World Electric Vehicle Journal

*Article*

# Bird's-Eye View Semantic Segmentation for Autonomous Driving through the Large Kernel Attention Encoder and Bilinear-Attention Transform Module

**Ke Li [1], Xuncheng Wu [1,*], Weiwei Zhang [2] and Wangpengfei Yu [2]**

[1] School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; jxgz-like@foxmail.com
[2] Shanghai Smart Vehicle Cooperating Innovation Center Co., Ltd., Shanghai 201805, China
* Correspondence: wuxunchengsues@163.com

**Abstract:** Building an autonomous driving system requires a detailed and unified semantic representation from multiple cameras. The bird's eye view (BEV) has demonstrated remarkable potential as a comprehensive and unified perspective. However, most current research focuses on innovating the view transform module, ignoring whether the crucial image encoder can construct long-range feature relationships. Hence, we redesign an image encoder with a large kernel attention mechanism to encode image features. Considering the performance gains obtained by the complex view transform module are insignificant, we propose a simple and effective Bilinear-Attention Transform module to lift the dimension completely. Finally, we redesign a BEV encoder with a CNN block of a larger kernel size to reduce the distortion of BEV features away from the ego vehicle. The results on the nuScenes dataset confirm that our model outperforms other models with equivalent training settings on the segmentation task and approaches state-of-the-art performance.

**Keywords:** camera; bird's eye view; autonomous driving; view transformation; semantic segmentation

**Citation:** Li, K.; Wu, X.; Zhang, W.; Yu, W. Bird's-Eye View Semantic Segmentation for Autonomous Driving through the Large Kernel Attention Encoder and Bilinear-Attention Transform Module. *World Electr. Veh. J.* **2023**, *14*, 239. https://doi.org/10.3390/wevj14090239

Academic Editors: Biao Yu, Linglong Lin and Jiajia Chen

Received: 23 July 2023
Revised: 14 August 2023
Accepted: 21 August 2023
Published: 1 September 2023

## 1. Introduction

The development of autonomous driving has become a highly dynamic area of research. To ensure safety, autonomous driving needs access to a robust, detailed, and rich representation of its surroundings, especially in urban driving scenarios. As one of the crucial technologies of autonomous driving, environment perception [1,2] is mainly achieved through the combination of cameras, radar, LIDAR, and other sensors to capture information about the environment around the vehicle, which is also a prerequisite and basis for the realization of autonomous driving. In recent years, offline High-Definition maps combined with environmental awareness have become a viable solution to achieve high-level autonomous driving functions as soon as possible. However, the High-Definition map is limited by the update frequency and update cost and is not the perfect choice for the solution. In this context, Bird's-Eye View (BEV) perception is gradually developed as an alternative solution that provides a more comprehensive and detailed representation of the surroundings in a top-down view of the scene and facilitates downstream tasks such as navigation and control of autonomous vehicles.

Due to the advantages of BEV in downstream tasks, the field of BEV sensing has grown rapidly in the past three years and has produced many excellent research studies. The history of the development of BEV perception can be traced back to [3], which proposes the Inverse Perspective Transformation (IMP) to accomplish the transformation of image views to BEV and is a pioneering work in view transformation. Existing BEV perception methods can be divided into four categories according to the view transform module: IMP-based methods, depth-based or voxel-based methods, MLP-based methods, and Transformer-based methods. However, the existing methods have some limitations and areas for improvement. The IMP-based methods, such as [4–6], are unable to generate high-quality BEV representations due to the large differences and severe deformations between the two views. In addition, these methods rely heavily on the horizon assumption, and the models perform very poorly when the environment does not conform to that assumption or when no prior information is available. The MLP-based methods, such as [7,8], ignore the geometric prior for calibrating the camera and usually transform the multi-view images separately, which cannot fully exploit the information embedded in the overlapping parts of the images. Transformer-based methods, such as [9,10], have undergone rapid development in the last two years and have shown excellent model results. However, these methods have efficiency problems in the training and inference processes, which limit their practical application. Moreover, these methods still rely on deep pretraining, indicating that depth information is still crucial for view transformation in such methods. In contrast, depth-based methods, such as [11–14], have higher computational efficiency and flexibility in multi-view image processing. However, there is still a gap between these methods and the state-of-the-art LiDAR-based models. To bridge this gap, we need to explore performance improvement paths further while maintaining computational efficiency.

Summarizing previous work, depth- or voxel-based BEV perception exhibits a high degree of modularity and reusability with a paradigm with three essential components: an image encoder, a view transform module, and a BEV encoder. These different modules form a comprehensive BEV perception pipeline. The image encoder is the foundation and provides the necessary feature extraction capabilities. The view transform module converts input data from multiple camera views into a uniform BEV representation for consistent and coherent processing. The BEV encoder module encodes the 3D voxel data, capturing the intrinsic spatial and semantic features for subsequent analysis. This modular design allows the decoupling of specific functions, increasing flexibility and facilitating the reuse of individual components in different BEV-perception systems. Most current research focuses on improving or innovating view transform modules, ignoring the image encoder and the BEV encoder. However, the state-of-the-art view transform module yields only a four-point performance improvement [15]. While it is true that the view transformation module is an integral part of the model both intuitively and practically, using simple bilinear sampling to perform the view transformation work is equally effective, though not at the most advanced level. The view transform module has much less impact on the model's performance in the current architecture than the selection of the appropriate input resolution and batch size [15]. Meanwhile, since the region size of BEV features is artificially set, the data features out of range in the original image will be discarded. Therefore, in the initial stage of the model, using a backbone network with stronger modeling capability to learn long-range relationships can further exploit the information in the images and improve the model's performance. In addition, the 2D to 3D transform module causes much information loss, especially distortion of features at long distances.

Following the above, we investigate three components: the image encoder, the view transform module, and the BEV encoder. The traditional CNN image encoders, such as ResNet [16] and ConvNeXt [17], cannot model long-range feature relationships. In contrast, the Transformer-based image encoder ignores more feature relationships between channels, while model training is more difficult and data requirements are greater. In addition, the Large Kernel Attention [18] module combines the advantages of convolution and self-attentive mechanisms, including local structural information, remote-dependent modeling capability, and adaptability. Therefore, we utilize the Large Kernel Attention in combination with the ConvNeXt module, where ConvNeXt can improve the modeling ability of local structural information in the model, and the Large Kernel Attention can further complement the modeling ability of the model for remote features while retaining the modeling ability of the model for channels. For the view transform module, a single bilinear sampling is used to complete the view transform task, which does not pay enough attention to the local information of image features. We combine bilinear sampling and attention mechanisms to design a view transform module to ensure that it can complete the view transformation simply and efficiently while having better transform performance. Meanwhile, some depth-based studies [15,19,20] all follow the same BEV features map settings yet simply use ResNet-18 as the BEV features encoder, which also leads to the fact that the BEV features map itself suffers from distortion of long-range features after the view transform module, which is not well solved, and therefore the model is less effective in the regions farther away from the center of the ego vehicle. To alleviate this contradiction, we redesigned a module for encoding BEV features using a combination of large kernel-size convolutional blocks [21], which can efficiently model the remote relations of BEV features. In this study, we select semantic segmentation as the evaluation task of our proposed model. We also conducted experiments on the nuScenes dataset to evaluate our proposed model's performance. Our proposed model shows good performance with a mIoU of 45.6 in the experiments. The results also illustrate the effectiveness of each component.

