Context and Geometry Aware Voxel Transformer for Semantic Scene Completion

Zhu Yu[1] **Runmin Zhang**[1] **Jiacheng Ying**[1] **Junchen Yu**[1]
Xiaohai Hu[3] **Lun Luo**[4] **Si-Yuan Cao**[2,1*] **Hui-Liang Shen**[1*]
[1]Zhejiang University [2]Ningbo Innovation Center, Zhejiang University
[3]University of Washington [4]HAOMO.AI Technology Co., Ltd.
https://github.com/pkqbajng/CGFormer

Abstract

Vision-based Semantic Scene Completion (SSC) has gained much attention due to its widespread applications in various 3D perception tasks. Existing sparse-to-dense approaches typically employ shared context-independent queries across various input images, which fails to capture distinctions among them as the focal regions of different inputs vary and may result in undirected feature aggregation of cross-attention. Additionally, the absence of depth information may lead to points projected onto the image plane sharing the same 2D position or similar sampling points in the feature map, resulting in depth ambiguity. In this paper, we present a novel context and geometry aware voxel transformer. It utilizes a context aware query generator to initialize context-dependent queries tailored to individual input images, effectively capturing their unique characteristics and aggregating information within the region of interest. Furthermore, it extend deformable cross-attention from 2D to 3D pixel space, enabling the differentiation of points with similar image coordinates based on their depth coordinates. Building upon this module, we introduce a neural network named CGFormer to achieve semantic scene completion. Simultaneously, CGFormer leverages multiple 3D representations (i.e., voxel and TPV) to boost the semantic and geometric representation abilities of the transformed 3D volume from both local and global perspectives. Experimental results demonstrate that CGFormer achieves state-of-the-art performance on the SemanticKITTI and SSCBench-KITTI-360 benchmarks, attaining a mIoU of 16.87 and 20.05, as well as an IoU of 45.99 and 48.07, respectively. Remarkably, CGFormer even outperforms approaches employing temporal images as inputs or much larger image backbone networks.

1 Introduction

Semantic Scene Completion (SSC) aims to jointly infer the complete scene geometry and semantics. It serves as a crucial step in a wide range of 3D perception tasks such as autonomous driving [11, 24], robotic navigation [42, 15], mapping and planning [34]. SSCNet [40] initially formulates the semantic scene completion task. Subsequently, many LiDAR-based approaches [5, 51, 38] have been proposed, but these approaches usually suffer from high-cost sensors.

Recently, there has been a shift towards vision-based SSC solutions. MonoScene [3] lifts the input 2D images to 3D volumes by densely assigning the same 2D features to both visible and occluded regions, leading to many ambiguities. With advancements in bird's-eye-view (BEV) perception [24, 54], transformer-based approaches [13, 47, 23] achieve feature lifting by projecting 3D queries from 3D

[*] Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

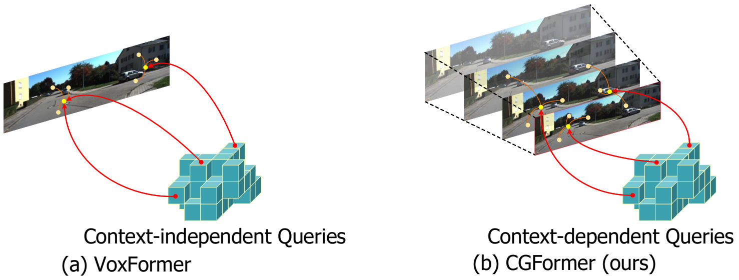


Figure 1: Comparison of feature aggregation. (a) VoxFormer [23] employs a set of shared context-independent queries for different input images, which fails to capture distinctions among them and may lead to undirected feature aggregation. Besides, due to the ignorance of depth information, multiple 3D points may be projected to the same 2D point, causing depth ambiguity. (b) Our CGFormer initializes the voxel queries based on individual input images, effectively capturing their unique features and aggregating information within the region of interest. Furthermore, the deformable cross-attention is extended from 2D to 3D pixel space, enabling the points with similar image coordinates to be distinguished based on their depth coordinates.

space to image plane and aggregating 3D features through deformable attention mechanisms [61]. Among these, VoxFormer [23] introduces a sparse-to-dense architecture, which first aggregates 3D information for the visible voxels using the depth-based queries and then completes the 3D information for non-visible regions by leveraging the reconstructed visible areas as starting points. Building upon VoxFormer [23], the following approaches further improve performance through self-distillation training strategy [43], extracting instance features from images [14], or incorporating an image-conditioned cross-attention module [60].

Despite significant progress, existing sparse-to-dense approaches typically employ shared voxel queries (referred to as context-independent queries) across different input images. These queries are defined as a set of learnable parameters that are independent of the input context. Once the training process is completed, these parameters remain constant for all input images during inference, failing to capture distinctions among various input images, as the focal regions of different images vary. Additionally, these queries also encounter the issue of undirected feature aggregation in cross-attention, where the sampling points may fall in irrelevant regions. Besides, when projecting the 3D points onto the image plane, many points may end with the same 2D position with similar sampling points in the 2D feature map, causing a crucial depth ambiguity problem. An illustrative diagram is presented in Fig. 1(a).

In this paper, we propose a context and geometry aware voxel transformer (CGVT) to lift the 2D features. We observe that context-dependent query tend to aggregate information from the points within the region of interest. Fig. 3 presents an example of the sampling points of the context-dependent queries. Thus, before the aggregation of cross-attention, we first utilizes a context aware query generator to predict context-aware queries from the input individual images. Additionally, we extend deformable cross-attention from 2D to 3D pixel space, which allows points ending with similar image coordinates to be distinguished by their depth coordinates, as illustrated in Fig. 1(b). Furthermore, we propose a simple yet efficient depth refinement block to enhance the accuracy of estimated depth probability. This involves incorporating a more precise estimated depth map from a pretrained stereo depth estimation network [39], avoiding the heavy computational burden as observed in StereoScene [16].

Based on the aforementioned module, we devise a neural network, named as CGFormer. To enhance the obtained 3D volume from the view transformation we integrate multiple representation, *i.e.*, voxel and tri-perspective view (TPV). The TPV representation offers a global perspective that encompasses more high-level semantic information, while the voxel representation focuses more on the fine-grained structures. Drawing from the above analyses, we propose a 3D local and global encoder (LGE) that dynamically fuses the results of the voxel-based and TPV-based branches, to further enhance the 3D features from both local and global perspectives.

Our contributions can be summarized as follows:

- We propose a context and geometry aware voxel transformer (CGVT) to improve the performance of semantic scene completion. This module initializes the queries based on the context of individual input images and extends the deformable cross-attention from 2D to 3D pixel space, thereby improving the performance of feature lifting.

- We introduce a simple yet effective depth refinement block to enhance the accuracy of estimated depth probability with only introducing minimal computational burden.
- We devise a 3D local and global encoder (LGE) to strengthen the semantic and geometric discriminability of the 3D volume. This encoder employs various 3D representations (voxel and TPV) to encode the 3D features, capturing information from both local and global perspectives.
- Benefiting from the aforementioned modules, our CGFormer attains state-of-the-art results with a mIoU of 16.63 and an IoU of 44.41 on SemanticKITTI, as well as a mIoU of 20.05 and an IoU of 48.07 on SSCBench-KITTI-360. Notably, our method even surpasses methods employing temporal images as inputs or using much larger image backbone networks.

2 Related Work

2.1 Vision-based 3D Perception

Vision-based 3D perception [24, 25, 31, 20, 12, 45, 62, 17, 44, 19, 58, 48, 52, 53, 56] has received extensive attention due to its ease of deployment, cost-effectiveness, and the preservation of intricate visual attributes, emerging as a crucial component in the autonomous driving. Current research efforts focus on constructing unified 3D representations (*e.g.,* BEV, TPV, voxel) from input images. LiftSplat [36] lifts image features by performing outer product between the 2D image features and their estimated depth probability to generate a frustum-shaped pseudo point cloud of contextual features. The pseudo point cloud features are then splatted to predefined 3D anchors through a voxel-pooling operation. Building upon this, BEVDet [12] extends LiftSplat to 3D object detection, while BEVDepth [21] further enhances performance by introducing ground truth supervision for the estimated depth probability. With the advancements in attention mechanisms [4, 61, 57, 33, 45], BEVFormer [24, 54] transforms image features into BEV features using point deformable cross-attention [61]. Additionally, many other methods, such as OFT [37], Petr [28, 29], Inverse Perspective Mapping (IPM) [10], have also been presented to transform 2D image features into a 3D representation.

2.2 Semantic Scene Completion

SSCNet [40] initially introduces the concept of semantic scene completion, aiming to infer the semantic voxels. Following methods [38, 18, 51, 5] commonly utilize explicit depth maps or LIDAR point clouds as inputs. MonoScene [3] is the pioneering method for directly predicting the semantic occupancy from the input RGB image, which presents a FLoSP module for 2D-3D feature projection. StereoScene[16] introduces explicit epipolar constraints to mitigate depth ambiguity, albeit with heavy computational burden for correlation. TPVFormer [13] introduces a tri-perspective view (TPV) representation to describe the 3D scene, as an alternative to the BEV representation. The elements fall into the field of view aggregates information from the image features using deformable cross-attention [24, 61]. Beginning from depth-based [23] sparse proposal queries, VoxFormer [23] construct the 3D representation in a coarse-to-fine manner. The 3D information from the visible queries are diffused to the overall 3D volume using deformable self-attention, akin to the masked autoencoder (MAE)[8]. HASSC [43] introduces a self-distillation training strategy to improve the performance of VoxFormer [23], while MonoOcc [60] further enhance the 3D volume with an image-conditioned cross-attention module. Symphonize [14] extracts high level instance features from the image feature map, serving as the key and value of the cross-attention.

