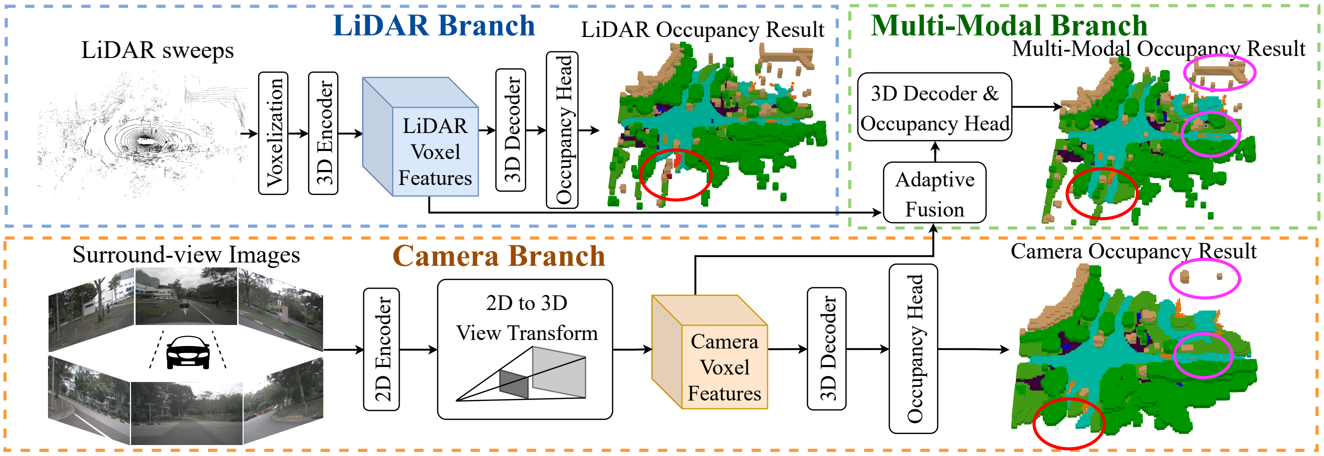

Figure 3: Overall architecture of three proposed baselines. The LiDAR branch utilizes 3D encoder to extract voxelized LiDAR features, and the camera branch uses 2D encoder to learn surround-view features, which are then transformed to generate 3D camera voxel features. In the multi-modal branch, the adaptive fusion module dynamically integrates features from two modalities. All three branches leverage 3D decoder and occupancy head to produce semantic occupancy. In the occupancy results figures, regions highlighted by red and purple circles indicate that the multi-modal branch can generate more complete and accurate predictions (better viewed when zoomed in).

along feature channel, σ denotes *Sigmoid* function and $\odot$ represents element-wise product. Based on the fused voxel features $F^{\mathcal{F}}$, the final occupancy can be predicted by the aforementioned 3D decoder and occupancy head.

To train the proposed baselines, cross-entropy loss $\mathcal{L}_{\text{ce}}$ and lovasz-softmax loss $\mathcal{L}_{\text{ls}}$ [2] are leveraged to optimize the network. Following [5], we also utilize affinity loss $\mathcal{L}_{\text{scal}}^{\text{geo}}$ and $\mathcal{L}_{\text{scal}}^{\text{sem}}$ to optimize the scene-wise and class-wise metrics (*i.e.*, geometric IoU and semantic mIoU). Besides, the explicit depth supervision $\mathcal{L}_{\text{d}}$ [25] is used to train a depth-aware *view transform* module. Therefore, the overall loss function can be derived as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{ls}} + \mathcal{L}_{\text{scal}}^{\text{geo}} + \mathcal{L}_{\text{scal}}^{\text{sem}} + \mathcal{L}_{\text{d}}, \tag{6}$$

where $\mathcal{L}_{\text{d}}$ is only calculated in the camera-based and multi-modal baseline.