In conclusion, our contributions are summarized as follows:

1. We redesign an image encoder combined with the Large Kernel Attention to address the conventional encoder's lack of remote feature modeling capability.
2. To overcome the great difference between image features and BEV features, a view transform module is designed by combining bilinear sampling and attention mechanisms to ensure that BEV features pay more attention to image features that are closely related.
3. To address the problem of distortion of BEV features at long distances, a BEV encoder with a large kernel size for BEV features is redesigned to obtain a larger receptive area.

The paper is organized as follows: Section 2 reviews the related work in BEV perception methods. Section 3 introduces our proposed model's overall architecture and each component's composition. Next, Section 4 describes the experimental setup, experiment results, comparative results, and detailed component analysis. The conclusion of this study is summarized in Section 5.

## 2. Related Work

In this section, we classify BEV perception into two categories according to the view transformation methods: geometry-based and network-based methods, and we describe the work related to each of these two categories.

### *2.1. Geometry-Based Method*

The IMP-based method utilizes the homographic matrix derived from the internal and external parameters of the camera. Cam2BEV [4] first employs IPM to transform multiple image features and finally obtains the BEV semantic map. To alleviate the distortion problem of the IPM-based method, TrafCam3D [5] proposes a dual-view network structure. For the pedestrian prediction problem, SHOT [6] projects each part of the pedestrian at different ground levels, respectively. The above studies demonstrate that IMP is effective enough under the condition that the flat-ground assumption is satisfied.

However, it is obvious that real-world driving scenarios cannot always satisfy the flat-ground assumption. Therefore, researchers began working on predicting the exact depth needed to accomplish the task of perspective view to BEV transformation. First, the Depth-based methods lift 2D features into 3D space by adding depth. Specifically, [22,23] predict each pixel's depth and directly utilize the existing LiDAR-based task head after transforming the 2D features into a pseudo-point cloud type. CaDDN [12] proposes a similar approach, but instead of directly generating a pseudo-point cloud, the pixels with predicted depth distribution are projected to the BEV, while the process uses depth supervision from the LiDAR. In addition, LSS [11] proposes to predict an explicit depth distribution for each 2D feature and then project the 2D features into BEV features. Based on the LSS, BEVDet [19] proposes a multi-camera model for 3D object detection tasks. BEVDet4D [24] exploits the previous camera frames to enhance the model's performance. BEVDepth [20] demonstrates that the performance of BEV-perception models can benefit from depth supervision and proposes a faster pooling operation.

### *2.2. Network-Based Method*

The network-based methods start with using MLP for the view transformation task, transforming the perspective view to BEV. VED [25] first proposes an end-to-end monocular real-time prediction model with a MLP layer to transform the perspective view to BEV. VPN [7] further applies the MLP-based view conversion module to scenarios with multiple camera inputs. Specifically, VPN first transforms the encoded image features from multiple cameras into BEV features using MLP and then fuses all BEV features. FISHING [26] then introduces LIDAR and radar features based on VPN to complete the post-fusion and achieve multimodal perception. To address the problem of spatial information loss caused by MLP, PON [8] uses feature pyramids to obtain multi-scale image features and then uses MLP for view transformation. HFT [27] makes a further comparison between the advantages and disadvantages of using the camera parameter-based MLP method.

The Transformer-based methods utilize the currently popular Transformer to design the view transform module without the camera parameters. Unlike the MLP-based method that starts with 2D features, this method, in turn, uses an attention mechanism to capture the corresponding 2D features. DETR [28] and STSU [29] accomplish the 2D detection task by capturing the corresponding features with the pre-defined query. DETR3D [30], on the other hand, extends DETR to 3D object detection by geometric feature sampling. For autonomous driving, PETR [10] and PETRv2 [31] further employ camera parameters to generate position encoding and utilize temporal cues, respectively.

## 3. Method

This section presents the detailed design of our proposed model for BEV semantic segmentation. In Section 3.1, we first describe the overall architectural details of our proposed model. In Section 3.2, we introduce the image encoder for extracting 2D features from multi-camera images and illustrate why and how to redesign the image encoder. Next, we explain in detail how our designed view transform module generates 3D voxel features from 2D features in Section 3.3. In Section 3.4, we give detailed illustrations of the BEV encoder of our proposed model.

### *3.1. Overall Architecture*

In this study, our model pipeline takes $N$ RGB images $\mathbf{F} = \{\mathbf{F}_i \in \mathbb{R}^{3\times H\times W}, i = 1, 2, \ldots, N\}$ from multiple cameras and the corresponding internal and external parameters as input. The output is a BEV segmentation map obtained by a specific task head. Specifically, our proposed model consists of three core components: the image encoder, the view transform module named the Bilinear-Attention module, and the BEV encoder. In particular, the Bilinear-Attention module consists of two submodules: the Bilinear Sampling module and the Deformable Attention module, as shown in Figure 1. The multi-camera images are first fed to the image encoder, which outputs 2D features $\mathbf{F}^{2d} = \left\{\mathbf{F}_i^{2d} \in \mathbb{R}^{C_{2d}\times H_{2d}\times W_{2d}}, i = 1, 2, \ldots, N\right\}$. Next, the 3D voxel $\mathbf{F}^{vox} \in \mathbb{R}^{C_{vox}\times Z_{vox}\times X_{vox}}$ and corresponding 3D coordinates generated in advance are projected onto the 2D features and constructed by bilinear sampling. The 3D voxel is then further refined by the Deformable Attention module. Finally, the 3D voxel is encoded into BEV features $\mathbf{F}^{bev} \in \mathbb{R}^{C_{bev}\times H_{bev}\times W_{bev}}$ by the BEV encoder. The details of each component in the pipeline are described in the following sections.