3 CGFormer

3.1 Overview

As shown in Fig. 2, the overall framework of CGFormer is composed of four parts: feature extraction to extract 2D image features, view transformation (Section 3.2) to lift the 2D features to 3D volumes,

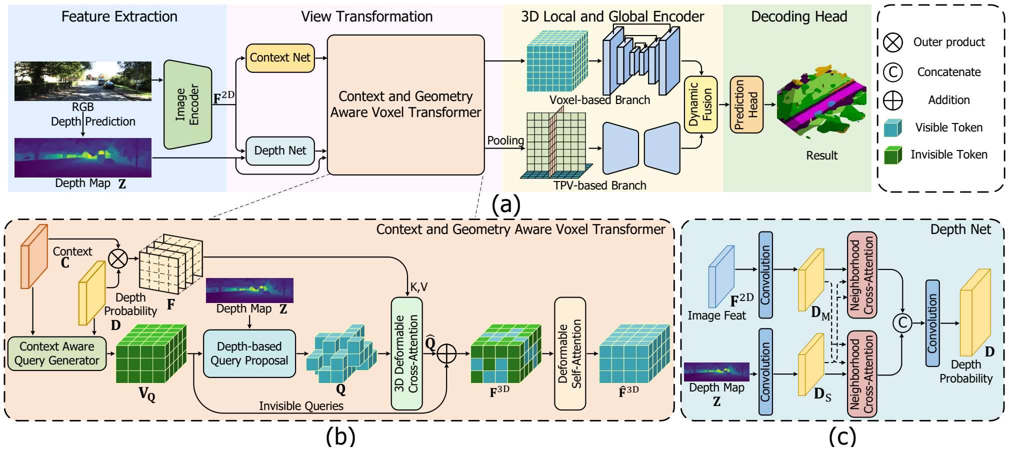


Figure 2: Schematics and detailed architectures of CGFormer. (a) The framework of the proposed CGFormer for camera-based semantic scene completion. The pipeline consists of the image encoder for extracting 2D features, the context and geometry aware voxel (CGVT) transformer for lifting the 2D features to 3D volumes, the 3D local and global encoder (LGE) for enhancing the 3D volumes and a decoding head to predict the semantic occupancy. (b) Detailed structure of the context and geometry aware voxel transformer. (c) Details of the Depth Net.

3D encoder (Section 3.3) to further enhance the semantic and geometric discriminability of the 3D features, and a decoding head to infer the final result.

Image Encoder. The image encoder consists of a backbone network for extracting multi-scale features and a feature pyramid network (FPN) to fuse them. We adopt $\mathbf{F}^{2\mathrm{D}} \in \mathbb{R}^{H \times W \times C}$ to represent the extracted 2D image feature, where C is the channel number, and (H, W) refers to the resolution.

Depth Estimator. In alignment with VoxFormer [23], we use an off-the-shelf stereo depth estimation model [39] to predict the depth $\mathbf{Z}(u, v)$ for each image pixel (u, v). The resulting estimated depth map is then employed to define the visible voxels located on the surface, serving as query proposals. Additionally, it is also used to refine the depth probability for lifting the 2D features.

3.2 View Transformation

A detailed diagram of our context and geometry aware voxel transformer is presented in Fig. 2(b). The process begins with a context aware query generator, which takes the context feature map to generate the context-dependent queries. Subsequently, the visible query proposals are located by the depth map from the pretrained depth estimation network. These proposals then attend to the image features to aggregate semantic and geometry information based on the 3D deformable cross-attention. Finally, the aggregated 3D information is further diffused from the updated proposals to the overall 3D volume via deformable self-attention.

Context-dependent Query Generation. Previous coarse-to-fine approaches [23, 43, 46, 60] typically employ shared context-independent queries for all the inputs. However, these approaches may overlook the differences among different images and aggregate information from irrelevant areas. In contrast, CGFormer first generates context-dependent queries from the image features using a context aware query generator. To elaborate, the extracted $\mathbf{F}^{2D}$ is fed to the context net and the depth net [21] to generate the context feature $\mathbf{C} \in \mathbb{R}^{H \times W \times C}$ and depth probability $\mathbf{D} \in \mathbb{R}^{H \times W \times D}$, respectively. Then the query generator f takes $\mathbf{C}$ and $\mathbf{D}$ as inputs to generate voxel queries $\mathbf{V_Q} \in \mathbb{R}^{X \times Y \times Z \times C}$

$$\mathbf{V_Q} = f(\mathbf{C}, \mathbf{D}), \tag{1}$$

where (X, Y, Z) denotes the spatial resolution of the 3D volume. Benefiting from the initialization, the sampling points of the context-dependent queries usually locate within the region of interest. An example of the deformable sampling points is displayed in Fig. 3.

Depth-based Query Proposal. Following VoxFormer [23], we determine the visible voxels through the conversion of camera coordinates to world coordinates using the pre-estimated depth map, except that we do not employ an additional occupancy network [38]. A pixel (u, v) will be transformed to a 3D point (x, y, z), utilizing the camera intrinsic and extrinsic matrices ($K \in \mathbb{R}^{4\times4}$, $E \in \mathbb{R}^{4\times4}$).

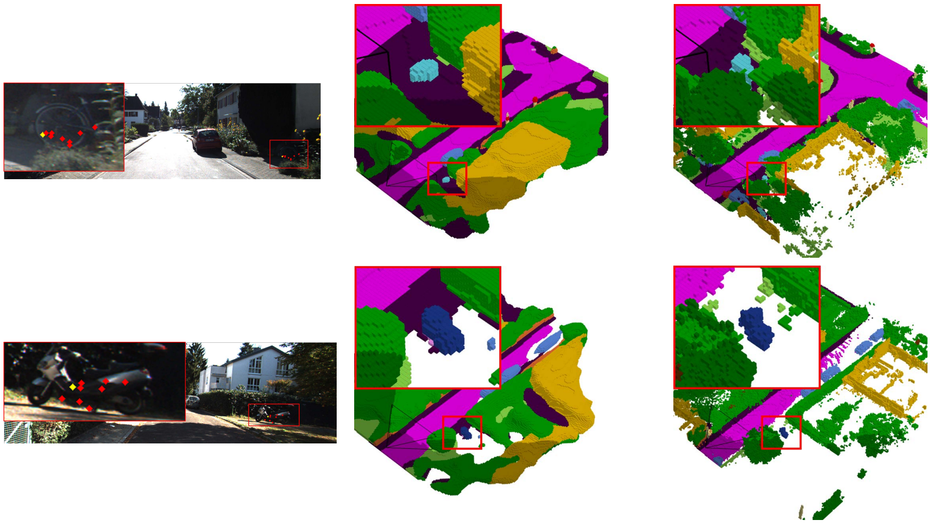


Figure 3: Visualization of the sampling locations for different small objects. The yellow dot represents the query point, while the red dots indicate the locations of the deformable sampling points. The sampling points of the context-dependent query (a) tend to be distributed within the regions of interest. Beneficial from this, CGFormer achieve better performance than previous methods.

The synthetic point cloud is further converted into a binary voxel grid map $\mathbf{M}$, where each voxel is masked as 1 if occupied by at least one point and 0 otherwise. The query proposals are selected based on the binary $\mathbf{M}$ as

$$\mathbf{Q} = \mathbf{V}_{\mathbf{Q}}[\mathbf{M}]. \tag{2}$$

3D Deformable Cross-Attention. The query proposals then attend to the image features to aggregate the visual features of the 3D scene. We first expand the dimension of $\mathbf{C}$ from 2D to 3D pixel space by conducting the outer product between $\mathbf{D}$ and $\mathbf{C}$, formulated as $\mathbf{F} = \mathbf{C} \otimes \mathbf{D}$. Here, the operator $\otimes$ refers to the outer product conducted at the last dimension. Taking $\mathbf{F}$ as key and value, for a 3D query $\mathbf{Q}_p$ located at position p, the 3D deformable cross-attention mechanism can be formulated as

$$\hat{\mathbf{Q}}_p = \text{3D-DCA}(\mathbf{Q}_p, \mathbf{F}, p) = \sum_{n=1}^{N} A_n W \phi(\mathbf{F}, \mathcal{P}(p) + \Delta p), \tag{3}$$

where n indexes the sampled point from a total of N points, $\mathcal{P}(p)$ denotes camera projection function to obtain the reference points in the 3D pixel space, $A_n \in [0, 1]$ is the learnable attention weight, and W denotes the projection weight. $\Delta p \in \mathbb{R}^3$ is the predicted offset to the reference point p, and $\phi(\mathbf{F}, \mathcal{P}_d(p) + \Delta p_d)$ indicates the trilinear interpolation used to sample features in the expanded 3D feature map $\mathbf{F}$. $\hat{\mathbf{Q}}$ denotes the updated query proposals. For simplicity, we only show the formulation of single-head attention.

Deformable Self-Attention. After several layers of deformable cross-attention, we merge $\hat{\mathbf{Q}}$ and the invisible elements of $\mathbf{V}_{\mathbf{Q}}$, indicated as $\mathbf{F}^{\text{3D}}$. The information of the updated query proposals is diffused to the overall 3D volume via deformable self-attention and $\hat{\mathbf{F}}^{\text{3D}}$ indicates the updated 3D volume. This process of a query located at position p can be formulated as

$$\hat{\mathbf{F}}^{\text{3D}}_p = \text{DSA}(\mathbf{F}^{\text{3D}}_p, \mathbf{F}^{\text{3D}}, p) = \sum_{n=1}^{N} A_n W \phi(\mathbf{F}^{\text{3D}}, p + \Delta p). \tag{4}$$

Depth Net. The depth probability serves as the inputs of both the context aware query generator and 3D deformable cross-attention, whose accuracy greatly influences the performance. The results of single image depth estimation are usually unsatisfactory. Incorporating epipolar constraint can help estimate more accurate depth map. However, computing correlation to achieve this may introduce much computation burden during the estimation of the semantic voxels. Instead, we introduce a simple yet effective depth refinement strategy. The detailed architecture of the depth net is shown in Fig. 2(c). We first estimate monocular depth feature $\mathbf{D}_{\mathrm{M}}$ from $\mathbf{F}^{\mathrm{2D}}$. The depth map generated from the stereo method [39] is encoded as stereo feature $\mathbf{D}_{\mathrm{S}}$ by several convolutions. These two features are further processed by cross-attention blocks for information interaction. To save memory, the window size of cross-attention is restricted within a range of the nearest neighboring pixels [7], which can be formulated as

$$\begin{aligned}\hat{\mathbf{D}}_{\mathrm{M}} &= \psi(\mathbf{D}_{\mathrm{M}}, \mathbf{D}_{\mathrm{S}}, \mathbf{D}_{\mathrm{S}}),\\ \hat{\mathbf{D}}_{\mathrm{S}} &= \psi(\mathbf{D}_{\mathrm{S}}, \mathbf{D}_{\mathrm{M}}, \mathbf{D}_{\mathrm{M}}),\end{aligned} \tag{5}$$

where $\hat{\mathbf{D}}_{\mathrm{M}}$ and $\hat{\mathbf{D}}_{\mathrm{S}}$ denote the enhanced $\mathbf{D}_{\mathrm{M}}$ and $\mathbf{D}_{\mathrm{S}}$, and ψ denotes the neighborhood attention [7]. We set the window size to 5×5. Finally, these two volumes are concatenated and fed to estimate the final depth probability. This strategy can boost the accuracy of the depth probability a lot with only minimal computation cost.

