BEV-SAN: Accurate BEV 3D Object Detection via Slice Attention Networks

Xiaowei Chi[1,2*] Jiaming Liu[1*] Ming Lu[1*] Rongyu Zhang[1]
Zhaoqing Wang[3] Yandong Guo[4] Shanghang Zhang[1†]
[1]National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University
[2]The Chinese University of Hong Kong [3]The University of Sydney [4]AI2Robotics

Abstract

Bird's-Eye-View (BEV) 3D Object Detection is a crucial multi-view technique for autonomous driving systems. Recently, plenty of works are proposed, following a similar paradigm consisting of three essential components, i.e., camera feature extraction, BEV feature construction, and task heads. Among the three components, BEV feature construction is BEV-specific compared with 2D tasks. Existing methods aggregate the multi-view camera features to the flattened grid in order to construct the BEV feature. However, flattening the BEV space along the height dimension fails to emphasize the informative features of different heights. For example, the barrier is located at a low height while the truck is located at a high height. In this paper, we propose a novel method named BEV Slice Attention Network (BEV-SAN) for exploiting the intrinsic characteristics of different heights. Instead of flattening the BEV space, we first sample along the height dimension to build the global and local BEV slices. Then, the features of BEV slices are aggregated from the camera features and merged by the attention mechanism. Finally, we fuse the merged local and global BEV features by a transformer to generate the final feature map for task heads. The purpose of local BEV slices is to emphasize informative heights. In order to find them, we further propose a LiDAR-guided sampling strategy to leverage the statistical distribution of LiDAR to determine the heights of local slices. Compared with uniform sampling, LiDAR-guided sampling can determine more informative heights. We conduct detailed experiments to demonstrate the effectiveness of BEV-SAN. Code will be released.

1. Introduction

Object detection is an essential computer vision task, which has wide applications in security, robotics, autonomous driving, etc. With the development of Deep Neural Networks (DNNs), a huge amount of methods are proposed for 2D [7–9, 18, 25, 26] and 3D [5, 24, 27, 33] object detection. As there are too many methods, we focus our introduction on the cutting-edge multi-view camera-based 3D object detection, which has gained increasing attention from the community. The Bird's-Eye-View (BEV) is a unified representation of the surrounding scene and is suitable for autonomous driving tasks. Therefore, plenty of 3D object detection methods [3, 11, 12, 14–17, 30, 32] are proposed for multi-view BEV perception recently.

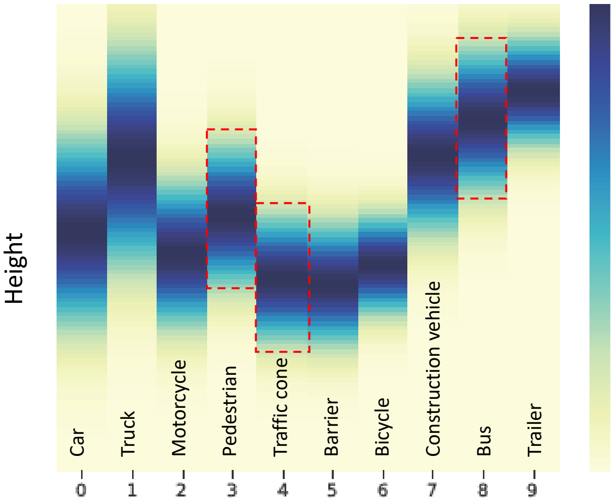

Figure 1. The statistics of 3D bounding boxes along the height dimension.

Although the model architectures of those methods are different, they commonly follow a similar paradigm consisting of three essential components including camera feature extraction, BEV feature extraction, and task heads. Among the three components, BEV feature construction is BEV-specific compared with 2D tasks. [16] presents a new framework that learns a unified BEV representation with spatio-temporal transformers. They first lift each query on the flattened BEV grid to a pillar-like query and then project the

[*]Equal contribution: liujiaming.pku@gmail.com
[†]Corresponding author: shanghang@pku.edu.cn

sampled 3D points to 2D views. The extracted features of hit views are weighted and summed as the output of spatial cross-attention. [15] first predicts the depth for RGB input and projects the image features to frustum space. Then they sum up the frustum features that fall into the same flatted BEV grid. Both methods have pros and cons, while they all flatten the BEV space along the height dimension.