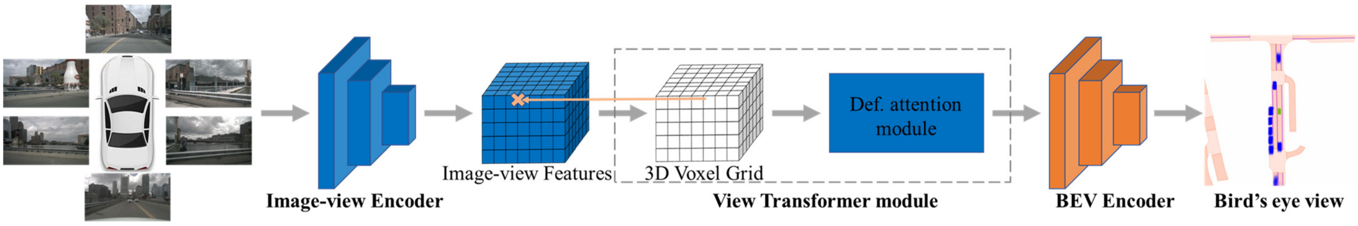


**Figure 1.** The pipeline of our proposed model.

### *3.2. Image Encoder*

A high-performance image feature encoder is crucial for computer vision tasks. Much research in BEV perception uses ResNet or EfficientNet as the backbone network of the image encoder because both ResNet and EfficientNet are proven and mature networks with excellent performance. However, when training the model, the backbone network is the last stage to update the parameters and is less affected because the model is updated by backpropagation. Theoretically, the performance of the model is better if a backbone network with better pre-training performance and more robustness is selected. In the past, the powerful modeling capability of Visual Transformer has greatly impacted the image field. ConvNeXt [17], on the other hand, proposes a new pure convolution that provides stronger performance by emulating the Visual Transformer model. However, ConvNeXt is limited by the convolutional kernel size, cannot model remote dependencies, and cannot provide a good balance between local and global modeling capabilities. Although the excellent Transformer-based backbone network has excellent remote feature modeling capability, its huge data demand, higher training difficulty, and more computational resources required than convolutional networks make it not applicable. The Large Kernel Attention module [18] overcomes the abovementioned drawbacks and nicely combines the advantages of self-attention and large kernel convolution. The Large Kernel Attention module consists of three components: a spatial local convolution (depth-wise convolution), a spatial long-range convolution (depth-wise dilation convolution), and a channel convolution (1×1 convolution), as shown in Figure 2. Specifically, a $K \times K$ convolution is decomposed into a $\lceil \frac{K}{d} \rceil \times \lceil \frac{K}{d} \rceil$ depth-wise dilation convolution with dilation $d$, a $(2d-1) \times (2d-1)$ depth-wise convolution, and a $1 \times 1$ convolution.

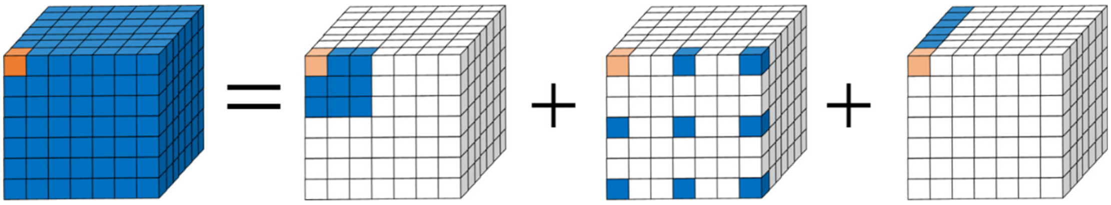


**Figure 2.** Decomposition diagram of large-kernel convolution from [18].

Through the above decomposition, the Large Kernel Attention module captures long-range relationships with slight computational cost and parameters, estimates the importance of a point, and generates an attention map. The Large Kernel Attention module can be written as

$$\textbf{Attention} = \text{Conv}_{1\times1}(\text{DW-D-Conv}(\text{DW-Conv}(\text{F}))) \tag{1}$$

$$\textbf{Output} = \textbf{Attention} \otimes \textbf{F} + \textbf{F} \tag{2}$$

where $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$ is the input feature, $\textbf{Attention} \in \mathbb{R}^{C\times H\times W}$ denotes attention map, which indicates the importance of each feature, and $\otimes$ means element-wise product. We redesign an image encoder combining the Large Kernel Attention and ConvNeXt blocks to address the above situation. The network architecture of our image encoder is shown in Figure 3.

Given $N$ images of the size $(H, W)$, we construct the image encoder shown in Figure 3a to downsample the input image by a factor of 8 to obtain the output features with a resolution of $(H/8, W/8)$. Specifically, the network architecture of the image encoder consists of five parts: the stem part, three downsampling blocks, four stages of ConvNeXt blocks, two blocks with the Large Kernel Attention, and a bilinear upsampling block. We first employ a convolutional layer with a kernel size of 4 and a stride of 4 and choose layer normalization to do the normalization operation in the stem part, as shown in Figure 3b. In this way, we can obtain the output feature map of the stem part as $(H/4, W/4)$. Next, we adopt three convolution stages of ConvNeXt [17], which contain convolution blocks in the order of (3, 3, 9). The specific composition of the convolution block is shown in Figure 3d.

It contains a depthwise convolution layer with a kernel size of 7 responsible for mixing information in the spatial dimension, layer normalization, the GELU activation function, and two pointwise convolution layers for mixing information in the channel dimension, respectively. To further obtain more global features, we employ the Large Kernel Attention mechanism as described before after the first and second convolutional stages, whose specific structure is shown in Figure 3e. In addition, a separate downsampling layer is added in the middle of each convolutional stage, consisting of a convolutional layer with a kernel size of 2 and a stride of 2, and a layer normalization to achieve spatial downsampling. Finally, to obtain richer features, we concatenate the output of the third convolution block with an upsampling multiplier of two through an upsampling block with the output of the fourth convolution block to obtain the final 2D features $\mathbf{F}^{2d}$. This up-convolution block consists of one up-sampling layer and two convolution layers with a kernel size of 3, as shown in Figure 3c.