3.3 3D Local and Global Encoder

The TPV representation is derived by compressing the voxel representation along its three axes. TPV focuses more on the global high-level semantic information, whereas the voxel emphasizes more on the details of intricate structures. We combine these two representations to enhance representation capability of the 3D features. As shown in Fig. 2(a), our 3D local and global encoder consists of the voxel-based and TPV-based branches. With the input 3D volume, the two branches first aggregate features in parallel. The dual-branch outputs are then fused dynamically and fed to the final decoding block.

Voxel-based Branch. The voxel-based branch aims to enhance the fine-grained structure features of the 3D volume. We adopt a lightweight ResNet [9] with 3D convolutions to extract multi-scale voxel features from $\hat{\mathbf{F}}^{\mathrm{3D}}$ and merge them into $\mathbf{F}^{\mathrm{3D}}_{\mathrm{voxel}} \in \mathbb{R}^{X \times Y \times Z \times C}$ using a 3D feature pyramid network.

TPV-based Branch. To obtain three-plane features of the TPV representation, we apply spatial pooling to the voxel features along three axes. Directly performing MaxPooling may cause the loss of the fine-grained details. Considering the balance between performance and efficiency, we apply the group spatial to channel operation [50, 25, 62] to transform the voxel representation to TPV representation. Specifically, we split the voxel features along the pooling axis into K groups and apply the MaxPooling within each group. Then we concatenate each group features along the channel dimension and use convolutions to reduce the channel dimension into C. Denote the 2D view features as $\mathbf{F}^{\mathrm{3D}}_{XY} \in \mathbb{R}^{X \times Y \times C}$, $\mathbf{F}^{\mathrm{3D}}_{XZ} \in \mathbb{R}^{X \times Z \times C}$, $\mathbf{F}^{\mathrm{3D}}_{YZ} \in \mathbb{R}^{Y \times Z \times C}$, the above process can be formulated as

$$\begin{aligned}\mathbf{F}^{\mathrm{3D}}_{XY} &= \mathrm{Conv}_{XY}(\mathrm{Concat}(\{\mathrm{Pooling}_{\{Z,i\}}(\hat{\mathbf{F}}^{\mathrm{3D}})\}_{i=1}^{K})),\\ \mathbf{F}^{\mathrm{3D}}_{YZ} &= \mathrm{Conv}_{YZ}(\mathrm{Concat}(\{\mathrm{Pooling}_{\{X,i\}}(\hat{\mathbf{F}}^{\mathrm{3D}})\}_{i=1}^{K})),\\ \mathbf{F}^{\mathrm{3D}}_{XZ} &= \mathrm{Conv}_{XZ}(\mathrm{Concat}(\{\mathrm{Pooling}_{\{Y,i\}}(\hat{\mathbf{F}}^{\mathrm{3D}})\}_{i=1}^{K}))\end{aligned} \tag{6}$$

After obtaining the TPV features, we employ a 2D image backbone to process each plane to produce corresponding multi-scale features, which are then fused by a 2D feature pyramid network. For simplicity, we leverage $\mathbf{F}^{\mathrm{3D}}_{XY}, \mathbf{F}^{\mathrm{3D}}_{XZ}, \mathbf{F}^{\mathrm{3D}}_{YZ}$ to represent the outputs of these networks, as well.

Dynamic Fusion. The dual-path outputs are fused dynamically. We generate aggregation weights $\mathbf{W} \in \mathbb{R}^{X \times Y \times Z \times 4}$ from $\mathbf{F}^{\mathrm{3D}}_{\mathrm{voxel}}$ using a convolution with a softmax on the last dimension. The final 3D feature volume $\mathbf{F}^{\mathrm{3D}}_{\mathrm{final}}$ is computed as

$$\mathbf{F}^{\mathrm{3D}}_{\mathrm{final}} = \sum_{i}^{4} \mathbf{W}_i \odot \mathbf{F}^{\mathrm{3D}}_{i}, \tag{7}$$

where $\mathbf{W}_i \in \mathbb{R}^{X \times Y \times Z \times 1}$ is a slice of the $\mathbf{W}$ and $\mathbf{F}^{\mathrm{3D}}_{i}$ is an element of the set $[\mathbf{F}^{\mathrm{3D}}_{\mathrm{voxel}}, \mathbf{F}^{\mathrm{3D}}_{XY}, \mathbf{F}^{\mathrm{3D}}_{XZ}, \mathbf{F}^{\mathrm{3D}}_{YZ}]$. The operation $\odot$ refers to the element multiplication and the dimension broadcasting of the features from TPV-based branches is omitted in the equation for conciseness. $\mathbf{F}^{\mathrm{3D}}_{\mathrm{final}}$ is finally fed to the decoding block to infer the complete scene geometry and semantics. Please refer to the supplementary material for more detailed schematics of the 3D local and global encoder.

3.4 Training Loss

To train the proposed CGFormer, cross-entropy loss weighted by class frequencies $\mathcal{L}_{ce}$ is leveraged to optimize the network. Following MonoScene [3], we also utilize affinity loss $\mathcal{L}_{scal}^{geo}$ and $\mathcal{L}_{scal}^{sem}$ to

Table 1: Quatitative results on SemanticKITTI [1] test set. * represents the reproduced results in [13, 59]. The best and the second best results are in **bold** and underlined, respectively.

Method	IoU	mIoU	road (15.30%)	sidewalk (11.13%)	parking (1.12%)	other-grnd. (0.56%)	building (14.1%)	car (3.92%)	truck (0.16%)	bicycle (0.03%)	motorcycle (0.03%)	other-veh. (0.20%)	vegetation (39.3%)	trunk (0.51%)	terrain (9.17%)	person (0.07%)	bicyclist (0.07%)	motorcyclist (0.05%)	fence (3.90%)	pole (0.29%)	traf.-sign (0.08%)
MonoScene* [3]	34.16	11.08	54.70	27.10	24.80	5.70	14.40	18.80	3.30	0.50	0.70	4.40	14.90	2.40	19.50	1.00	1.40	0.40	11.10	3.30	2.10
TPVFormer [13]	34.25	11.26	55.10	27.20	27.40	6.50	14.80	19.20	3.70	1.00	0.50	2.30	13.90	2.60	20.40	1.10	2.40	0.30	11.00	2.90	1.50
SurroundOcc [47]	34.72	11.86	56.90	28.30	30.20	6.80	15.20	20.60	1.40	1.60	1.20	4.40	14.90	3.40	19.30	1.40	2.00	0.10	11.30	3.90	2.40
OccFormer [59]	34.53	12.32	55.90	30.30	31.50	6.50	15.70	21.60	1.20	1.50	1.70	3.20	16.80	3.90	21.30	2.20	1.10	0.20	11.90	3.80	3.70
IAMSSC [49]	43.74	12.37	54.00	25.50	24.70	6.90	19.20	21.30	3.80	1.10	0.60	3.90	22.70	5.80	19.40	1.50	2.90	0.50	11.90	5.30	4.10
VoxFormer-S [23]	42.95	12.20	53.90	25.30	21.10	5.60	19.80	20.80	3.50	1.00	0.70	3.70	22.40	7.50	21.30	1.40	2.60	0.20	11.10	5.10	4.90
VoxFormer-T [23]	43.21	13.41	54.10	26.90	25.10	7.30	23.50	21.70	3.60	1.90	1.60	4.10	24.40	8.10	24.20	1.60	1.10	0.00	13.10	6.60	5.70
DepthSSC [55]	**44.58**	13.11	55.64	27.25	25.72	5.78	20.46	21.94	3.74	1.35	0.98	4.17	23.37	7.64	21.56	1.34	2.79	0.28	12.94	5.87	6.23
Symphonize [14]	42.19	15.04	58.40	29.30	26.90	11.70	24.70	23.60	3.20	3.60	**2.60**	5.60	24.20	10.00	23.10	**3.20**	1.90	**2.00**	16.10	7.70	8.00
HASSC-S [43]	43.40	13.34	54.60	27.70	23.80	6.20	21.10	22.80	4.70	1.60	1.00	3.90	23.80	8.50	23.30	1.60	4.00	0.30	13.10	5.80	5.50
HASSC-T [43]	42.87	14.38	55.30	29.60	25.90	11.30	23.10	23.00	2.90	1.90	1.50	4.90	24.80	9.80	26.50	1.40	3.00	0.00	14.30	7.00	7.10
StereoScene [16]	43.34	15.36	61.90	31.20	30.70	10.70	24.20	22.80	2.80	3.40	2.40	6.10	23.80	8.40	27.00	2.90	2.20	0.50	16.50	7.00	7.20
H2GFormer-S [46]	44.20	13.72	56.40	28.60	26.50	4.90	22.80	23.40	4.80	0.80	0.90	4.10	24.60	9.10	23.80	1.20	2.50	0.10	13.30	6.40	6.30
H2GFormer-T [46]	43.52	14.60	57.90	30.40	30.00	6.90	24.00	23.70	5.20	0.60	1.20	5.00	**25.20**	10.70	25.80	1.10	0.10	0.00	14.60	7.50	**9.30**
MonoOcc-S [60]	-	13.80	55.20	27.80	25.10	9.70	21.40	23.20	5.20	2.20	1.50	5.40	24.00	8.70	23.00	1.70	2.00	0.20	13.40	5.80	6.40
MonoOcc-L [60]	-	15.63	59.10	30.90	27.10	9.80	22.90	23.90	**7.20**	**4.50**	2.40	**7.70**	25.00	9.80	26.10	2.80	**4.70**	0.60	16.90	7.30	8.40
CGFormer (ours)	44.41	**16.63**	**64.30**	**34.20**	**34.10**	**12.10**	**25.80**	**26.10**	4.30	3.70	1.30	2.70	24.50	**11.20**	**29.30**	1.70	3.60	0.40	**18.70**	**8.70**	**9.30**

Table 2: Quantitative results on SSCBench-KITTI360 test set. The results for counterparts are provided in [22]. The best and the second best results for all camera-based methods are in **bold** and underlined, respectively. The best results from the LiDAR-based methods are in red.