4. Cascade Occupancy Network

Compared with front-view occupancy perception [1], the input of the surrounding occupancy perception covers ∼5× perceptive range. Therefore, the complexity lies in the computational burden of high-resolution 3D prediction. For efficiency, the stride parameter S is set as 4 in the proposed baselines (*i.e.*, the volumetric size of the output is $(10 \times 128 \times 128)$). Notably, we empirically find that using a smaller stride parameter (*e.g.*, S=2) enhances the performance. However, the GPU memory is approximately 2× upscaled (∼40 GB in the training phase). Therefore, we propose the Cascade Occupancy Network for an efficient yet accurate surrounding occupancy perception.

Specifically, CONet introduces a coarse-to-fine pipeline, which can be efficiently built upon the proposed baselines. Taking the multi-modal baseline for example (termed as multi-modal CONet), the overall framework is shown in Fig. 4. The coarse occupancy $O^{\mathcal{M}} \in \mathbb{R}^{\frac{D}{S} \times \frac{H}{S} \times \frac{W}{S} \times 18}$ is first generated by the multi-modal baseline, where the occupied voxels $V_{\text{o}} \in \mathbb{R}^{N_{\text{o}} \times 3}$ (N_{o} is the number of occupied voxels, and 3 denotes the (x, y, z) indices in voxel coordinates) are split as high-resolution occupancy queries $Q_{\text{H}} \in \mathbb{R}^{N_{\text{o}} 8^{\eta-1} \times 3}$:

$$Q_{\text{H}} = \mathcal{T}_{\text{v}\to\text{w}}(\mathcal{F}_{\text{s}}(V_{\text{o}}, \eta)), \tag{7}$$

where $\mathcal{F}_{\text{s}}$ is the voxel split function (*i.e.*, for (x_0, y_0, z_0) in V_{o}, the split indices are $\{x_0 + \frac{i}{\eta}, y_0 + \frac{j}{\eta}, z_0 + \frac{k}{\eta}\}(i, j, k \in (0, \eta - 1)))$, η is the split ratio (typically set as 4), and $\mathcal{T}_{\text{v}\to\text{w}}$ transforms the voxel coordinates to the world coordinates. Subsequently, we project Q_{H} on 2D image plane to sample semantic features $F^{\mathcal{S}} = \mathcal{G}_{\text{S}}(F^{mv}, \mathcal{T}_{\text{w}\to\text{c}}(Q_{\text{H}}))$, and transform Q_{H} to voxel space to sample geometric features $F^{\mathcal{G}} = \mathcal{G}_{\text{S}}(F^{\mathcal{F}}, \mathcal{T}_{\text{w}\to\text{v}}(Q_{\text{H}}))$ ($\mathcal{G}_{\text{S}}$ is the *grid sample* function [19], $\mathcal{T}_{\text{w}\to\text{c}}$ and $\mathcal{T}_{\text{w}\to\text{v}}$ are transformations from world coordinates to camera coordinates and voxel coordinates). The sampled features are then fused and regularized by FC layers to produce fine-grained occupancy predictions:

$$O^{\text{fg}} = \mathcal{G}_f(\mathcal{G}_f(F^{\mathcal{S}}) + \mathcal{G}_f(F^{\mathcal{G}})), \tag{8}$$

where $\mathcal{G}_f$ are FC layers. Finally, O^{fg} can be reshaped to the volumetric representation $O^{\text{vol}} \in \mathbb{R}^{\frac{\eta D}{S} \times \frac{\eta H}{S} \times \frac{\eta W}{S} \times 18}$:

$$O^{\text{vol}}(x, y, z) = \begin{cases} O^{\text{fg}}(\mathcal{T}_{\text{v}\to\text{q}}(x, y, z)) & (x, y, z) \in \mathcal{T}_{\text{w}\to\text{v}}(Q_{\text{H}}) \\ \text{Empty Label} & (x, y, z) \notin \mathcal{T}_{\text{w}\to\text{v}}(Q_{\text{H}}), \end{cases} \tag{9}$$