Motivated by the fact that different object classes locate at different heights. For instance, barrier is located at a low height while the truck is located at a high height. Flattening the BEV space along the height dimension fails to exploit the benefit of different heights. In this paper, we propose a novel method named BEV Slice Attention Network (BEV-SAN) to explore the intrinsic properties of different heights. We first sample along the height dimension to build the global and local BEV slices, which are represented as the upper and lower bounds of BEV slice height. The global slices are similar to former works [15, 16], which aim at covering the large height range of BEV space, while the local BEV slices aim at emphasizing informative heights. We aggregate the features from multi-view cameras to construct the features of global and local BEV slices. To merge the global and local slices, we first use the height attention mechanism to fuse the global and local slices separately. Then we adopt a transformer to fuse the merged global and local features. The final fused feature map is used for task-specific heads. In this paper, we mainly conduct the evaluation of BEV-SAN on 3D object detection. It is to be noted that our method can also be used in other BEV perception tasks such as map segmentation and planning.

In order to improve the performance, we further propose a LiDAR-guided sampling strategy to leverage the statistical distribution of LiDAR to determine the optimal heights of local slices. We project the LiDAR points to the BEV space and calculate the histogram along the height dimension. According to the histogram, we can sample the upper and lower height bounds of local slices. Compared with uniform sampling or random sampling, our strategy can choose informative ranges for BEV perception. We want to point out that we only use LiDAR data to build the local BEV slices. Our contributions can be concluded as follows:

- We propose a novel method named BEV Slice Attention Network (BEV-SAN) that exploits the features of different heights in BEV space, achieving an accurate performance of BEV 3D object detection.
- We present a LiDAR-guided sampling strategy to determine the optimal heights of local slices, resulting in informative ranges for BEV perception.
- We conduct detailed experiments to demonstrate the effectiveness of our method. Our method can also be applied to other BEV perception tasks like map segmentation and planning.

2. Relate work

Monocular 3D object detection Monocular 3D object detection is a useful but challenging technique in autonomous driving since it needs to predict the 3D bounding boxes from a single 2D image. Deep3DBox [22] firstly regresses relatively stable 3D bounding box properties using DNNs and combines them with geometric constraints to generate the final results. M3D-RPN [1] designs depth-aware convolutional layers and 3D region proposal network, significantly improving the performance of monocular 3D object detection. SMOKE [19] predicts a 3D bounding box for each detected 2D object by combining a single keypoint estimate with regressed 3D variables. FCOS3D [29] proposes a one-stage framework that predicts the decoupled 2D and 3D attributes for 3D targets. MonoDLE [21] quantifies the impact introduced by each sub-task of monocular 3D object detection and proposes three strategies to reduce the localization error. PGD [28] constructs geometric relation graphs across predicted objects and uses the graph to improve the depth estimation for monocular 3D object detection. MonoPair [6] improves monocular 3D object detection by considering the relationship of paired samples. RTM3D [13] predicts the nine perspective key points in 3D space and recovers the dimension, location, and orientation from the nine key points. MonoFlex [35] proposes a flexible framework that explicitly decouples the truncated objects and adaptively combines multiple approaches for object depth estimation. GUP-Net [20] proposes to tackle the error amplification problem introduced by the projection process. MonoDETR [34] introduces a novel framework using a depth-guided transformer and achieves state-of-the-art performance on benchmarks.