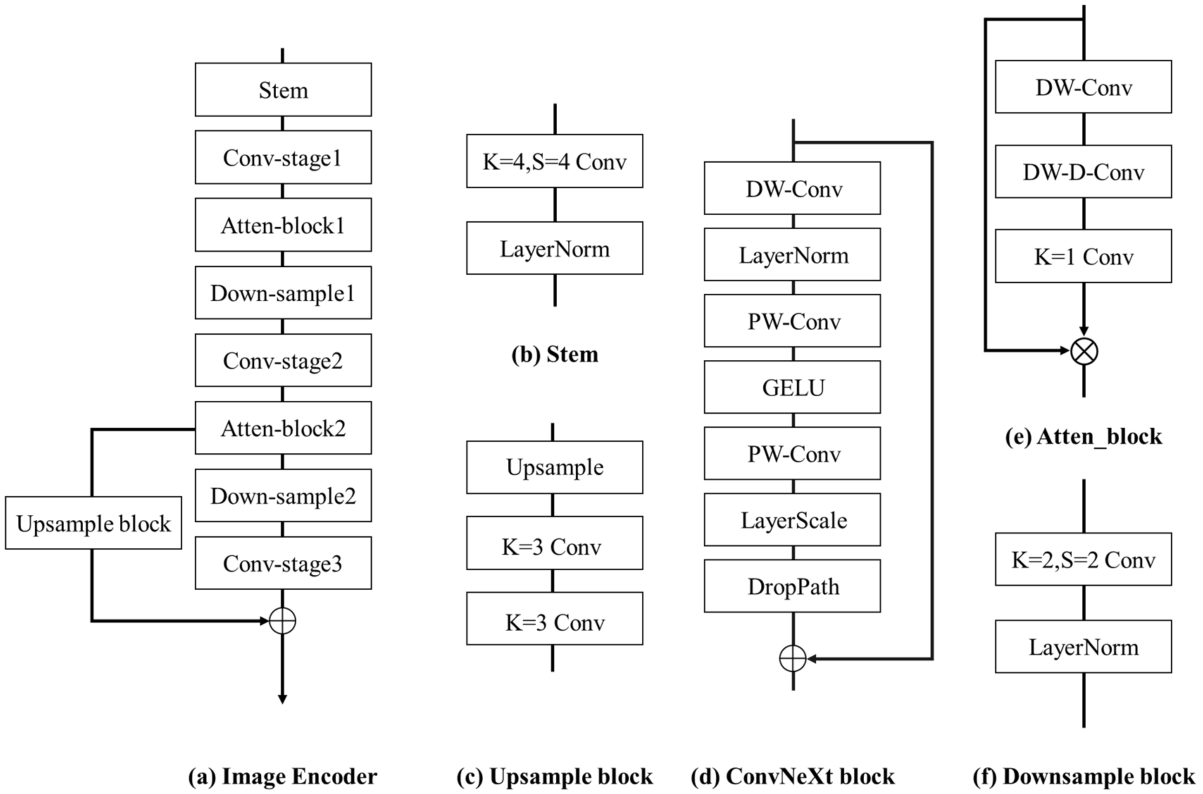


**Figure 3.** The structure of our redesigned image encoder. (**a**) The overall architecture of the image-view encoder. (**b**) The structure of the Stem part. (**c**) The structure of the Upsample block. (**d**) The structure of the ConvNeXt block. (**e**) The structure of the Large Kernel Attention block. (**f**) The structure of the downsample block.

### *3.3. View Transform Module*

The depth-based approach achieves dimensionality lifting of 2D features by predicting a corresponding set of depth values for each 2D feature, then puts all features into a pre-generated view frustum, and finally forms a BEV feature map by pooling calculations. Although the method achieves satisfactory results, the predicted depth values depend on the ground plane assumption, and the accuracy of depth prediction seriously affects the model's overall performance. In addition, the Simple-BEV [15] also demonstrates the substitutability of the Lift-Splat method [11]. For the lifting operation of 2D features, [15] employs a pre-generated set of 3D voxels to obtain sub-pixel features by projecting each voxel in the 2D feature map using bilinear sampling for each voxel. It has been experimentally demonstrated that this method is more efficient due to the absence of hyperparameters while maintaining its effectiveness. However, there are better choices than this view transform method because simple sampling implies a lack of global information modeling capability, which impacts model performance. To address the above situation, we designed a view transform module named the Bilinear-Attention module, as shown in Figure 4. The structure of the proposed Bilinear-Attention module is composed of a Bilinear Sample module and a Compressed Attention module.

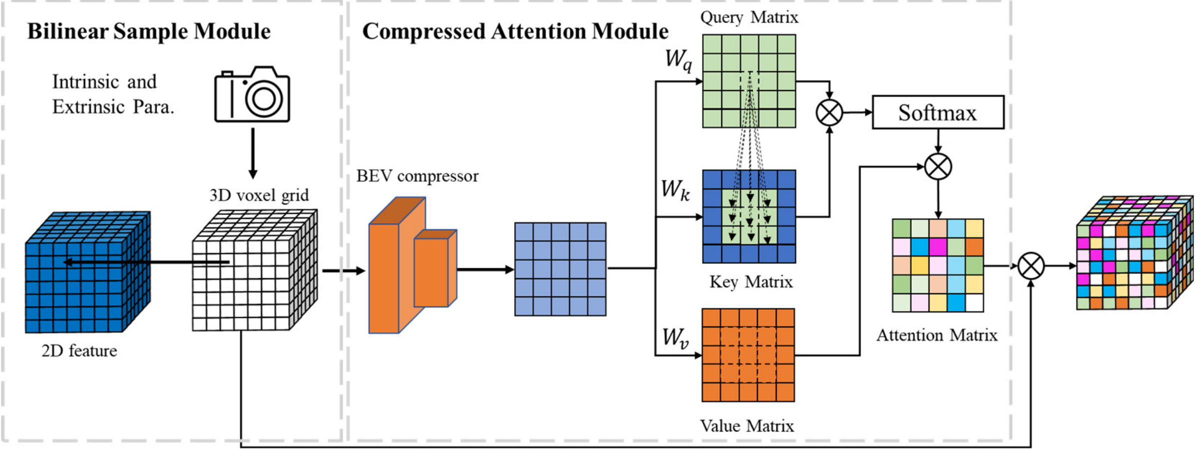


**Figure 4.** The overall architecture of the Bilinear-Attention module. The view transform module has two components: the Bilinear Sample module to complete the 2D feature lifting dimension operation and the Compressed Attention module to refine the 3D voxel features.

We assume that the input feature map size for the view transform module is (1, 2). We first generate a 3D feature for each voxel grid by bilinear sampling using a pre-defined 3D voxel grid with dimensions $(Z, Y, X)$ and its corresponding 3D coordinate information. Specifically, according to the set hyperparameters $(Z, Y, X)$, a 3D voxel grid is generated along with the 3D coordinate information corresponding to each grid, which is then converted to the corresponding 2D coordinates using the coordinate system conversion formula based on 2D and 3D space, as follows:

$$\lambda \mathbf{p} = \left[ \mathbf{K} \mid \mathbf{0}_3 \right] \begin{bmatrix} \mathbf{R} & 0 \\ \mathbf{0}_3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{0}_3^T & -\mathbf{C} \\ 0 & 1 \end{bmatrix} \mathbf{P} \tag{3}$$

where $\mathbf{p} = \begin{pmatrix} x & y & 1 \end{pmatrix}^T$ denotes the 2D pixel position. $\mathbf{P} = \begin{pmatrix} X & Y & Z & 1 \end{pmatrix}^T$ being a 3D point defined with homogeneous coordinates. The projection matrix that incorporates the intrinsic parameters is denoted as $\mathbf{K}$ throughout this thesis. Mathematically, the position and orientation of the camera are defined by a 3 × 1 vector $\mathbf{C}$ and a 3 × 3 rotation matrix $\mathbf{R}$. Next, bilinear sampling is performed according to the corresponding 2D coordinates to generate the corresponding voxels for each 3D grid to obtain the sampled 3D voxel grid $\mathbf{F}'$, whose dimensional size is $(N, Y, Z, X)$. This method, however, has a limited receptive field for mapping the generated 3D features onto the 2D feature maps. To address this problem, we then compress the dimension using a convolutional layer with a kernel size of 3 to obtain the output features $X$ used to generate the Query $Q$ and Key $K$ on the basis of a 3D voxel grid filled with features of size $(N, 1, Z, X)$, and the process can be expressed as follows:

$$Q = W_q X \tag{4}$$

$$K = W_k X \tag{5}$$

$$V = W_v X \tag{6}$$

where the $W_q$, $W_k$, and $W_v$ are the learnable parameters. Meanwhile, the dimensions of $Q$ and $K$ are set to be the same $(N, 1, Z, X)$ and the dimensions of value $V$ are the same as