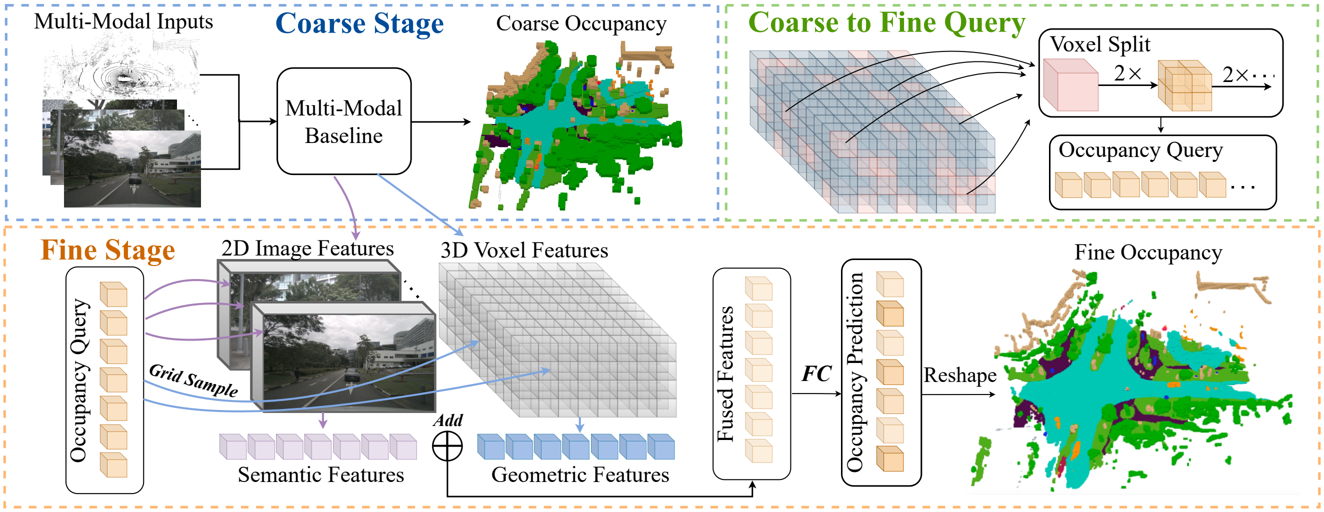

Figure 4: Overall framework of the multi-modal CONet. (1) The coarse occupancy is first generated by the multi-modal baseline. (2) Then the occupied voxels are split to produce high-resolution occupancy queries. (3) Subsequently, we project queries to sample from 2D image features and 3D voxel features. The sampled features are fused and regularized by Fully-Connected (FC) layers to generate fine-grained occupancy predictions.

where $\mathcal{T}_{\mathrm{v}\rightarrow\mathrm{q}}$ transforms the voxel coordinates to indices of the high-resolution query Q_{H}. Notably, the CONet can also be generalized to camera-based and LiDAR-based baselines. For camera-based CONet, we sample Q_{H} from F^{mv} and $F^{\mathcal{C}}$. For LiDAR-based CONet that without multi-view 2D features, we only sample Q_{H} from $F^{\mathcal{L}}$.

For optimization, we use the same pipeline as that of baselines, except that the training losses are calculated on both (coarse and fine) predictions.

5. OpenOccupancy Experiment

In this section, the experiment setup is first given. Then we delve into surrounding occupancy assessment, including camera-based methods, LiDAR-based methods and multi-modal methods. In the next step, we analyze the baseline performance under different experiment settings. Finally, the efficiency and effectiveness of CONet are investigated.

5.1. Experiment Setup

a weight decay of 0.01 and an initial learning rate of 3e-4. We adopt the cosine learning rate scheduler with linear warming up in the first 500 iterations, and a similar augmentation strategy as BEVDet [17]. All models are trained for 15 epochs with a batch size of 8 on 8 A100 GPUs.