Method	IoU	mIoU	car (2.85%)	bicycle (0.01%)	motorcycle (0.01%)	truck (0.16%)	other-veh. (5.75%)	person (0.02%)	road (14.98%)	parking (2.31%)	sidewalk (6.43%)	other-grnd. (2.05%)	building (15.67%)	fence (0.96%)	vegetation (41.99%)	terrain (7.10%)	pole (0.22%)	traf.-sign (0.06%)	other-struct. (4.33%)	other-obj. (0.28%)
LiDAR-based methods																				
SSCNet [40]	53.58	16.95	31.95	0.00	0.17	10.29	0.00	0.07	65.70	17.33	41.24	3.22	44.41	6.77	43.72	28.87	0.78	0.75	8.69	0.67
LMSCNet [38]	47.35	13.65	20.91	0.00	0.00	0.26	0.58	0.00	62.95	13.51	33.51	0.20	43.67	0.33	40.01	26.80	0.00	0.00	3.63	0.00
Camera-based methods																				
MonoScene [3]	37.87	12.31	19.34	0.43	0.58	8.02	2.03	0.86	48.35	11.38	28.13	3.32	32.89	3.53	26.15	16.75	6.92	5.67	4.20	3.09
TPVFormer [13]	40.22	13.64	21.56	1.09	1.37	8.06	2.57	2.38	52.99	11.99	31.07	3.78	34.83	4.80	30.08	17.52	7.46	5.86	5.48	2.70
OccFormer [59]	40.27	13.81	22.58	0.66	0.26	9.89	3.82	2.77	54.30	13.44	31.53	3.55	36.42	4.80	31.00	19.51	7.77	8.51	6.95	4.60
VoxFormer [23]	38.76	11.91	17.84	1.16	0.89	4.56	2.06	1.63	47.01	9.67	27.21	2.89	31.18	4.97	28.99	14.69	6.51	6.92	3.79	2.43
IAMSSC [49]	41.80	12.97	18.53	2.45	1.76	5.12	3.92	3.09	47.55	10.56	28.35	4.12	31.53	6.28	29.17	15.24	8.29	7.01	6.35	4.19
DepthSSC [55]	40.85	14.28	21.90	2.36	4.30	11.51	4.56	2.92	50.88	12.89	30.27	2.49	37.33	5.22	29.61	21.59	5.97	7.71	5.24	3.51
Symphonies [14]	44.12	18.58	**30.02**	1.85	**5.90**	**25.07**	**12.06**	**8.20**	54.94	13.83	32.76	**6.93**	35.11	**8.58**	38.33	11.52	14.01	9.57	**14.44**	**11.28**
CGFormer (ours)	**48.07**	**20.05**	29.85	**3.42**	3.96	17.59	6.79	6.63	**63.85**	**17.15**	**40.72**	5.53	**42.73**	8.22	**38.80**	**24.94**	**16.24**	**17.45**	10.18	6.77

optimize the scene-wise and class-wise metrics (geometric IoU and semantic mIoU). We also employ explicit depth loss $\mathcal{L}_d$ [24] to supervise the estimated depth probability of the depth net. Hence, the overall loss function can be formulated as

$$\mathcal{L} = \lambda_d \mathcal{L}_d + \mathcal{L}_{ce} + \mathcal{L}_{scal}^{geo} + \mathcal{L}_{scal}^{sem}, \tag{8}$$

where we set the weight of depth loss λ_d to 0.001.

4 Experiments

4.1 Quantitative Results

We conducted experiments to evaluate the performance of our CGFormer with the latest state-of-the-art methods on the SemanticKITTI [1] and SSCBench-KITTI-360 [22]. Refer to the supplement material for information of datasets, metrics and detailed implementation details.

We list the results on SemanticKITTI hidden test set in Table 1. CGFormer achieves a mIoU of 16.63 and an IoU of 44.41. Notably, CGFormer outperforms all competing methods in terms of mIoU, demonstrating its superior performance. Regarding IoU, CGFormer surpasses all other methods except DepthSSC [55]. While ranking 2nd in terms of IoU, CGFormer exhibits only a slight marginal difference of 0.17 in comparison to DepthSSC [55]. However, CGFormer significantly outperforms DepthSSC [55] by a substantial margin of 3.52 in terms of mIoU. Additionally, compared to VoxFormer-T [23], HASSC-T [43], H2GFormer-t [46], and MonoOcc-L [3], although they utilize temporal stereo image pairs as inputs or larger image backbone networks, our CGFormer exceeds them a lot in terms of both metrics. For specific instances, CGFormer achieves the best or second-best performance on most of the classes, such as road, sidewalk, parking, and building.

Table 3: Ablation study of the architectural components on SemanticKITTI [1] validation set. CGVT: context and geometry aware voxel transformer. LGE: local and global encoder. 3D-DCA: 3D deformable cross attention. CAQG: context aware query generator. LB: local voxel-based branch. $\mathcal{T}_{XY}$, $\mathcal{T}_{YZ}$, $\mathcal{T}_{XZ}$: planes of the TPV-based branch. DF: dynamic fusion. There are 32M predefined parameters.

Method	CGVT		LGE					IoU↑	mIoU↑	Params (M)	Memory (M)
	3D-DCA	CAQG	LB	$\mathcal{T}_{XY}$	$\mathcal{T}_{YZ}$	$\mathcal{T}_{XZ}$	DF				
Baseline								37.99	12.71	76.57	13222
(a)	✓							40.14	14.34	86.17	15150
(b)	✓	✓						42.86	15.60	86.19	15488
(c)	✓	✓	✓					44.84	16.41	93.78	17843
(d)	✓	✓	✓	✓	✓	✓		44.63	16.54	122.42	19188
(e)	✓	✓	✓	✓			✓	45.46	16.38	122.12	19024
(f)	✓	✓	✓		✓		✓	45.53	16.74	122.12	18912
(g)	✓	✓	✓			✓	✓	45.71	16.49	122.12	18912
(h)	✓	✓	✓	✓	✓	✓	✓	**45.99**	**16.87**	122.42	19330

Table 4: Ablation on the choices of context aware query generator.

Method	IoU	mIoU	Params (M)	Memory (M)
w/o CAQG	40.14	14.34	86.17	15150
More attention layers	41.83	14.43	86.97	21596
FLoSP [3]	41.54	14.66	86.19	15907
Voxel Pooling	**42.86**	**15.60**	86.19	15488

Table 5: Ablation on the depth refinement block of the depth net.

Model	IoU	mIoU	Params (M)	Memory (M)
w/o stereo feature $\mathbf{D}_S$	44.57	15.08	116.44	18547
w/o neighborhood attn.	45.72	16.26	122.32	19260
w / StereoScene [16]	45.55	16.49	156.50	22347
full model	45.99	16.87	122.42	19330

To demonstrate the versatility of our model, we also conduct experiments on the SSCBench-KITTI-360 [40] dataset and list the results in Table 2. It is observed that CGFormer surpasses all the published camera-based methods by a margin of 3.95 IoU and 1.67 mIoU. Notably, CGFormer even outperforms the two LiDAR-based methods in terms of mIoU. The above analyses further verifies the effectiveness and excellent performance of CGFormer.

4.2 Ablation Study

We conduct ablation analyses on the components of CGFormer on the SemanticKITTI validation set.

Ablation on the context and geometry aware voxel transformer (CGVT). Table 3 presents a detailed analysis of various architectural components within CGFormer. The baseline can be considered as a simplified version of VoxFormer [23], without utilizing the extra occupancy network [38] to generate coarse occupancy masks. It begins with an image encoder to extract image features, a depth-based query proposal layer to define the visible queries, the deformable cross-attention layer to aggregate features for visible areas, the deformable self-attention layer to diffuse features from the visible regions to the invisible ones. Extending the cross-attention to 3D deformable cross-attention (a) brings a notable improvement of 1.63 mIoU and 2.15 IoU. The performance is further enhanced by giving the voxel queries a good initialization by introducing the context ware query generator (b).

Ablation on the local and global encoder (LGE). After obtaining the 3D features, CGFormer incorporates multiple representation to refine the 3D volume form both local and global perspectives. Through experimentation, we validate that the voxel-based and TPV-based branches collectively contribute to performance improvement with a suitable fusion strategy. Specifically, we compate the results of solely utilizing the local voxel-based branch (c), simply adding the results of dual branches (d), and dynamically fusing the dual-branch outputs (h), as detailed in Table 3. The accuracy is notably elevated to an IoU of 45.99 and mIoU of 16.87 by dynamically fusing the dual-branch outputs. Additionally, we conduct an ablation study on the three TPV planes utilized in the TPV-based branch (e,f,g). The results demonstrate that any individual plane improves performance compared to the model with only the local branch. Combining the information from all three planes into the TPV representation achieves superior performance, underscoring the complementary nature of the three TPV planes in effectively representing complex 3D scenes.

Ablation on the context aware query generator. We present the ablation analysis of the context-aware query generator in Table 4. We remove the CAQG and increase the number of attention layers, where the results of the previous layers can be viewed as a initialization of the queries for the later layers. This configuration (6 cross-attention layers and 4 self-attention layers) significantly improves IoU but only marginally lifts mIoU, and it requires much more training memory. Employing

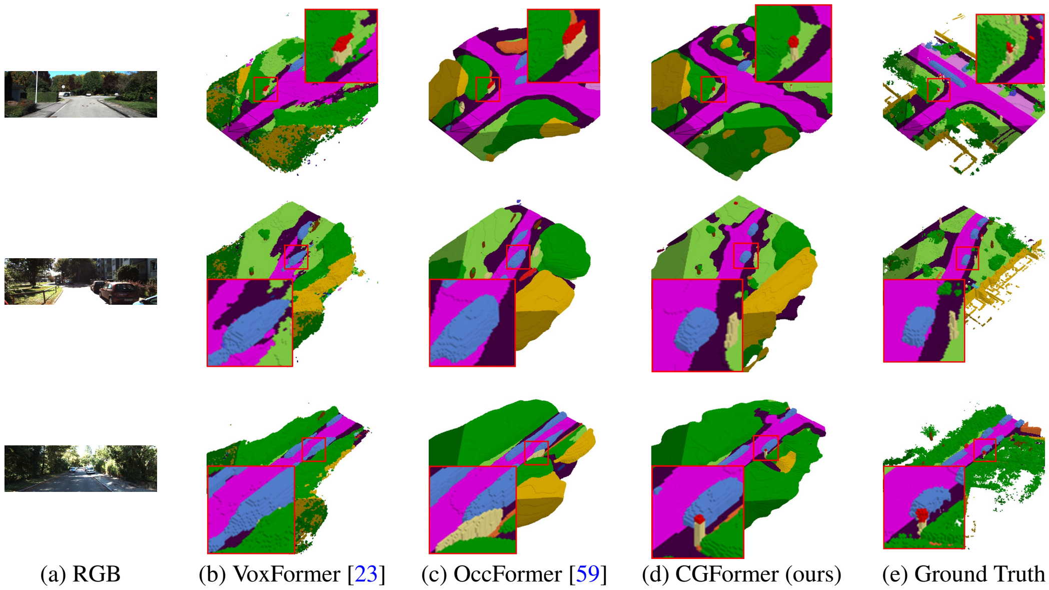


Figure 4: Qualitative visualization results on the SemanticKITTI [1] validation set.