Multi-View BEV 3D object detection As a unified representation of the surrounding scene, BEV 3D object detection is becoming prevailing in the multi-view camera systems. Recently, plenty of methods are proposed for multi-view BEV 3D object detection. DETR3D [30] uses a sparse set of 3D object queries to index the extracted 2D features from multi-view camera images. They make the bounding box prediction per query using the set-to-set loss. BEVDet [12] first predicts the depth for each camera image and then projects the extracted image features to BEV space by the LSS operation [23]. Finally, the task-specific head is constructed upon the BEV feature. BEVDet4D [11] fuses the feature from the previous frame with the current frame to lift the BEVDet paradigm from 3D space to spatial-temporal 4D space. BEVFormer [16] exploits both the spatial and temporal information by interacting with spatial and temporal space through pre-defined grid-shaped BEV queries. PETR [17] encodes the position information of 3D coordinates into image features and performs end-to-end object detection based on 3D position-aware features. BEVDepth [15] reveals that the quality of intermediate depth is the

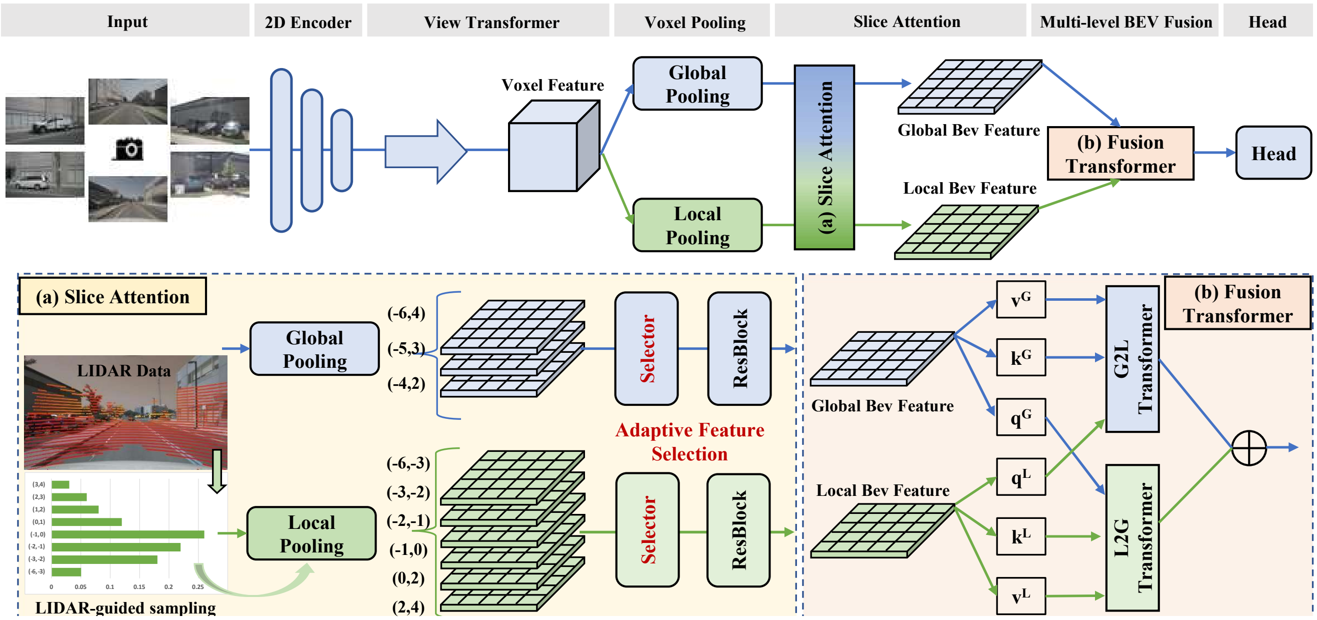

Figure 2. The pipeline of the proposed SAN method. Our method constructs the BEV feature based on the global and local slices. We use a two-stage fusion strategy to merge the features of global and local slices for task heads.

key to improving multi-view 3D object detection. They get explicit depth supervision utilizing encoded intrinsic and extrinsic parameters. PolarDETR [4] uses the Polar Parametrization for 3D detection by reformulating position parametrization, velocity decomposition, perception range, label assignment, and loss function in the polar coordinate system. BEVStereo [14] introduces an effective temporal stereo method to dynamically select the scale of matching candidates for multi-view stereo. They further design an iterative algorithm to update more valuable candidates, making it adaptive to moving candidates. STS [31] proposes a surround-view temporal stereo technique to leverage the geometry correspondence between frames across time to improve the quality of depth.