3D features $\mathbf{F}'$. On top of the obtained Query, Key, and Value, we adopt a self-attentive mechanism to refine the 3D features further and increase the global interaction of the features. We first compute the attention map using the dot product of Query and Key. In the next step, we employ the computed attention map and $V$ to generate the final voxel features. The process can be expressed as follows:

$$\mathbf{Attention} = \mathrm{Softmax}(QK^T) \tag{7}$$

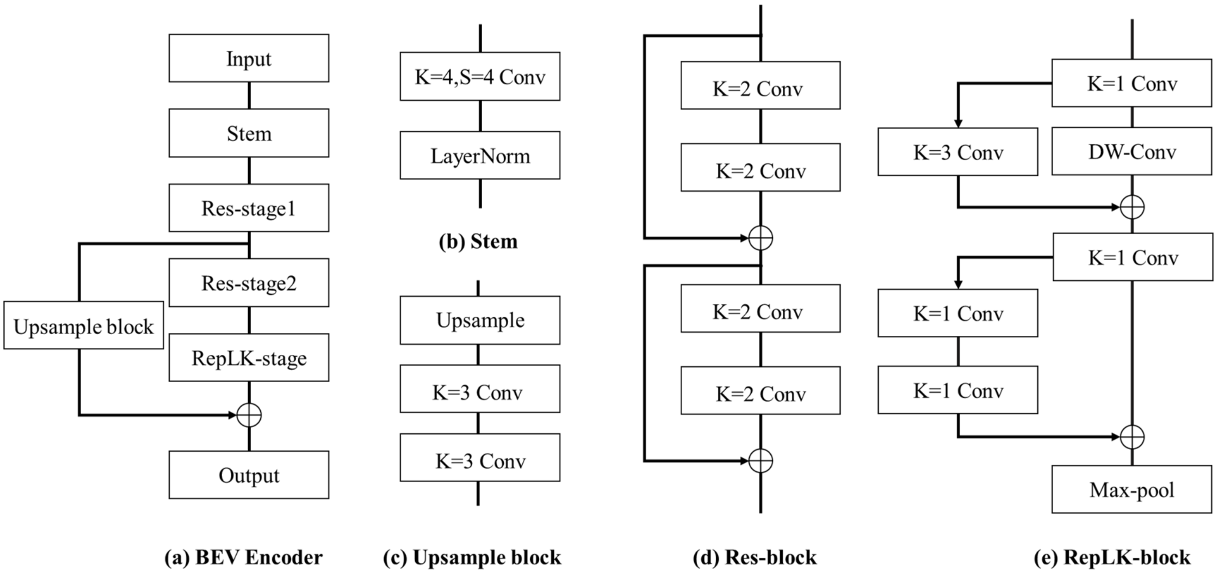
$$\mathbf{F}^{vox} = \mathbf{Attention} \otimes V \oplus \mathbf{F}' \tag{8}$$

where $Attention$ denotes the obtained attentional map. After our proposed feature diffusion module, the final output $\mathbf{F}^{vox}$ of the view transform module, a 3D voxel grid with rich features, is obtained.

### 3.4. BEV Encoder

The BEV feature encoder of many depth-based research studies is ResNet-18 with small receptive fields, while traditional convolution methods are less capable of modeling the long-range relationships of BEV features due to the limitations of the moving window, which cannot focus on the long-range features outside the center of the ego vehicle. In addition, the view transform module causes much information loss, especially distortion of features at long distances. Although the current BEV perception algorithms are generally set within 100 m of the surrounding area, the problem of information loss at long distances is still apparent. This means that many studies using ResNet-18 as a standard BEV feature encoder are weakened in their ability to model the long-range relationships of BEV features by the loss of information at long distances, thus leading to significantly further deterioration of the long-range results compared to the near-range results. Therefore, to enhance the spatial long-range modeling capability, we redesigned a BEV feature encoder with large kernel-size convolutional blocks [21], as shown in Figure 5, to try to focus on the features away from the center of the BEV features.

We assume that the 3D voxel features after the view transform module is $\mathbf{F}^{vox} \in \mathbb{R}^{C_{vox} \times Z_{vox} \times X_{vox}}$, where $C_{bev}$, $Z_{vox}$, $X_{vox}$ denote the number of channels, height, and width of the 3D voxel features initially obtained, respectively. As shown in Figure 5a, the network architecture of our BEV feature encoder consists of three components: a stem part, two stages consisting of ResNet-18 blocks, and a stage consisting of RepLK blocks. In particular, we first employ a convolutional layer with a kernel size of 7, followed by a batch normalization layer and a ReLU activation layer in the stem part, as shown in Figure 5b. After the stem part, we can get the output features as $(Z_{vox}/2, X_{vox}/2)$. Furthermore, we connect the two Res-stages to encode the feature maps further and downsample to $(Z_{vox}/4, X_{vox}/4)$, where the Res-stage consists of the ResNet-18 block shown in Figure 5d. Finally, we adopt several RepLK blocks to form a RepLK stage, as shown in Figure 5e, to sample the feature map $(H_{vox}/8, W_{vox}/8)$. The core components of the RepLK block are a large kernel convolution part and a feedforward part, where a depth-wise convolutional layer of the kernel size of 31 further encodes BEV features, and a $1 \times 1$ convolutional layer in the feedforward part is responsible for changing the number of channels or feature map size. To obtain richer BEV features, we upsample the output features after each downsampling to their original size and then perform channel dimension stitching. Finally, the final output of the BEV encoder is fed into a specific task head to obtain the prediction results.



**(a) BEV Encoder** **(b) Stem** **(c) Upsample block** **(d) Res-block** **(e) RepLK-block**

**Figure 5.** The network architecture of the BEV feature encoder. (**a**) The structure of the BEV encoder. (**b**) The structure of the Stem part. (**c**) The structure of the Upsample block. (**d**) The structure of the ResNet-18 blocks. (**e**) The structure of the RepLK blocks.

## 4. Experiments

Authors should discuss the results and how they can be interpreted from the perspective of previous studies and the working hypotheses. The findings and their implications should be discussed in the broadest possible context. Future research directions may also be highlighted.

### *4.1. Setup*

**Dataset.** We evaluated our proposed model on the nuScenes dataset. The nuScenes dataset contains 1000 scenes, of which 850 are used for training and validation purposes, and the remaining 150 are reserved for testing. Each scene lasts 20 s, providing much temporal information for analysis. The nuScenes dataset contains a comprehensive sensor suite, including six cameras, one LiDAR sensor, and five radar sensors, where each camera has known corresponding internal and external parameters. In total, the dataset contains 40,000 keyframes capturing scenes from multiple angles and sensor modes. The camera images in the dataset have a resolution of 1600 × 900 pixels, ensuring a high level of detail and visual fidelity for visual perception tasks.