5.2. Surrounding Occupancy Assessment

Equipped with the OpenOccupancy benchmark, we analyze the surrounding occupancy perception performance of six modern approaches (MonoScene [5], TPVFormer [18], 3DSketch [7], AICNet [21], LMSCNet [32], JS3C-Net [43] and the proposed baselines and CONet. From the results in Tab. 2, it can be observed that:

(1) Compared with single-view methods, the surrounding occupancy perception paradigm shows superior performance. Specifically, the proposed camera-based baseline and TPVFormer relatively improve MonoScene 51% and 15% on mIoU. Besides, the LiDAR-based baseline and surrounding occupancy perception methods [32, 43] surpass the RGBD paradigms [21, 7] on both IoU and mIoU. Therefore, it is promising to develop surrounding occupancy perception approaches on the OpenOccupancy benchmark.

(2) The proposed baselines show adaptability and scalability for the surrounding occupancy perception. For the camera-based methods, our baseline relatively improves TPVFormer by 19% and 31% on IoU and mIoU. For the LiDAR-based methods, our baseline outperforms LMSCNet and is comparable to JS3C-Net (Note that JS3C-Net is a two-stage method). Additionally, the proposed baselines explicitly optimize the network in a unified voxel representation, which can be naturally extended for multi-modal fusion. Consequently, the proposed multi-modal baseline relatively enhances 3DSketch, AICNet, LMSCNet, and JS3C-Net by 45%, 46%, 35%, and 25% on mIoU.

(3) Information from the camera and LiDAR are complementary to each other, and the multi-modal baseline significantly enhances the performance. Experiment results show that the LiDAR-based approach shows superior performance on large structured regions (*e.g.*, *drivable surface, sidewalk, vegetation*), while the camera-based baseline gains better performance on small objects (*e.g.*, *bicycle, pedestrian, motorcycle, traffic cone*). Notably, the multi-

Method	Input	Surround	IoU	mIoU	barrier ■	bicycle ■	bus ■	car ■	const. veh. ■	motorcycle ■	pedestrian ■	traffic cone ■	trailer ■	truck ■	drive. suf. ■	other flat ■	sidewalk ■	terrain ■	manmade ■	vegetation ■
MonoScene [5]	C	✗	17.1	7.2	7.3	4.3	9.6	7.1	6.2	3.5	5.9	4.7	5.6	4.9	15.6	6.8	7.9	7.6	10.5	7.9
TPVFormer [18]	C	✓	15.1	8.3	9.7	4.5	11.5	10.7	5.5	4.6	6.3	5.4	6.9	6.9	14.1	9.8	8.9	9.0	9.9	8.5
3DSketch [7]	C&D	✗	25.3	11.0	12.3	5.2	10.3	12.1	7.1	4.9	5.5	6.9	8.4	7.4	21.9	15.4	13.6	12.1	12.1	21.2
AICNet [21]	C&D	✗	23.2	10.9	11.8	4.5	12.1	12.7	6.0	3.9	6.4	6.3	8.4	7.8	24.2	13.4	13.0	11.9	11.5	20.5
LMSCNet [32]	L	✓	26.7	11.8	12.9	5.2	12.8	12.6	6.6	4.9	6.3	6.5	8.8	7.7	24.3	12.7	16.5	14.5	14.2	22.1
JS3C-Net [43]	L	✓	29.6	12.7	14.5	4.4	13.5	12.0	7.8	4.4	7.3	6.9	9.2	9.2	27.4	15.8	15.9	16.4	14.0	**24.8**
C-baseline (ours)	C	✓	17.9	10.9	9.3	7.2	11.0	12.5	7.0	9.3	8.9	5.2	4.9	10.2	23.1	17.4	15.4	14.3	8.4	11.0
L-baseline (ours)	L	✓	22.3	11.9	11.1	4.0	11.4	12.9	7.2	6.2	10.1	4.4	8.1	11.0	23.3	15.8	15.5	15.6	15.0	18.7
M-baseline (ours)	C&L	✓	23.5	15.9	14.3	12.7	15.0	16.4	12.6	16.4	15.3	9.5	9.8	15.6	24.8	19.0	17.4	17.9	16.6	20.6
C-CONet (ours)	C	✓	21.6	13.6	13.6	8.4	14.7	18.3	7.1	11.0	11.8	8.8	5.2	13.0	32.7	21.1	20.1	17.6	5.1	8.4
L-CONet (ours)	L	✓	**30.1**	15.9	18.0	3.9	14.2	18.7	8.3	6.3	11	5.8	14.1	14.3	**35.3**	20.2	21.5	**20.9**	**19.2**	23.0
M-CONet (ours)	C&L	✓	26.5	**20.5**	**23.3**	**16.1**	**22.2**	**24.6**	**13.3**	**20.1**	**21.2**	**14.4**	**17.0**	**21.3**	31.8	**22.0**	**21.8**	20.5	17.7	20.4