3. Methods

Our method follows the pipeline of existing methods such as BEVDepth [15], which consist of three components: camera feature extraction, BEV feature construction, and task heads. To be more specific, Given an input multi-view image $I_k \in R^{3\times H\times W}$, we adopt a shared backbone model to extract the feature $F_k \in R^{C\times H_f\times W_f}$, where k is the index of the camera. we also predict the depth distribution map for each input image $D_k \in R^{D\times H_f\times W_f}$. Then we project the camera features to viewing frustum $V_k \in R^{C\times D\times H_f\times W_f}$ and construct the flattened BEV feature $B \in R^{C\times H_e\times W_e}$ with the proposed **Slice Attention Module**. Finally, the task-specific heads are applied to the BEV feature. We will first introduce the motivation in Sec. 3.1 and then present the proposed Slice Attention Module in Sec. 3.2. The whole framework of our method is illustrated in Fig. 2.

Table 1. The mAP results of Traffic Cone, Person and Bus with BEV slices of different height ranges.

Height	Traffic Cone	Person	Bus
$[-2, 0]$	0.087	0.0	0.001
$[0, 1]$	**0.436**	0.217	0.273
$[1, 2]$	0.367	0.245	0. 307
$[2, 3]$	0.406	**0.265**	0.340
$[3, 4]$	0.368	0.257	**0.348**

3.1. Motivation

In the practical applications of autonomous driving, the detection targets vary in shape and size, causing severe bias in visual-based learning. For example, barrier is located at a low height while the truck is located at a high height. However, existing methods like BEVDepth [15] sum up the frustum features that fall into the same flattened BEV grid. Therefore, they fail to exploit the benefit of different heights for BEV perception. In this section, we intend to demonstrate the motivation for slicing the BEV space based on different heights. We first visualize the heights of annotated 3D bounding boxes according to their object classes. As shown in Fig. 1, different object classes actually have different height distributions. This is consistent with our

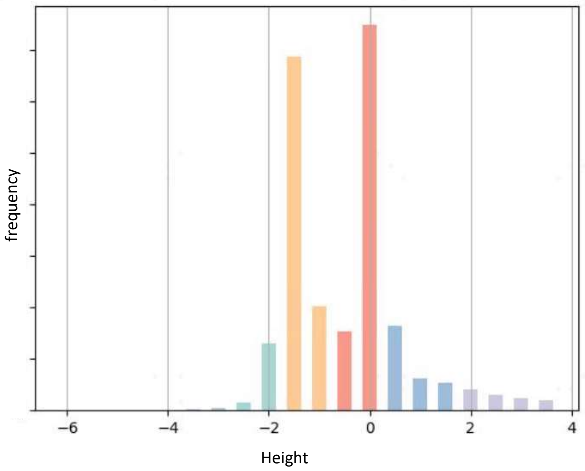

Figure 3. The statistics of LiDAR points along the height dimension. We use this LiDAR histogram to guide the sampling of local slices, which emphasizes the informative heights.

motivation.

To further study this motivation, we adjust the height range of BEVDepth [15] and evaluate the 3D object detection performance of different classes as shown in Tab. 1. As can be seen, the traffic cone, which is lower compared with person and bus, shows obviously different performances at different height ranges (0.436 in [0,1] and 0.368 in [3,4] separately). This indicates that the height range will greatly affect the detection performance of different object classes. This observation inspires us to take better advantage of different heights to improve detection performance. We will introduce the proposed Slice Attention Module in the next section.

3.2. Slice Attention Module

In this section, we introduce the proposed Slice Attention Module. We define the slice using the height range in BEV space. We will first explain how to sample the BEV space to generate the global and local slices. The global slices are sampled to cover the large height ranges of BEV space. The local slices are sampled to emphasize the informative heights. Then we present our method to fuse the sampled global and local slices with an attention mechanism. Finally, we fuse the global feature and local feature for the task heads.