**Evaluation Metrices.** We follow the evaluation metrics of traditional segmentation tasks and measure the intersection-over-union (IoU) between the segmentation results and the ground truth. The IoU for each class can be written as follows:

$$\mathrm{IoU}(\mathrm{S}_{p}, \mathrm{S}_g) = \left| \frac{(\mathrm{S}_p \cap \mathrm{S}_g)}{(\mathrm{S}_p \cup \mathrm{S}_g)} \right| \tag{9}$$

And the average IoU for all classes can be written as:

$$\mathrm{mIoU}(\mathrm{S}_p, \mathrm{S}_g) = \frac{1}{N} \sum_{n=1}^{N} \mathrm{IoU}(\mathrm{S}_p, \mathrm{S}_g) \tag{10}$$

where $\mathrm{S}_p \in \mathbb{R}^{H_g \times H_g \times N}$ and $\mathrm{S}_g \in \mathbb{R}^{H_g \times H_g \times N}$ denote the segmentation prediction results and the ground truth, respectively. $H_g$ and $W_g$ denote the height and width of the ground truth, respectively. $N$ is the number of dataset categories.

**Details.** For the image encoder, we employ the ConvNeXt block that has been pre-trained on the ImageNet dataset in advance. Our proposed model and reproduced model are trained on two NVIDIA GeForce RTX 3060 12G GPUs. Except for the specially stated hyperparameters, we follow the settings in ConvNeXt [17] and VAN [18]. For training, we use the AdamW optimizer, whose learning rate is set to $1 \times e^{-3}$ and the weight decay is set to $1 \times e^{-2}$. The loss function is computed using binary-cross-entropy loss functions. For the hyperparameters $(Z, Y, X)$ of the view transform module, we follow the same settings $(200, 8, 200)$ as in the baselines of this task.

### 4.2. Experiment Result

In this section, to evaluate the performance of the proposed model, we comprehensively compare our proposed model with other state-of-the-art methods, including FISHING [26], LSS [11], FIERY [13], CVT [32], GKT [33], TIIM [34], BEVFormer [9], and Simple-BEV [15], as shown in Table 1. For the LSS and the Simple-BEV, we show the retraining results using the same configuration as the results reported in the original papers. The results of CVT and GKT are as reported in the original papers. For the model performance of other methods, we use all the data from this study [15]. For a fair comparison, we use only the single time step model without considering the time model and only consider the model's performance with multi-camera images as input. The results of evaluating the models on nuScenes are shown in the table. The proposed model achieves 45.4 mIoU on the nuScenes dataset, which outperforms most current segmentation methods and is similar to state-of-the-art performance. For the LSS and the Simple-BEV, we retrain with eight batch size settings, and two batch size settings and obtain results of 33.0 mIoU and 43.8 mIoU, respectively. In addition, to further demonstrate the performance of our proposed model, we visualized four key frames of the nuScenes dataset, as shown in Figure 6.

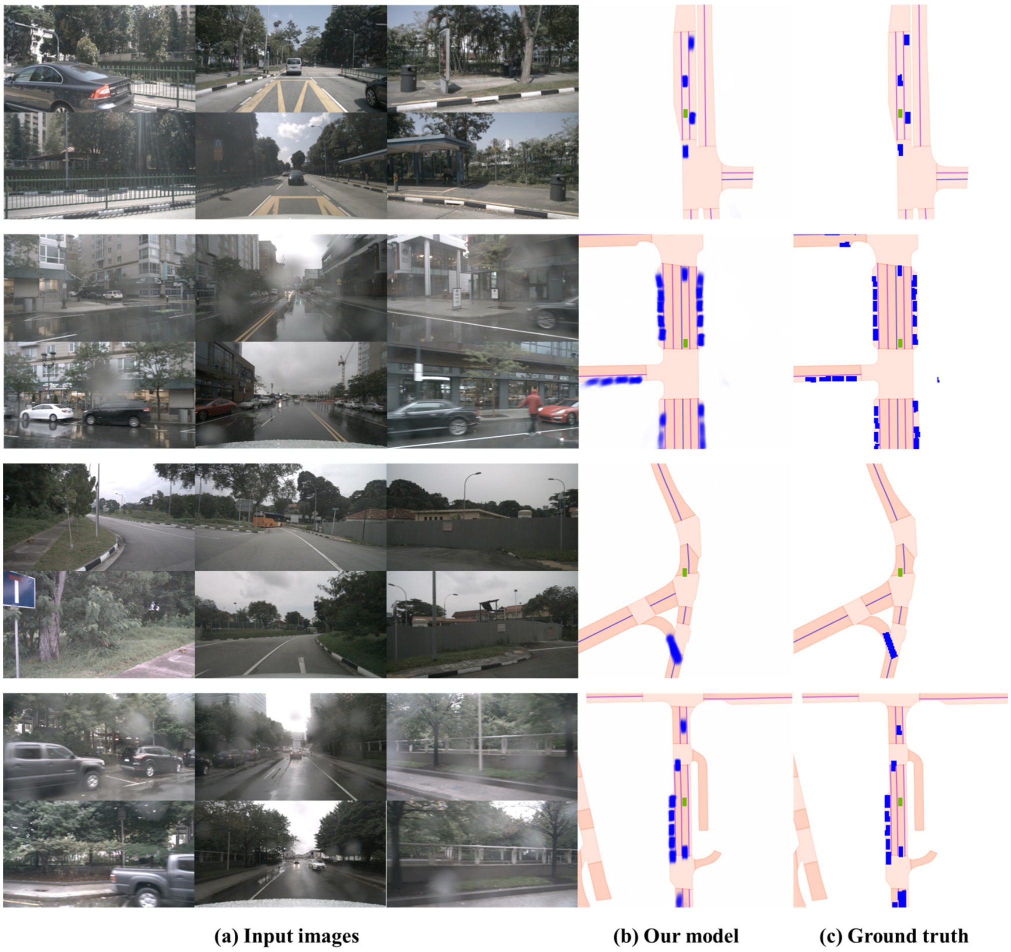


**Figure 6.** The visualization of results includes (**a**) the multi-camera input images, (**b**) the prediction results of our proposed model, and (**c**) the ground truth.

**Table 1.** Comparison of results of BEV segmentation on nuScenes. () denotes our reproduced results of our same setting.

| Method | Lifting | Batch Size | mIoU |
|---|---|---|---|
| FISHING [26] | MLP | - | 30.0 |
| LSS [11] | Depth Estimation | 8 (4) | 33.0 (32.1) |
| FIERY [13] | Depth Estimation | 12 | 35.8 |
| CVT [32] | Deformable Attn. | 16 | 36.0 |
| GKT [33] | Geometry Attn. | 16 | 37.2 |
| TIIM [34] | Ray Attn. | 8 | 38.9 |
| BEVFormer [9] | Deformable Attn. | 1 | 44.4 |
| Simple-BEV [15] | Bilinear | 2 (40) | 42.5 (47.4) |
| Ours | Bilinear-Attn. | 2 | 45.6 |

### *4.3. Detailed Analysis*

In this section, we test the nuScenes dataset using different model combinations to validate our proposed components' effectiveness. Here we emphasize in advance that, without special instructions, our experimental settings are all batch size 2 and input resolution 448 × 800. In order to compare the segmentation performance of the components in detail, we compare the experimental results according to two categories: encoders and view transform modules.