Table 2: Performance on nuScenes-Occupancy (validation set). We report the geometric metric IoU, semantic metric mIoU, and the IoU for each semantic class. The C, D, L denotes *camera, depth, LiDAR*. For *Surround*=✓, the method directly predicts surrounding semantic occupancy with 360-degree inputs. Otherwise, the method produces the results of each camera view, and then concatenates them as surrounding outputs.

modal baseline adaptively fuses intermediate features from both modalities, relatively enhancing the LiDAR-based and camera-based baseline by 46% and 34% on mIoU.

(4) The complexity of surrounding occupancy perception lies in the computational burden of high-resolution 3D predictions, which can be alleviated by the proposed CONet. The volumetric size ($40 \times 512 \times 512$) of the ground truth occupancy in our benchmark is $\sim 5\times$ larger than that of [1], and directly predicting high-resolution occupancy is computationally unfeasible. For efficiency, the proposed baselines produce low-resolution results, yet the performance is restricted. Therefore, we propose CONet to efficiently refine the low-resolution prediction. Notably, the CONet built upon camera-based, LiDAR-based and multi-modal baselines relatively improves the mIoU by 25%, 34% and 29% with marginal latency overhead (efficiency comparison is in Tab. 4). Additionally, we provide visualization (see Fig. 5) to verify that the CONet can generate fine-grained occupancy results based on coarse predictions.

5.3. Baselines under Different Settings

In this subsection, we analyze baseline performance under different experiment settings (*e.g.*, input size, backbone selection, fusion method), and the results are shown in Tab. 3. For the camera-based baseline, using a larger input size (1600×900) relatively improves IoU and mIoU by 15% and 21%. Besides, replacing ResNet50 with ResNet101 further enhances mIoU by 8%. For the LiDAR-based baseline, it is observed that utilizing multi-sweeps as input (following [45, 44, 20], 10 sweeps are used) relatively improves the single-sweep counterpart by 26% and 6% on IoU and mIoU. For the multi-modal baseline, the *concatenation* and *add* operations are suboptimal for feature fusion. In contrast, the proposed adaptive fusion dynamically integrates features from two modalities, which relatively enhances the mIoU by 7%.

Method	2D Backbone	Input Size	Fusion	IoU	mIoU
C	R-50	704 × 256	-	15.6	9.0
C	R-50	1600 × 900	-	17.9	10.9
C	R-101	1600 × 900	-	19.1	11.8
L	-	1 sweep	-	17.7	11.2
L	-	10 sweeps	-	22.3	11.9
M	R-50	1600 × 900 10 sweeps	Cat.	23.0	14.8
M	R-50	1600 × 900 10 sweeps	Add.	22.9	14.9
M	R-50	1600 × 900 10 sweeps	Adaptive	23.5	15.9

Table 3: Ablation study on the proposed baselines, where *C,L,M* denotes camera, LiDAR and multi-modal, and *Cat.* represents the *concatenation*.