FLoSP [3] with the occupancy-aware depth module [35] can improve performance without much additional computational burden. By replacing it with the voxel pooling [36], the model achieves the best performance. Thus, we employ voxel pooling as our context aware query generator.

Ablation on the depth refinement block. Table 5 presents analyses of the impact of each module within the depth net. By removing the stereo feature $\mathbf{D}_S$ and employing the same structure as BEVDepth [21], we observe a performance drop of 1.42 IoU and 1.79 mIoU. When incorporating the stereo feature $\mathbf{D}_S$ but fusing it with a simple concatenate operation without using the neighborhood attention, the model achieves a mIoU of 45.72 and an IoU of 16.26. These results emphasize that deactivating any component of the depth net leads to a decrease of the accuracy of the full network. Additionally, we replace the depth refinement module with the dense correlation module from StereoScene [16]. Compared to this, our depth refinement module achieves comparable results while using significantly fewer parameters and less training memory.

4.3 Qualitative Results

Fig. 4 presents visualizations of predicted results on the SemanticKITTI validation set obtained from MonoScene[3], VoxFormer [23], OccFormer [59], and our proposed CGFormer. Notably, CGFormer outperforms other methods by effectively capturing the semantic scene and inferring previously invisible regions. The predictions generated by CGFormer exhibit distinctly clearer geometric structures and improved semantic discrimination, particularly for classes like cars and the overall scene layout. This enhancement is attributed to the precision achieved through the context and geometry aware voxel transformer applied in our proposed approach. In contrast, the instances generated by the comparison methods appear to be influenced by depth ambiguity, resulting in vague shapes.

5 Conclusions

In this paper, we present CGFormer, a novel neural network for Semantic Scene Completion. CGFormer dynamically generates distinct voxel queries, which serve as a good starting point for the attention layers, capturing the unique characteristics of various input images. To improve the accuracy of the estimated depth probability, we propose a simple yet efficient depth refinement module, with minimal computational burden. To boost the semantic and geometric representation abilities, CGFormer incorporates multiple representations (voxel and TPV) to encode the 3D volumes from both local and global perspectives. We experimentally show that our CGFormer achieves state-of-the-art performance on the SemanticKITTI and SSCBench-KITTI-360 benchmarks.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2023YFB3209800, in part by Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F020003, and in part by the National Natural Science Foundation of China under Grant No. 62301484.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 7, 8, 9, 14, 15, 16, 17, 18

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 14, 15

[3] 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*, pages 3981–3991, 2022. 1, 3, 6, 7, 8, 9, 14, 16, 17, 18

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020. 3

[5] Ran Cheng, Christopher Agia, Yuan Ren, Xinhai Li, and Liu Bingbing. S3cnet: A sparse semantic scene completion network for lidar point clouds. In *Conference on Robot Learning*, pages 2148–2161, 2021. 1, 3

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 14

[7] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6185–6194, 2023. 5, 6

[8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 14

[10] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye-view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15273–15282, 2021. 3

[11] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 1

[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. 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*, pages 9223–9232, 2023. 1, 3, 7, 14, 15, 16

[14] Haoyi Jiang, Tianheng Cheng, Naiyu Gao, Haoyang Zhang, Wenyu Liu, and Xinggang Wang. Symphonize 3d semantic scene completion with contextual instance queries. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3, 7, 14, 15, 16

[15] Bu Jin, Yupeng Zheng, Pengfei Li, Weize Li, Yuhang Zheng, Sujie Hu, Xinyu Liu, Jinwei Zhu, Zhijie Yan, Haiyang Sun, et al. Tod3cap: Towards 3d dense captioning in outdoor scenes. *arXiv preprint arXiv:2403.19589*, 2024. 1

[16] Bohan Li, Yasheng Sun, Xin Jin, Wenjun Zeng, Zheng Zhu, Xiaoefeng Wang, Yunpeng Zhang, James Okae, Hang Xiao, and Dalong Du. Stereoscene: Bev-assisted stereo matching empowers 3d semantic scene completion. *arXiv preprint arXiv:2303.13959*, 2023. 2, 3, 7, 8, 9, 15, 16

[17] Hongyang Li, Hao Zhang, Zhaoyang Zeng, Shilong Liu, Feng Li, Tianhe Ren, and Lei Zhang. Dfa3d: 3d deformable attention for 2d-to-3d feature lifting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6684–6693, 2023. 3

[18] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3348–3356, 2020. 3

[19] Pengfei Li, Ruowen Zhao, Yongliang Shi, Hao Zhao, Jirui Yuan, Guyue Zhou, and Ya-Qin Zhang. Lode: Locally conditioned eikonal implicit scene completion from sparse lidar. In *IEEE International Conference on Robotics and Automation*, pages 8269–8276. IEEE, 2023. 3

[20] Yinhao 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1486–1494, 2023. 3

[21] 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*, pages 1477–1485, 2023. 3, 4, 9

[22] Yiming Li, Sihang Li, Xinhao Liu, Moonjun Gong, Kenan Li, Nuo Chen, Zijun Wang, Zhiheng Li, Tao Jiang, Fisher Yu, Yue Wang, Hang Zhao, Zhiding Yu, and Chen Feng. Sscbench: A large-scale 3d semantic scene completion benchmark for autonomous driving. *arXiv preprint arXiv:2306.09001*, 2023. 7, 14, 16

[23] Yiming Li, Zhiding Yu, Christopher B. Choy, Chaowei Xiao, José M. Álvarez, 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*, pages 9087–9098, 2023. 1, 2, 3, 4, 5, 7, 8, 9, 14, 15, 16, 17, 18

[24] 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 *Proceedings of the European Conference on Computer Vision*, pages 1–18, 2022. 1, 3, 7

[25] 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*, pages 10421–10434, 2022. 3, 6

[26] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 3292–3310, 2022. 14

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 14

[28] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022. 3

[29] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 3

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 14

[31] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye-view representation. In *IEEE International Conference on Robotics and Automation*, pages 2774–2781, 2023. 3

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 14

[33] Wenyu Lv, Shangliang Xu, Yian Zhao, Guanzhong Wang, Jinman Wei, Cheng Cui, Yuning Du, Qingqing Dang, and Yi Liu. Detrs beat yolos on real-time object detection. *arXiv preprint arXiv:2304.08069*, 2023. 3

[34] Jianbiao Mei, Yu Yang, Mengmeng Wang, Junyu Zhu, Xiangrui Zhao, Jongwon Ra, Laijian Li, and Yong Liu. Camera-based 3d semantic scene completion with sparse guidance network. *arXiv preprint arXiv:2312.05752*, 2023. 1

[35] Ruihang Miao, Weizhou Liu, Mingrui Chen, Zheng Gong, Weixin Xu, Chen Hu, and Shuchang Zhou. Occdepth: A depth-aware method for 3d semantic scene completion. *arXiv preprint arXiv:2302.13540*, 2023. 9

[36] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proceedings of the European Conference on Computer Vision*, pages 194–210, 2020. 3, 9

[37] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018. 3

[38] Luis Roldão, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *Proceedings of the International Conference on 3D Vision*, pages 111–119, 2020. 1, 3, 4, 7, 8, 16

[39] Faranak Shamsafar, Samuel Woerz, Rafia Rahim, and Andreas Zell. Mobilestereonet: Towards lightweight deep networks for stereo matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2417–2426, 2022. 2, 4, 5, 15

[40] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 190–198, 2017. 1, 3, 7, 8, 16

[41] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019. 14

[42] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. In *Advances in Neural Information Processing Systems*, pages 64318–64330, 2023. 1

[43] Song Wang, Jiawei Yu, Wentong Li, Wenyu Liu, Xiaolu Liu, Junbo Chen, and Jianke Zhu. Not all voxels are equal: Hardness-aware semantic scene completion with self-distillation. *arXiv preprint arXiv:2404.11958*, 2024. 2, 3, 4, 7, 16

[44] Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17850–17859, 2023. 3

[45] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, , and Justin M. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191, 2021. 3

[46] Yu Wang and Chao Tong. H2gformer: Horizontal-to-global voxel transformer for 3d semantic scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5722–5730, 2024. 4, 7, 16

[47] 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*, pages 21729–21740, 2023. 1, 7, 14, 16

[48] Yuan Wu, Zhiqiang Yan, Zhengxue Wang, Xiang Li, Le Hui, and Jian Yang. Deep height decoupling for precise vision-based 3d occupancy prediction. *arXiv preprint arXiv:2409.07972*, 2024. 3

[49] Haihong Xiao, Hongbin Xu, Wenxiong Kang, and Yuqiong Li. Instance-aware monocular 3d semantic scene completion. *IEEE Transactions on Intelligent Transportation Systems*, 2024. 7, 16

[50] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. Mbev: Multi-camera joint 3d detection and segmentation with unified birds-eye-view representation. *arXiv preprint arXiv:2204.05088*, 2022. 6

[51] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3101–3109, 2021. 1, 3

[52] Zhiqiang Yan, Yuankai Lin, Kun Wang, Yupeng Zheng, Yufei Wang, Zhenyu Zhang, Jun Li, and Jian Yang. Tri-perspective view decomposition for geometry-aware depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4874–4884, 2024. 3

[53] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. In *Proceedings of the European Conference on Computer Vision*, pages 214–230, 2022. 3

[54] 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*, pages 17830–17839, 2023. 1, 3

[55] Jiawei Yao and Jusheng Zhang. Depthssc: Depth-spatial alignment and dynamic voxel resolution for monocular 3d semantic scene completion. *arXiv preprint arXiv:2311.17084*, 2023. 7, 16

[56] Zhu Yu, Zehua Sheng, Zili Zhou, Lun Luo, Si-Yuan Cao, Hong Gu, Huaqi Zhang, and Hui-Liang Shen. Aggregating feature point cloud for depth completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8732–8743, 2023. 3

[57] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2302.14325*, 2022. 3

[58] Jiahui Zhang, Hao Zhao, Anbang Yao, Yurong Chen, Li Zhang, and Hongen Liao. Efficient semantic scene completion network with spatial group convolution. In *Proceedings of the European Conference on Computer Vision*, pages 733–749, 2018. 3

[59] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9433–9443, 2023. 5, 7, 9, 15, 16, 17, 18

[60] Yupeng Zheng, Xiang Li, Pengfei Li, Yuhang Zheng, Bu Jin, Chengliang Zhong, Xiaoxiao Long, Hao Zhao, and Qichao Zhang. Monoocc: Digging into monocular semantic occupancy prediction. *arXiv preprint arXiv:2403.08766*, 2024. 2, 3, 4, 7, 16

[61] 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, 3

[62] Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2308.16896*, 2023. 3, 6

A Appendix / Supplemental Material

In the appendix, we mainly provide implementation details and more experiment results.