**Image encoder and BEV encoder:** We employ an experimental comparison using different encoder combinations, and the view transform module defaults to our proposed Bilinear-Attention module. Specifically, we select the classical ResNet-101 and ResNet-18 as an image encoder and a BEV encoder, respectively, together with our proposed two encoders, and combine them into four experimental setups for our experiments. As shown in Table 2, Conv-LKA and Res-RepLK denote our proposed image encoder and BEV encoder, respectively. It can be observed that our proposed image encoder and BEV encoder can improve the performance with an increase of 1.8 and 1.3 in mIoU, respectively. Finally, when both of our proposed encoders are used, the model's overall performance is improved by 2.6 mIoU. The growth of FLOPs is also obvious when our proposed encoders are used. As shown in Table 3, we perform comparison experiments on BEV encoders with different kernel sizes. We can observe that a larger convolutional kernel size is beneficial for the model's performance but leads to performance degradation when the kernel size exceeds a certain limit. Experimental results show that the optimal kernel size is 13 × 13, while the larger kernel size does not significantly impact the overall number of parameters in our proposed model. Finally, we use the retrained simple-BEV to compare it with our proposed method and visualize the results, as shown in Figure 7.

**Table 2.** Ablations of the different encoder combinations.

| ResNet-101 | Conv-LKA | ResNet-18 | Res-RepLK | Parameters | FLOPs | mIoU |
|---|---|---|---|---|---|---|
| √ | - | √ | - | 42.1 M | 428.3 G | 43.0 |
| √ | - | - | √ | 40.6 M | 512.7 G | 44.3 |
| - | √ | √ | - | 38.1 M | 552.1 G | 44.8 |
| - | √ | - | √ | 36.6 M | 653.5 G | 45.6 |

**Table 3.** Ablations of the different kernel sizes of the BEV encoder.

| Kernel Size | Parameters | mIoU |
|---|---|---|
| 7 × 7 | 36.5 M | 43.9 |
| 9 × 9 | 36.5 M | 44.5 |
| 13 × 13 | 36.6 M | 45.6 |
| 31 × 31 | 36.7 M | 45.4 |

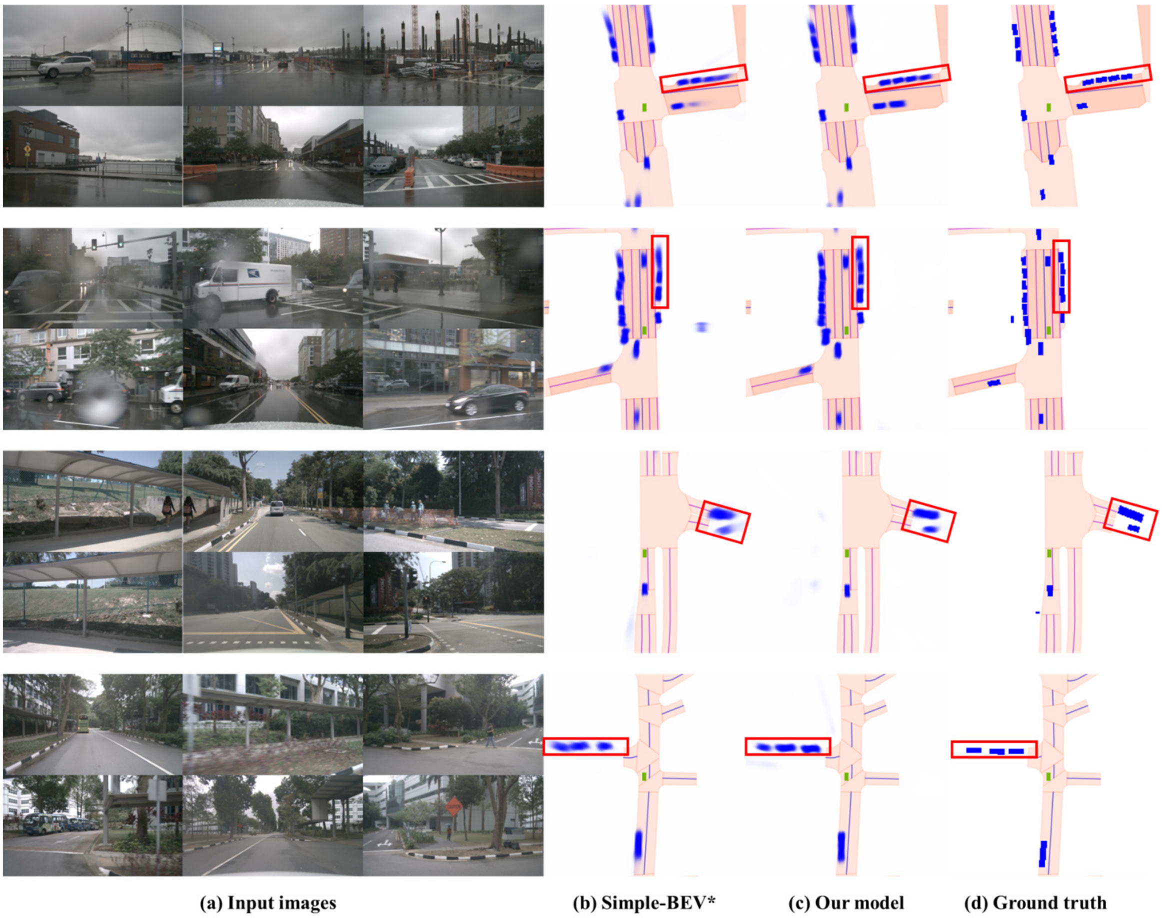


**Figure 7.** The comparison results of our proposed model and Simple-BEV. "*" denotes re-training with equivalent training settings. (**a**) The multi-camera input images; (**b**) the prediction results of retrained Simple-BEV; (**c**) the prediction results of our proposed model; and (**d**) the ground truth.

**View transform module:** We introduce another commonly used depth-based view transform method to compare and analyze with our proposed view transform module. Our proposed Bilinear-Attention module is split into a Bilinear Sample module and a Compressed Attention module for the ablation experiments. As shown in Table 4, Bilinear and Attention denote the Bilinear Sample module and the Compressed Attention module, respectively. LKA-RepLK indicates that both of our proposed encoders are used. We can observe that using the Bilinear Sample module alone does not perform as well as the depth prediction, but it is very close. Moreover, it can be observed that our proposed view transform module achieves an 8.2 improvement over the MLP approach in mIoU. Using the Bilinear Sample module alone also yields a 7.5 improvement in mIoU.