3.2.1 Global and Local Slices

For the multi-view images, we can extract the features by a shared backbone model $F_k \in R^{C \times H_f \times W_f}$, where k is the index of the camera. We can aggregate the image features to construct the BEV feature $B_s \in R^{C \times H_e \times W_e}$ given the height range $s = [l, u]$ in BEV space. We define a height range as a BEV slice.

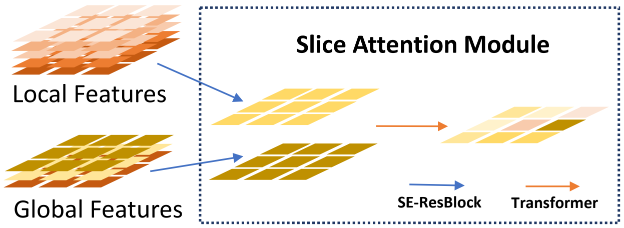

Figure 4. The pipeline of slice feature fusion. Our fusion strategy contains two stages. The first stage is based on channel attention to merge local slices and global slices separately. The second stage is based on a dual-branch transformer, which explores the spatial attention.

Global Slices We empirically determine the global slices as $\{s_g\} = [[-6, 4], [-5, 3], [-4, 2]]$. Although the largest range $[-6, 4]$ contains the overall information of the whole space, the corresponding BEV feature representation is significantly different from $[-5, 3]$ or $[-4, 2]$. Since the height information is viewed as channel dimension, we adopt a channel-wise attention [10] to adaptively aggregate the multiple global-level slices. The attention mechanism between three global slices provides a learnable way to fully explore different semantic knowledge and thus improve the global contextual representation in BEV latent space. The attention between three global slices will be necessary to help improve the performance at the global level. We denote the constructed features of global slices as $\{B^i_{s_g}\}$.

Local Slices The goal of local slices is to emphasize the informative height ranges. We construct the local slices by sampling from the overall range $[-6, 4]$. In order to sample reasonable local slices, we present a LiDAR-guided sampling strategy to determine the optimal heights of local slices. We transform the LiDAR points to BEV space and calculate the histogram along the height dimension as shown in Fig. 3. We find that most LiDAR points are located around -2 and 0. However, those regions contain small objects while regions outside [-2,2] contain large objects. In order to sample more effective local slices, we design a novel strategy to consider the distribution differences between classes. Specifically, we accumulate the histogram and choose the local slices from the accumulated distribution. We slice the overall range $[-6, 4]$ to six bins, including $[-6, -3]$, $[-3, -2]$, $[-2, -1]$,$[-1, 0]$, $[0, 2]$, and $[2, 4]$. Similar to global slices, we also utilize the channel attention mechanism to reweight the local slices, which effectively aggregates the information of different heights. The local slices are denoted as $\{s_l\}$ and the aggregated features are denoted as $\{B^j_{s_l}\}$.

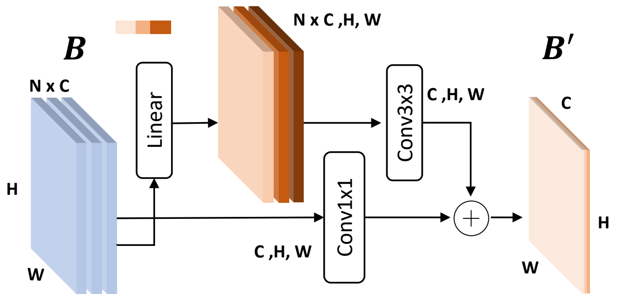

Figure 5. Illustration of the SE attention residual block for merging local and global slices separately.

3.2.2 Fusion of Slice Features

After obtaining the global features $\{B^i_{s_g}\}$ and local features $\{B^j_{s_l}\}$, we can fuse them together into the feature map for task heads. Our method introduces a two-stage attention structure to progressively fuse the features as shown in Fig. 4. In the first stage, we fuse the global features and local features via the attention mechanism. This will generate the global fused feature $B_g \in R^{C\times H_e\times W_e}$ and local fused feature $B_l \in R^{C\times H_e\times W_e}$. In the second stage, we use a transformer to fuse B_g and B_l and generate the feature map for task heads.