A.1 Datasets and Metrics

Datasets. We evaluate our CGFormer on two datasets: SemanticKITTI [1] and SSC-Bench-KITTI-360 [22]. These datasets are derived from the KITTI Odometry [6] and KITTI-360 [26] Benchmarks, respectively. The evaluation focuses on a specific spatial volume: $51.2m$ in front of the car, $25.6m$ to the left and right sides, and $6.4m$ above the car. Voxelization of this volume results in a set of 3D voxel grids with a resolution of $256 \times 256 \times 32$, where each voxel measures $0.2m \times 0.2m \times 0.2m$. SemanticKITTI provides RGB images with dimensions of 1226×370 as inputs, encompassing 20 unique semantic classes (19 semantic classes and 1 free class). The dataset includes 10 sequences for training, 1 sequence for validation, and 11 sequences for testing. SSC-Bench-KITTI-360 [22] offers 7 sequences for training, 1 sequence for validation, and 1 sequence for testing. It contains 19 unique semantic classes (18 semantic classes and 1 free class), with input RGB images having a resolution of 1408×376.

Metrics. Following previous methods [3, 23, 13], we report the intersection over union (IoU) and mean IoU (mIoU) metrics for occupied voxel grids and voxel-wise semantic predictions, respectively. The interplay between IoU and mIoU offers a comprehensive perspective on the model's effectiveness in capturing both geometry and semantic aspects of the scene.

A.2 Implementation Details

Network Structures. Consistent with previous researches [13, 3, 47], we utilize a 2D UNet based on a pretrained EfficientNetB7 [41] as the image backbone. The CGVT generates a 3D feature volume with dimensions of $128 \times 128 \times 16$ and 128 channels. The numbers of deformable attention layers for cross-attention and self-attention are 3 and 2 respectively. We use 8 sampling points around each reference point for the cross and self-attention head. The voxel-based branch of the LGE comprises 3 stages with 2 residual blocks [9] each. SwinT [30] is employed as the 2D backbone in the TPV-based branch. Both are followed by feature pyramid networks (FPNs) [27] to aggregate multi-scale features for dynamic fusion. The final prediction has dimensions of $128 \times 128 \times 16$ and is upsampled to $256 \times 256 \times 32$ through trilinear interpolation to align the resolution with the ground truth.

Training Setup. We train CGFormer for 25 epochs on 4 NVIDIA 4090 GPUs, with a batch size of 4. It approximately consumes 19 GB of GPU memory on each GPU during the training phase. We employ the AdamW [32] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and set the maximum learning rate to 3×10^{-4}. The cosine annealing learning rate strategy is adopted for the learning rate decay, where the cosine warmup strategy is applied for the first 5% iterations.

A.3 Results Using Monocular Inputs

In alignment with previous methods [23, 14], we evaluate the performance of our CGFormer using only a monocular RGB image as input. We replace the depth estimation network with AdaBins [2] and present the results on the semantickitti validation set in the table 7. To better demonstrate the advantage of our CGFormer, we also include the results of VoxFormer, Symphonize, and OccFormer. Compared to the stereo-based methods when using only a monocular image (VoxFormer, Symphonize), CGFormer achieves superior performance in terms of both IoU and mIoU. Furthermore, our method also surpasses OccFormer, the state-of-the-art monocular method.

Table 6: The performance of the CGFormer with more lightweight backbone networks.

Backbone Networks	IoU	mIoU	Parameters	Training Memory
EfficientNetB7, Swin Block	**45.99**	**16.87**	122.42	19330
ResNet50, Swin Block	45.99	16.79	80.46	19558
ResNet50, ResBlock	45.86	16.85	**54.8**	**18726**

Table 7: Comparison of the performance using monocular inputs. For stereo-based methods, we replace the MobileStereoNet [39] with Adabins [2].

Method	IoU	mIoU
VoxFormer-S [23]	38.68	10.67
VoxFormer-T [23]	38.08	11.27
Symphonize [14]	38.37	12.20
OccFormer [59]	36.50	13.46
CGFormer (ours)	**41.82**	**14.06**

Table 8: Comparison of training memory and inference time with SOTA methods on the and SemanticKITTI test set. These metrics were measured on the NVIDIA 4090 GPU.

Method	TPVFormer [13]	OccFormer [59]	VoxFormer [23]	Symphonize [14]	StereoScene [16]	CGFormer (ours)
Training Meomry (M)	18564	18080	18725	17757	19000	19330
Inference Time (ms)	207	199	204	216	258	205
IoU	34.25	34.53	42.95	42.19	43.34	**44.41**
mIoU	11.26	12.20	12.20	15.04	15.36	**16.63**

A.4 Reults with More Lightweight Backbone Networks

We reanalyze the components of CGFormer, finding that replacing EfficientNetB7, used as the image backbone, and the Swin blocks, used in the TPV branch backbone, with more lightweight ResNet50 and residual blocks, respectively, can significantly reduce the number of parameters of our network. Besides, we also remove the predefined parameters as we find it doesn't influence the final performance. The results on the semantickitti validation set are presented in the Table 6. Compared to the original architecture, CGFormer maintains stable performance regardless of the backbone networks used for the image encoder and TPV branch encoder, underscoring its effectiveness, robustness, and potential.

A.5 Additional Quantitative Results

For more comprehensive comparison, we list the results with input modality and image backbones in Table 9 and Table 10. Table 11 presents the comparison results of CGFormer with the state-of-the-art methods on the SemanticKITTI validation set. CGFormer outperforms all other methods in terms of both IoU and mIoU. Additionally, it ranks either first or second on most of the classes, demonstrating consistent performance across various semantic categories, as indicated in previous tables.

A.6 Computational Cost

In Table 8, we display the training memory and inference time of CGFormer, along with those of the comparison methods. Additionally, the table includes the corresponding IoU and mIoU metrics for comprehensive comparison. As shown in the table, CGFormer achieves the best performance in terms of both IoU and mIoU, with comparable training memory and inference time.

A.7 Additional Qualitative Results

We offer additional visualization results in Fig.6 and Fig.7. These examples are randomly selected from the SemanticKITTI [1] validation set.

A.8 Failure Cases

We provide two failure cases in Fig. 5.

A.9 Limitations

While CGFormer exhibits strong performance on benchmarks, but the accuracy on most of the categories (*e.g.*, person, bicyclist, other vehicle) is unsatisfactory. Improving the performance on these instances could be beneficial for the downstream application tasks. Furthermore, there is a need to explore designing depth estimation networks under multi-view scenarios to extend the geometry-aware view transformation to these scenes. Despite these limitations, we are confident that CGFormer will contribute to advancing the field of 3D perception.

Table 9: Quatitative results on SemanticKITTI [1] test set. * represents the reproduced results in [13, 59]. The best and the second best results are in **bold** and underlined, respectively. Our CGFormer outperforms temporal stereo-based (Stereo-T) methods or those methods with larger image backbones in terms of IoU and mIoU.