**Table 4.** Ablations of the different view transform module combinations.

| Encoder | MLP | Depth | Bilinear | Attention | mIoU |
|---|---|---|---|---|---|
| LKA-RepLK | √ | - | - | - | 37.2 |
| LKA-RepLK | - | √ | - | - | 44.8 |
| LKA-RepLK | - | - | √ | - | 44.7 |
| LKA-RepLK | - | - | √ | √ | 45.6 |

## 5. Conclusions

This study proposes a camera-based model to accomplish the semantic segmentation task from the BEV perspective. To obtain an image view encoder with more powerful encoding performance and capable of capturing long-distance relationships, we redesign the backbone network with the Large Kernel Attention module. In addition, we propose the Bilinear Sample module to complete the lifting operation instead of directly predicting the depth, and then refine the 3D features with our proposed Compressed Attention module. We redesign the structure of the BEV encoder with RepLK to address the problem of long-range distortion of BEV features. We evaluated our proposed model on the nuScenes dataset. Our experiment results demonstrate that our model outperforms other models with equivalent training settings on the segmentation task while approaching state-of-the-art performance. While our current work focuses on semantic segmentation, we recognize the significance of expanding our evaluation to include object detection tasks. We are committed to enhancing computational efficiency and model size, ensuring that our approach remains practical for real-world applications. This optimization will contribute to the scalability and deployability of our model. Meanwhile, we recognize the challenges associated with detecting small or distant objects in the Bird's-Eye view. Our future work will involve dedicated optimization strategies to address these challenges and improve the model's performance in such scenarios. Finally, point clouds provide invaluable depth and spatial information, we envision integrating point cloud data alongside other sensor modalities to augment our model's understanding of the environment.

**Author Contributions:** Conceptualization, K.L. and X.W.; methodology, K.L.; software, K.L.; validation, K.L.; formal analysis, K.L.; investigation, X.W.; resources, W.Z.; data curation, X.W.; writing—original draft preparation, K.L.; writing—review and editing, X.W.; visualization, X.W.; supervision, W.Y.; project administration, W.Y.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision And Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
2. Li, Q.; Wang, Y.; Wang, Y.; Zhao, H. Hdmapnet: An online hd map construction and evaluation framework. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 4628–4634.
3. Mallot, H.A.; Bülthoff, H.H.; Little, J.; Bohrer, S. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biol. Cybern.* **1991**, *64*, 177–185. [CrossRef] [PubMed]
4. Reiher, L.; Lampe, B.; Eckstein, L. A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–7.
5. Zhu, M.; Zhang, S.; Zhong, Y.; Lu, P.; Peng, H.; Lenneman, J. Monocular 3D vehicle detection using uncalibrated traffic cameras through homography. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 3814–3821.
6. Song, L.; Wu, J.; Yang, M.; Zhang, Q.; Li, Y.; Yuan, J. Stacked homography transformations for multi-view pedestrian detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 6049–6057.
7. Pan, B.; Sun, J.; Leung, H.Y.T.; Andonian, A.; Zhou, B. Cross-view semantic segmentation for sensing surroundings. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4867–4873. [CrossRef]
8. Roddick, T.; Cipolla, R. Predicting semantic map representations from images using pyramid occupancy networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11138–11147.
9. Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; Dai, J. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 1–18.
10. Liu, Y.; Wang, T.; Zhang, X.; Sun, J. Petr: Position embedding transformation for multi-view 3d object detection. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 531–548.
11. Philion, J.; Fidler, S. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XIV 16, 2020. pp. 194–210.

12. Reading, C.; Harakeh, A.; Chae, J.; Waslander, S.L. Categorical depth distribution network for monocular 3D object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual. 19–25 June 2021; pp. 8555–8564.
13. Hu, A.; Murez, Z.; Mohan, N.; Dudas, S.; Hawke, J.; Badrinarayanan, V.; Cipolla, R.; Kendall, A. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 15273–15282.
14. Xie, E.; Yu, Z.; Zhou, D.; Philion, J.; Anandkumar, A.; Fidler, S.; Luo, P.; Alvarez, J.M. M$^2$BEV: Multi-Camera Joint 3D Detection and Segmentation with Unified Birds-Eye View Representation. *arXiv* **2022**, arXiv:2204.05088.
15. Harley, A.W.; Fang, Z.; Li, J.; Ambrus, R.; Fragkiadaki, K. Simple-BEV: What really matters for multi-sensor bev perception? In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 2759–2765.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
17. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
18. Guo, M.-H.; Lu, C.-Z.; Liu, Z.-N.; Cheng, M.-M.; Hu, S.-M. Visual attention network. *arXiv* **2022**, arXiv:2202.09741. [CrossRef]
19. Huang, J.; Huang, G.; Zhu, Z.; Ye, Y.; Du, D. Bevdet: High-performance multi-camera 3D object detection in bird-eye-view. *arXiv* **2021**, arXiv:2112.11790.
20. Li, Y.; Ge, Z.; Yu, G.; Yang, J.; Wang, Z.; Shi, Y.; Sun, J.; Li, Z. Bevdepth: Acquisition of reliable depth for multi-view 3D object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 1477–1485.
21. Ding, X.; Zhang, X.; Han, J.; Ding, G. Scaling up your kernels to 31 × 31: Revisiting large kernel design in cnns. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11963–11975.
22. Wang, Y.; Chao, W.-L.; Garg, D.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8445–8453.
23. You, Y.; Wang, Y.; Chao, W.-L.; Garg, D.; Pleiss, G.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-lidar++: Accurate depth for 3D object detection in autonomous driving. *arXiv* **2019**, arXiv:1906.06310.
24. Huang, J.; Huang, G. Bevdet4d: Exploit temporal cues in multi-camera 3D object detection. *arXiv* **2022**, arXiv:2203.17054.
25. Lu, C.; van de Molengraft, M.J.G.; Dubbelman, G. Monocular semantic occupancy grid mapping with convolutional variational encoder-decoder networks. *IEEE Robot. Autom. Lett.* **2019**, *4*, 445–452. [CrossRef]
26. Hendy, N.; Sloan, C.; Tian, F.; Duan, P.; Charchut, N.; Xie, Y.; Wang, C.; Philbin, J. Fishing net: Future inference of semantic heatmaps in grids. *arXiv* **2020**, arXiv:2006.09917.
27. Zou, J.; Zhu, Z.; Huang, J.; Yang, T.; Huang, G.; Wang, X. HFT: Lifting Perspective Representations via Hybrid Feature Transformation for BEV Perception. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 7046–7053.
28. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
29. Can, Y.B.; Liniger, A.; Paudel, D.P.; Van Gool, L. Structured bird's-eye-view traffic scene understanding from onboard images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 15661–15670.
30. Wang, Y.; Guizilini, V.C.; Zhang, T.; Wang, Y.; Zhao, H.; Solomon, J. Detr3d: 3D object detection from multi-view images via 3D-to-2D queries. In Proceedings of the Conference on Robot Learning, Auckland, New Zealand, 14–18 December 2022; pp. 180–191.
31. Liu, Y.; Yan, J.; Jia, F.; Li, S.; Gao, Q.; Wang, T.; Zhang, X.; Sun, J. Petrv2: A unified framework for 3D perception from multi-camera images. *arXiv* **2022**, arXiv:2206.01256.
32. Zhou, B.; Krähenbühl, P. Cross-view transformers for real-time map-view semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13760–13769.
33. Chen, S.; Cheng, T.; Wang, X.; Meng, W.; Zhang, Q.; Liu, W. Efficient and robust 2D-to-bev representation learning via geometry-guided kernel transformer. *arXiv* **2022**, arXiv:2206.04584.
34. Saha, A.; Mendez, O.; Russell, C.; Bowden, R. Translating images into maps. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 9200–9206.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.