5.4. Efficiency and Effectiveness of CONet

For efficiency, the proposed baselines generate low-resolution predictions (*i.e.*, the stride parameter S is set as 4, and the output volumetric size is $(10 \times 128 \times 128)$). As shown in Tab. 4, using a smaller stride parameter (*e.g.*,

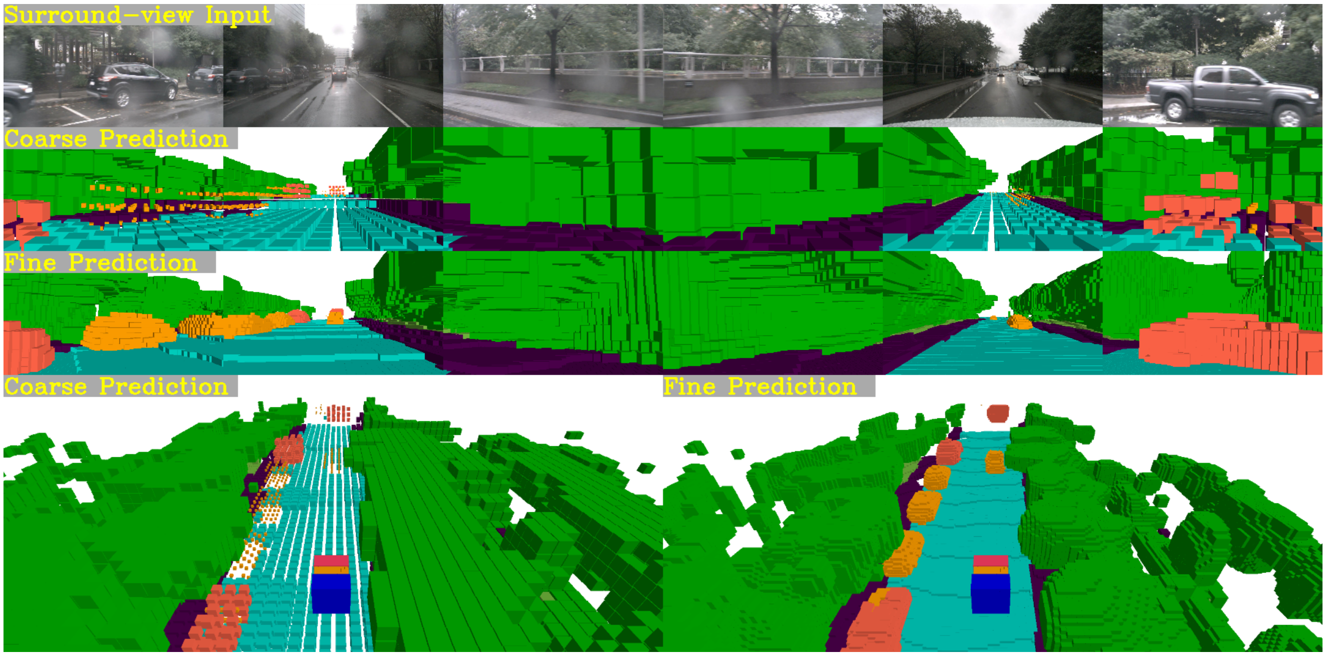

Figure 5: Visualization of the semantic occupancy predictions, where the $1st$ row is surround-view images. In $2nd$ and $3rd$ rows, we show the camera view of coarse and fine occupancy generated by the multi-modal baseline and multi-modal CONet. In the $4th$ row, we compare their global-view predictions.