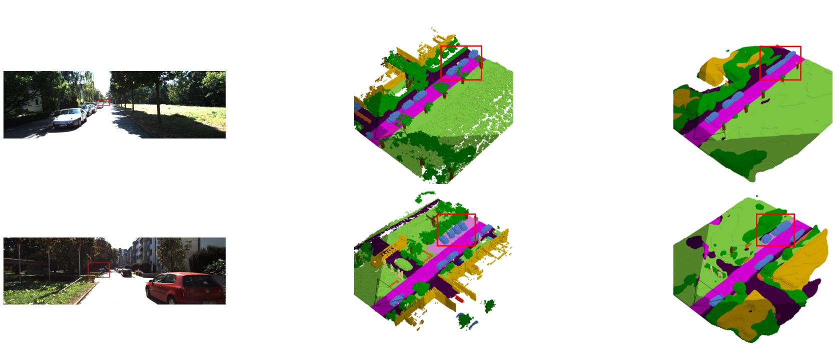
Method	Input	Image Backbone	IoU	mIoU	road (15.30%) ■	sidewalk (11.13%) ■	parking (1.12%) ■	other-grnd. (0.56%) ■	building (14.1%) ■	car (3.92%) ■	truck (0.16%) ■	bicycle (0.03%) ■	motorcycle (0.03%) ■	other-veh. (0.20%) ■	vegetation (39.3%) ■	trunk (0.51%) ■	terrain (9.17%) ■	person (0.07%) ■	bicyclist (0.07%) ■	motorcyclist (0.05%) ■	fence (3.90%) ■	pole (0.29%) ■	traf.-sign (0.08%) ■
MonoScene* [3]	Mono	EfficientNetB7	34.16	11.08	54.70	27.10	24.80	5.70	14.40	18.80	3.30	0.50	0.70	4.40	14.90	2.40	19.50	1.00	1.40	0.40	11.10	3.30	2.10
TPVFormer [13]	Mono	EfficientNetB7	34.25	11.26	55.10	27.20	27.40	6.50	14.80	19.20	3.70	1.00	0.50	2.30	13.90	2.60	20.40	1.10	2.40	0.30	11.00	2.90	1.50
SurroundOcc [47]	Mono	EfficientNetB7	34.72	11.86	56.90	28.30	30.20	6.80	15.20	20.60	1.40	1.60	1.20	4.40	14.90	3.40	19.30	1.40	2.00	0.10	11.30	3.90	2.40
OccFormer [59]	Mono	EfficientNetB7	34.53	12.32	55.90	30.30	31.50	6.50	15.70	21.60	1.20	1.50	1.70	3.20	16.80	3.90	21.30	2.20	1.10	0.20	11.90	3.80	3.70
IAMSSC [49]	Mono	ResNet50	43.74	12.37	54.00	25.50	24.70	6.90	19.20	21.30	3.80	1.10	0.60	3.90	22.70	5.80	19.40	1.50	2.90	0.50	11.90	5.30	4.10
VoxFormer-S [23]	Stereo	ResNet50	42.95	12.20	53.90	25.30	21.10	5.60	19.80	20.80	3.50	1.00	0.70	3.70	22.40	7.50	21.30	1.40	2.60	0.20	11.10	5.10	4.90
VoxFormer-T [23]	Stereo-T	ResNet50	43.21	13.41	54.10	26.90	25.10	7.30	23.50	21.70	3.60	1.90	1.60	4.10	24.40	8.10	24.20	1.60	1.10	0.00	13.10	6.60	5.70
DepthSSC [55]	Stereo	ResNet50	**44.58**	13.11	55.64	27.25	25.72	5.78	20.46	21.94	3.74	1.35	0.98	4.17	23.37	7.64	21.56	1.34	2.79	0.28	12.94	5.87	6.23
Symphonize [14]	Stereo	MaskDINO	42.19	15.04	58.40	29.30	26.90	11.70	24.70	23.60	3.20	3.60	**2.60**	5.60	24.20	10.00	23.10	**3.20**	1.90	**2.00**	16.10	7.70	8.00
HASSC-S [43]	Stereo	ResNet50	43.40	13.34	54.60	27.70	23.80	6.20	21.10	22.80	4.70	1.60	1.00	3.90	23.80	8.50	23.30	1.60	4.00	0.30	13.10	5.80	5.50
HASSC-T [43]	Stereo-T	ResNet50	42.87	14.38	55.30	29.60	25.90	11.30	23.10	23.00	2.90	1.90	1.50	4.90	24.80	9.80	26.50	1.40	3.00	0.00	14.30	7.00	7.10
StereoScene [16]	Stereo	EfficientNetB7	43.34	15.36	61.90	31.20	30.70	10.70	24.20	22.80	2.80	3.40	2.40	6.10	23.80	8.40	27.00	2.90	2.20	0.50	16.50	7.00	7.20
H2GFormer-S [46]	Stereo	ResNet50	44.20	13.72	56.40	28.60	26.50	4.90	22.80	23.40	4.80	0.80	0.90	4.10	24.60	9.10	23.80	1.20	2.50	0.10	13.30	6.40	6.30
H2GFormer-T [46]	Stereo-T	ResNet50	43.52	14.60	57.90	30.40	30.00	6.90	24.00	23.70	5.20	0.60	1.20	5.00	**25.20**	10.70	25.80	1.10	0.10	0.00	14.60	7.50	**9.30**
MonoOcc-S [60]	Stereo	ResNet50	-	13.80	55.20	27.80	25.10	9.70	21.40	23.20	5.20	2.20	1.50	5.40	24.00	8.70	23.00	1.70	2.00	0.20	13.40	5.80	6.40
MonoOcc-L [60]	Stereo	InternImage-XL	-	15.63	59.10	30.90	27.10	9.80	22.90	23.90	**7.20**	**4.50**	2.40	**7.70**	25.00	9.80	26.10	2.80	**4.70**	0.60	16.90	7.30	8.40
CGFormer (ours)	Stereo	EfficientNetB7	44.41	**16.63**	**64.30**	**34.20**	**34.10**	**12.10**	**25.80**	**26.10**	4.30	3.70	1.30	2.70	24.50	**11.20**	**29.30**	1.70	3.60	0.40	**18.70**	**8.70**	**9.30**

Table 10: Quantitative results on SSCBench-KITTI360 test set. The results for counterparts are provided in [22]. The best and the second best results for all camera-based methods are in **bold** and underlined, respectively. The best results from the LiDAR-based methods are in red.

Method	Input	Image Backbone	IoU	mIoU	car (2.85%) ■	bicycle (0.01%) ■	motorcycle (0.01%) ■	truck (0.16%) ■	other-veh. (5.75%) ■	person (0.02%) ■	road (14.98%) ■	parking (2.31%) ■	sidewalk (6.43%) ■	other-grnd. (2.05%) ■	building (15.67%) ■	fence (0.96%) ■	vegetation (41.99%) ■	terrain (7.10%) ■	pole (0.22%) ■	traf.-sign (0.06%) ■	other-struct. (4.33%) ■	other-obj. (0.28%) ■
LiDAR-based methods																						
SSCNet [40]	LiDAR	-	53.58	16.95	31.95	0.00	0.17	10.29	0.00	0.07	65.70	17.33	41.24	3.22	44.41	6.77	43.72	28.87	0.78	0.75	8.69	0.67
LMSCNet [38]	LiDAR	-	47.35	13.65	20.91	0.00	0.00	0.26	0.58	0.00	62.95	13.51	33.51	0.20	43.67	0.33	40.01	26.80	0.00	0.00	3.63	0.00
Camera-based methods																						
MonoScene [3]	Mono	EfficientNetB7	37.87	12.31	19.34	0.43	0.58	8.02	2.03	0.86	48.35	11.38	28.13	3.32	32.89	3.53	26.15	16.75	6.92	5.67	4.20	3.09
TPVFormer [13]	Mono	EfficientNetB7	40.22	13.64	21.56	1.09	1.37	8.06	2.57	2.38	52.99	11.99	31.07	3.78	34.83	4.80	30.08	17.52	7.46	5.86	5.48	2.70
OccFormer [59]	Mono	EfficientNetB7	40.27	13.81	22.58	0.66	0.26	9.89	3.82	2.77	54.30	13.44	31.53	3.55	36.42	4.80	31.00	19.51	7.77	8.51	6.95	4.60
VoxFormer [23]	Stereo	ResNet50	38.76	11.91	17.84	1.16	0.89	4.56	2.06	1.63	47.01	9.67	27.21	2.89	31.18	4.97	28.99	14.69	6.51	6.92	3.79	2.43
IAMSSC [49]	Mono	ResNet50	41.80	12.97	18.53	2.45	1.76	5.12	3.92	3.09	47.55	10.56	28.35	4.12	31.53	6.28	29.17	15.24	8.29	7.01	6.35	4.19
DepthSSC [55]	Stereo	ResNet50	40.85	14.28	21.90	2.36	4.30	11.51	4.56	2.92	50.88	12.89	30.27	2.49	37.33	5.22	29.61	21.59	5.97	7.71	5.24	3.51
Symphonies [14]	Stereo	MaskDINO	44.12	18.58	**30.02**	1.85	**5.90**	**25.07**	**12.06**	**8.20**	54.94	13.83	32.76	**6.93**	35.11	**8.58**	38.33	11.52	14.01	9.57	**14.44**	**11.28**
CGFormer (ours)	Stereo	EfficientNetB7	**48.07**	**20.05**	29.85	**3.42**	3.96	17.59	6.79	6.63	**63.85**	**17.15**	**40.72**	5.53	**42.73**	8.22	**38.80**	**24.94**	**16.24**	**17.45**	10.18	6.77

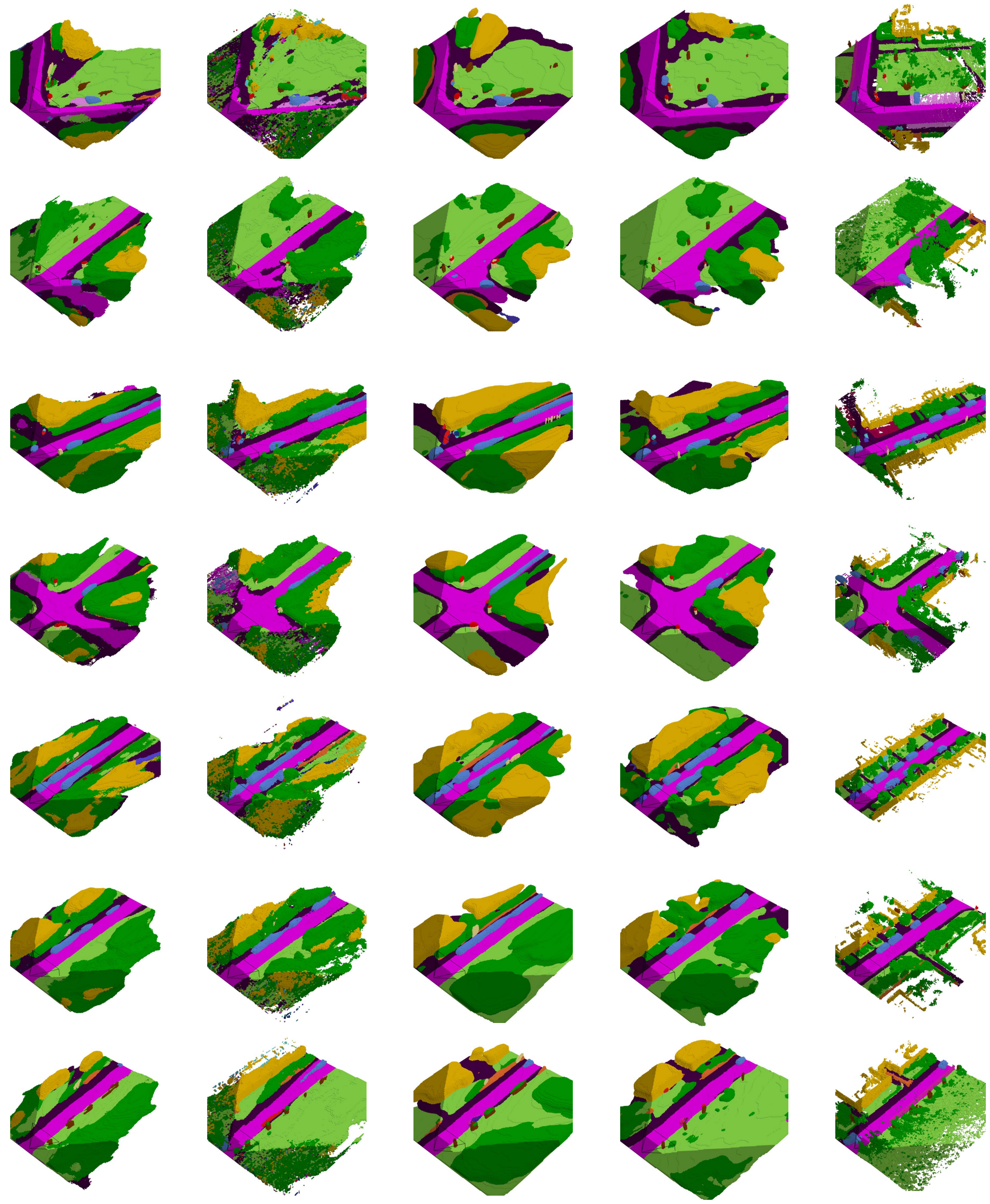
Table 11: Quatitative results on SemanticKITTI [1] validation set. * represents the reproduced results in [13, 59, 55]. The best and the second best results are in **bold** and underlined, respectively.