To be more specific, in the first stage, we adopt the attention mechanism similar to the Squeeze-and-Excitation (SE) operation [10]. Taking local features as an example, the features of local slices are denoted as $\{B^j_{s_l}\} \in R^{J\times C\times H_e\times W_e}$, where J is the number of local slices. As shown in Fig. 5, we first use 1x1 convolution to reduce the channel number from $J\times C$ to C. We use global average pooling to extract the $J\times C$ feature and reweight the input feature. Another 3x3 convolution is used to reduce the channel number from $J\times C$ to C. Finally, we add the two parts to deliver the fused feature $B_l \in R^{C\times H_e\times W_e}$. The features of global slices $\{B^i_{s_g}\}$ can be fused into B_g in the same way.

In the second stage, we need to fuse B_g and B_l with a transformer. As shown in Fig. 2, the transformer contains two branches (denoted as G2L and L2G) using B_g and B_l as the inputs. One feature will be transformed into a set of *Key/Value* pairs to interact with features from the other. For example, the *Query/Key/Value* pair in G2L Transformer is: $q = q^L, k = k^G, v = v^G$ where L stands for local-level and G represents global-level. Finally, we sum up the outputs of the two branches to obtain the final feature map for task heads.

4. Experiment

In this section, we first give the experimental details in Sec. 4.1. Then we evaluate the proposed SAN on nuScenes [2] and compare it with several baseline methods in Sec. 4.2. Besides, we also conduct detailed ablation study to evaluate each component of our method in Sec. 4.3. We further show the computational cost in Sec. 4.4.

4.1. Experimental Details

Dataset We use the nuScenes [2] dataset to evaluate the performance of our distillation framework. NuScenes contains 1k sequences, each of which is composed of six groups of surround-view camera images, one group of Lidar data, and their sensor information. The camera images are collected with the resolution of 1600 × 900 at 12Hz and the LiDAR frequency for scanning is 20Hz. The dataset provides object annotations every 0.5 seconds, and the annotations include 3D bounding boxes for 10 classes {Car, Truck, Bus, Trailer, Construction vehicle, Pedestrian, Motorcycle, Bicycle, Barrier, Traffic cone }. We follow the official split that uses 750, 150, and 150 sequences as training, validation, and testing sets respectively. So we get 28130 batches of data for training, 6019 batches for validation, and 6008 batches for testing.

Metrics We use mean Average Precision(mAP) and Nuscenes Detection Score(NDS) as our main evaluation metrics. We also adopt other officially released metrics concluding Average Translation Error (ATE), Average Scale Error (ASE), Average Orientation Error (AOE), Average Velocity Error (AVE), and Average Attribute Error (AAE). Note that NDS is a weighted sum of mAP and other metric scores.

Implementation Details We use BEVDepth [15] as the baseline. The image backbone is ResNet-50, and the input image size is [256,704]. Following BEVDepth, image augmentation includes random cropping, random scaling, random flipping, and random rotation. The BEV feature generated by the model is also augmented by random scaling, random flipping, and random rotation. The base learning rate is 2e-4, and the batch size is 6 for each GPU. During training, we use 8 V100 GPUs, and the training takes 40 epochs. We decay the learning rate on epochs 23 and 33 with ratio $\alpha = 1e-7$. To conduct a fair comparison, all methods share these settings. Apart from BEVDepth, we also evaluate the proposed method on the BEVDet [12].