Method	GPU Mem.	GFLOPs	IoU	mIoU
C-baseline ($S=4$)	17 GB	2241	17.9	10.9
C-baseline ($S=2$)	35 GB	6677	19.3	12.9
C-CONet	22 GB	2371	21.6	13.6
L-baseline ($S=4$)	7.5 GB	749	22.3	11.9
L-baseline ($S=2$)	22 GB	5899	29.3	15.3
L-CONet	8.5 GB	810	30.1	15.9
M-baseline ($S=4$)	19 GB	3050	23.5	15.9
M-baseline ($S=2$)	40 GB	13117	26.3	20.1
M-CONet	24 GB	3066	26.5	20.5

Table 4: Efficiency analysis on CONet, where *C,L,M* denotes camera, LiDAR and multi-modal, *GPU Mem.* represents the GPU memory consumption at training phase, and S is the stride parameter that controls the output size.

Method	Sem. Feat.	Geo.Feat.	IoU	mIoU
M-baseline	-	-	23.5	15.9
M-CONet	✓		22.9	12.7
M-CONet		✓	26.2	19.6
M-CONet	✓	✓	26.5	20.5

Table 5: Ablation study on feature sampling strategies of the CONet. *M* represents multi-modal, *Sem. Feat.* and *Geo. Feat.* denotes semantic features and geometric features.

S=2) enhances the performance, yet the training-time GPU memory is $\sim$2× upscaled, and GFLOPs are $\sim$8× upscaled. Therefore, we propose the CONet for efficient surrounding occupancy perception. Compared with high-resolution baselines (S=2), the CONet built upon low-resolution baselines (S=4) achieves better performance on all the metrics. Besides, the CONet reduces $\sim$15 GB training-time GPU memory, and relatively decreases GFLOPs by $\sim$70%. Additionally, we conduct ablation study to investigate the effectiveness of the feature sampling strategy in CONet. As shown in Tab. 5, solely sampling from $F^{\mathcal{S}}$ degrades the performance, as 2D semantic features are insufficient for high-resolution 3D predictions. In contrast, sampling from geometric features $F^{\mathcal{G}}$ can improve the baseline by 23% on mIoU. Notably, combining the two features further enhances the performance, which relatively improves the baseline by 29%.

6. Conclusion

In this paper, we propose OpenOccupancy, which is the first benchmark for surrounding semantic occupancy perception in driving scenarios. Specifically, we introduce the nuScenes-Occupancy, which extends the nuScenes dataset with dense semantic occupancy annotations based on the proposed AAP pipeline. In the OpenOccupancy benchmark, we establish camera-based, LiDAR-based and multi-modal baselines. Additionally, the CONet is proposed to alleviate the computational burden of high-resolution occupancy predictions. Comprehensive experiments are conducted on the OpenOccupancy benchmark, where the results show that camera-based and LiDAR-based baseline are complementary to each other, and multi-modal baseline further enhances the performance by 46% and 34%. Besides, the proposed CONet relatively improves the baseline by ∼30% with minimal latency overhead. We hope the OpenOccupancy benchmark will be beneficial in the development of surrounding semantic occupancy perception.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 1, 2, 3, 4, 5, 7

[2] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018. 5

[3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CVPR*, 2019. 1, 2, 3

[4] Yingjie Cai, Xuesong Chen, Chao Zhang, Kwan-Yee Lin, Xiaogang Wang, and Hongsheng Li. Semantic scene completion via integrating instances and scene in-the-loop. In *CVPR*, 2021. 3, 4

[5] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022. 3, 4, 5, 6, 7

[6] Ming-Fang Chang, Deva Ramanan, James Hays, John Lambert, Patsorn Sangkloy, Jasvinder A. Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter W. Carr, and Simon Lucey. Argoverse: 3d tracking and forecasting with rich maps. *CVPR*, 2019. 1

[7] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior. In *CVPR*, 2020. 3, 4, 6, 7

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 1, 2

[9] Aloisio Dourado, Teofilo E De Campos, Hansung Kim, and Adrian Hilton. Edgenet: Semantic scene completion from a single rgb-d image. In *ICPR*, 2021. 3, 4