Method	Input	Image Backbone	IoU	mIoU	road (15.30%) ■	sidewalk (11.13%) ■	parking (1.12%) ■	other-grnd. (0.56%) ■	building (14.1%) ■	car (3.92%) ■	truck (0.16%) ■	bicycle (0.03%) ■	motorcycle (0.03%) ■	other-veh. (0.20%) ■	vegetation (39.3%) ■	trunk (0.51%) ■	terrain (9.17%) ■	person (0.07%) ■	bicyclist (0.07%) ■	motorcyclist (0.05%) ■	fence (3.90%) ■	pole (0.29%) ■	traf.-sign (0.08%) ■
MonoScene* [3]	Mono	EfficientNetB7	36.86	11.08	56.52	26.72	14.27	0.46	14.09	23.26	6.98	0.61	0.45	1.48	17.89	2.81	29.64	1.86	1.20	0.00	5.84	4.14	2.25
TPVFormer [13]	Mono	EfficientNetB7	35.61	11.36	56.50	25.87	20.60	0.85	13.88	23.81	8.08	0.36	0.05	4.35	16.92	2.26	30.38	0.51	0.89	0.00	5.94	3.14	1.52
OccFormer [59]	Mono	EfficientNetB7	36.50	13.46	**58.85**	26.88	19.61	0.31	14.40	25.09	**25.53**	0.81	1.19	8.52	19.63	3.93	32.62	2.78	2.82	0.00	5.61	4.26	2.86
IAMSSC [49]	Mono	ResNet50	44.29	12.45	54.55	25.85	16.02	0.70	17.38	26.26	8.74	0.60	0.15	5.06	24.63	4.95	30.13	1.32	3.46	0.01	6.86	6.35	3.56
VoxFormer-S [23]	Stereo	ResNet50	44.02	12.35	54.76	26.35	15.50	0.70	17.65	25.79	5.63	0.59	0.51	3.77	24.39	5.08	29.96	1.78	3.32	0.00	7.64	7.11	4.18
VoxFormer-T [23]	Stereo-T	ResNet50	44.15	13.35	53.57	26.52	19.69	0.42	19.54	26.54	7.26	1.28	0.56	7.81	26.10	6.10	33.06	1.93	1.97	0.00	7.31	9.15	4.94
DepthSSC [55]	Stereo	ResNet50	45.84	13.28	55.38	27.04	18.76	0.92	19.23	25.94	6.02	0.35	1.16	7.50	26.37	4.52	30.19	2.58	**6.32**	0.00	8.46	7.42	4.09
Symphonize [14]	Stereo	MaskDINO	41.92	14.89	56.37	27.58	15.28	0.95	21.64	28.68	20.44	2.54	**2.82**	**13.89**	25.72	6.60	30.87	**3.52**	2.24	0.00	8.40	9.57	5.76
HASSC-S [43]	Stereo	ResNet50	44.82	13.48	57.05	28.25	15.90	1.04	19.05	27.23	9.91	0.92	0.86	5.61	25.48	6.15	32.94	2.80	4.71	0.00	6.58	7.68	4.05
HASSC-T [43]	Stereo-T	ResNet50	44.58	14.74	55.30	29.60	**25.90**	**11.30**	23.10	23.00	2.90	1.90	1.50	4.90	24.80	**9.80**	26.50	1.40	3.00	0.00	**14.30**	7.00	7.10
H2GFormer-S [46]	Stereo	ResNet50	44.57	13.73	56.08	29.12	17.83	0.45	19.74	28.21	10.00	0.50	0.47	7.39	26.25	6.80	34.42	1.54	2.88	0.00	7.24	7.88	4.68
H2GFormer-T [46]	Stereo-T	ResNet50	44.69	14.29	57.00	29.37	21.74	0.34	20.51	28.21	6.80	0.95	0.91	9.32	**27.44**	7.80	36.26	1.15	0.10	0.00	7.98	9.88	5.81
CGFormer (ours)	Stereo	EfficientNetB7	**45.99**	**16.87**	**65.51**	**32.31**	20.82	0.16	**23.52**	**34.32**	19.44	**4.61**	2.71	7.67	26.93	8.83	**39.54**	2.38	4.08	0.00	9.20	**10.67**	**7.84**



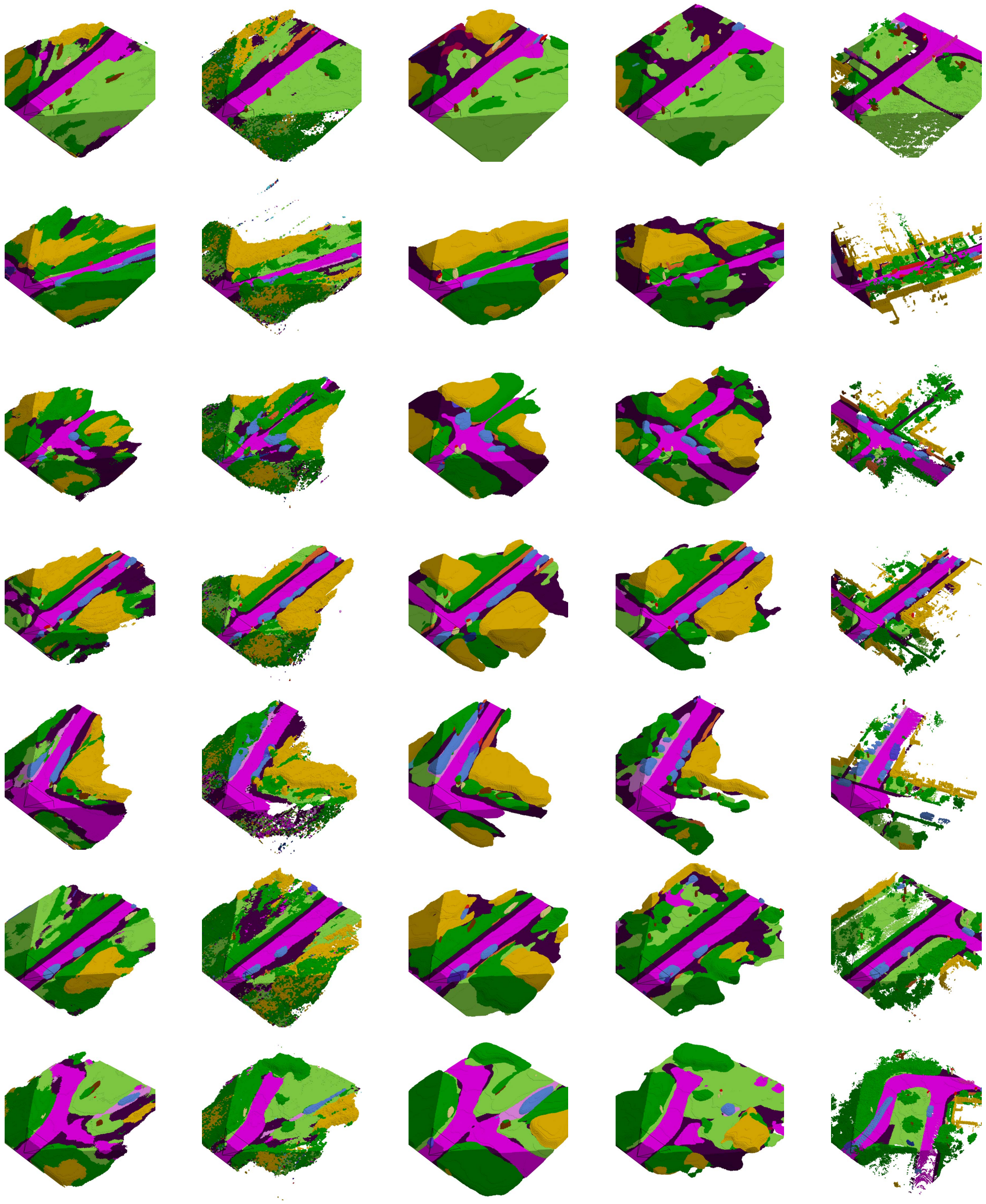
(a) RGB (b) Ground Truth (c) CGFormer (ours)

Figure 5: Failure cases.



(a) MonoScene [3] (b) VoxFormer [23] (c) OccFormer [59] (d) CGFormer (ours) (e) Ground Truth

Figure 6: More qualitative comparison results on the SemanticKITTI [1] validation set.



(a) MonoScene [3] (b) VoxFormer [23] (c) OccFormer [59] (d) CGFormer (ours) (e) Ground Truth

Figure 7: More qualitative comparison results on the SemanticKITTI [1] validation set.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We describe the motivation of our work at the beginning of the abstract. Then we describe the methodology we employed to tackle the presented issues. Our contributions are summarized in the last of introduction. Refer to the introduction to validate it.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a subsection of limitation in the supplement material.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We provide formulas for better understanding our job, but we don't propose new theorems.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the detailed implementation in the supplement material. Code is also provided to validate it.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide code as an additional zip file. However, as the checkpoint of the model is too large to upload, we provide a anonymous link for downloading the pretrained checkpoints and scripts for retraining the model. Code will be publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the details of datasets and metrics, which includes the data splits and evaluation metrics. Hyper-parameters and optimizer can be found in the implementation details in the supplement material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following previous methods, we obtain the results by submitting results to the leaderboard. And all of our experiments are conducted using the same seed 7240.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our model is trained on 4 NVIDIA 4090 GPUs, with a batch size of 4. It approximately consumes 19GB of GPU memory on each GPU during the training phase. Refer to the implementation details for detailed information.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, the research conducted in the paper conforms with the NeurIPS Code of Ethics. The provided code and link are both anonymous.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We mentioned that we are confident that CGFormer will contribute to advancing the field of 3D perception and autonomous driving at the end of the conclusion, while we also pointed that the accuracy should be further improved for actual application.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No, our paper did not utilize any high-risk data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We will thank all the assets we refer to when the code is publicly available. We respect these methods and cite them in our paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.