[10] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J Brostow. Structured prediction of unobserved voxels from a single depth image. In *CVPR*, 2016. 1, 2

[11] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *RA-L*, 2022. 1, 2, 3, 4

[12] Martin Garbade, Yueh-Tung Chen, Johann Sawatzky, and Juergen Gall. Two stream 3d semantic scene completion. In *CVPR Workshops*, 2019. 3

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1

[14] David Griffiths and Jan Boehm. Synthcity: A large scale synthetic point cloud. *arXiv preprint arXiv:1907.04758*, 2019. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[16] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *3DV*, 2016. 1, 2

[17] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 6

[18] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2302.07817*, 2023. 3, 6, 7

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *NeurIPS*, 28, 2015. 5

[20] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *CVPR*, 2018. 7

[21] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *CVPR*, 2020. 3, 4, 6, 7

[22] Jie Li, Yu Liu, Dong Gong, Qinfeng Shi, Xia Yuan, Chunxia Zhao, and Ian Reid. Rgbd based dimensional decomposition residual network for 3d semantic scene completion. In *CVPR*, 2019. 3, 4

[23] Jie Li, Yu Liu, Xia Yuan, Chunxia Zhao, Roland Siegwart, Ian Reid, and Cesar Cadena. Depth based semantic scene completion with position importance aware loss. *RA-L*, 2019. 3, 4

[24] Siqi Li, Changqing Zou, Yipeng Li, Xibin Zhao, and Yue Gao. Attention-based multi-modal fusion network for semantic scene completion. In *AAAI*, 2020. 3, 4

[25] 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. *arXiv preprint arXiv:2206.10092*, 2022. 5

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 4

[27] Shice Liu, Yu Hu, Yiming Zeng, Qiankun Tang, Beibei Jin, Yinhe Han, and Xiaowei Li. See and think: Disentangling semantic scene completion. *NeurIPS*, 31, 2018. 3, 4

[28] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 4

[29] nuScenes Contributors. The devkit of the nuscenes dataset. https://github.com/nutonomy/nuscenes-devkit, 2019. 3

[30] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IV*, 2020. 2

[31] Christoph B Rist, David Emmerichs, Markus Enzweiler, and Dariu M Gavrila. Semantic scene completion using local deep implicit functions on lidar data. *TPAMI*, 2021. 3

[32] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *3DV*, 2020. 3, 6, 7

[33] Luis Roldao, Raoul De Charette, and Anne Verroust-Blondet. 3d semantic scene completion: A survey. *IJCV*, 2022. 2

[34] O. Scheel, L. Bergamini, M. Woczyk, B Osiński, and P. Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients. *CoRL*, 2021. 1

[35] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. *ECCV*, 2012. 1, 2

[36] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 1, 2, 3, 4

[37] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 3, 4

[38] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay K. Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *CVPR*, 2020. 1

[39] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Forknet: Multi-branch volumetric semantic completion from a single depth image. In *ICCV*, 2019. 3, 4

[40] Shun-Cheng Wu, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scfusion: Real-time incremental scene reconstruction with semantic completion. In *3DV*, 2020. 3

[41] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013. 1, 2

[42] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *CVPR*, 2020. 4

[43] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI*, 2021. 3, 6, 7

[44] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018. 4, 7

[45] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021. 7

[46] Jiahui Zhang, Hao Zhao, Anbang Yao, Yurong Chen, Li Zhang, and Hongen Liao. Efficient semantic scene completion network with spatial group convolution. In *ECCV*, 2018. 3

[47] Pingping Zhang, Wei Liu, Yinjie Lei, Huchuan Lu, and Xiaoyun Yang. Cascaded context pyramid for full-resolution 3d semantic scene completion. In *ICCV*, 2019. 3

[48] Min Zhong and Gang Zeng. Semantic point completion network for 3d semantic scene completion. In *ECAI*. 2020. 3

[49] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 4