4.2. Main Results

Results on nuScenes val set We first evaluate our method on nuScenes val set. The baseline methods are BEVDet and BEVDepth. We report the results of BEVDepth under different height ranges [-5,3], [-4,2], and [-6,4]. The default height range of BEVDet and BEVDepth is [-5,3]. As can be seen from Tab. 2, our method can improve the baseline method by 0.03 in NDS and all the evaluation metrics are also improved. To further evaluate our method, we conduct the experiments with CBGS strat-

Table 2. 3D Object Detection Results on nuScenes val set without CBGS

Method	Voxel Range	Backbone	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
BEVDepth	[-5,3]	R50	0.328	0.293	0.742	0.283	0.758	1.216	0.403
BEVDepth	[-4,2]	R50	0.330	0.293	0.740	0.282	0.745	1.201	0.397
BEVDepth	[-6,4]	R50	0.336	0.296	0.732	0.283	0.713	1.218	0.396
BEVDet	[-5,3]	R50	0.298	0.274	0.754	0.295	0.881	1.25	0.418
SANet(BEVDet)	slice	R50	0.320	0.292	0.746	0.286	0.797	1.167	0.403
SANet(BEVDepth)	Slice	R50	0.366	0.310	0.705	0.278	0.608	1.070	0.300
BEVDepth	[-6,4]	R101	0.371	0.313	0.697	0.278	0.579	1.086	0.304
SANet(BEVDepth)	Slice	R101	0.379	0.319	0.681	0.270	0.567	0.996	0.290

Table 3. 3D Object Detection Results on nuScenes val set with CBGS.

Method	Voxel Range	Backbone	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
BEVDet	[-5,3]	R50	0.372	0.299	0.724	0.273	0.578	0.929	0.266
PETR	[-5,3]	R50	0.381	0.313	0.768	0.278	0.564	0.923	0.225
BEVDepth	[-5,3]	R50	0.470	0.341	0.619	0.273	0.451	0.462	0.198
SANet(BEVDepth)	Slice	R50	0.482	0.351	0.618	0.271	0.434	0.426	0.192

Table 4. 3D Object Detection Results of Each Object Class on nuScenes val set.

Method	Truck	trailer	Car	Bus	Pedestrian	Motorcycle	Bicycle	Traffic cone	Barrier
BEVDepth	0.237	0.153	0.466	0.332	0.247	0.289	0.267	0.417	0.465
SANet	0.244	0.165	0.491	0.350	0.265	0.302	0.272	0.432	0.503
BEVDepth+CBGS	0.269	0.171	0.545	0.352	0.351	0.318	0.250	0.530	0.559
SANet+CBGS	0.272	0.166	0.555	0.358	0.365	0.315	0.282	0.544	0.582

egy [36], which will take much longer training time. As can be seen from Tab. 3, our method can still improve performance even with the CBGS strategy. We conduct this experiment based on ResNet-50 in consideration of computation cost.

Results of different object classes Since the motivation of our method is to handle different object classes with different heights. Therefore, we show the results of different object classes in Tab. 4. We compare the mAP of the proposed SAN and baseline methods. For the results without CBGS strategy, SAN outperforms the baseline BEVDepth in each object class. The performance gain of the traffic cone even reaches 0.038. For the results with CBGS, the SAN also shows significant improvement. For example, our method improves the baseline BEVDepth by 0.032 in bicycles and 0.023 in traffic cones. These results show that our method gives different attention to objects with different shapes.

Qualitative results We show the qualitative results of the baselines and our method. As can be seen from Fig. 6, the proposed SAN improves the performance of 3D object detection. In this figure, we compare the results of SAN and BEVDepth [15]. We also show the feature visualization in Fig. 7. As can be seen from this figure, the original BEV feature does not capture the top left object, while our method fuses the features of different slices. Therefore, the enhanced BEV feature successfully captures the top left object.

Table 5. Ablation Study of Global and Local Slices.

Local	Global	NDS	mAP
·	·	0.330	0.296
✓	·	0.351	0.310
·	✓	0.343	0.307
✓	✓	**0.366**	**0.310**

4.3. Ablation study

Global and Local Slices Our method uses both the global and local slices to construct the BEV feature. The global slices aim to cover the large ranges of BEV height, while the local slices aim to emphasize the informative heights. Therefore, we conduct an ablation study to evaluate the contributions of global and local slices. As shown in Tab. 5, both types contribute to performance improvement.

LiDAR-Guided Sampling In this paper, we propose to use a LiDAR-guided sampling strategy to obtain the local

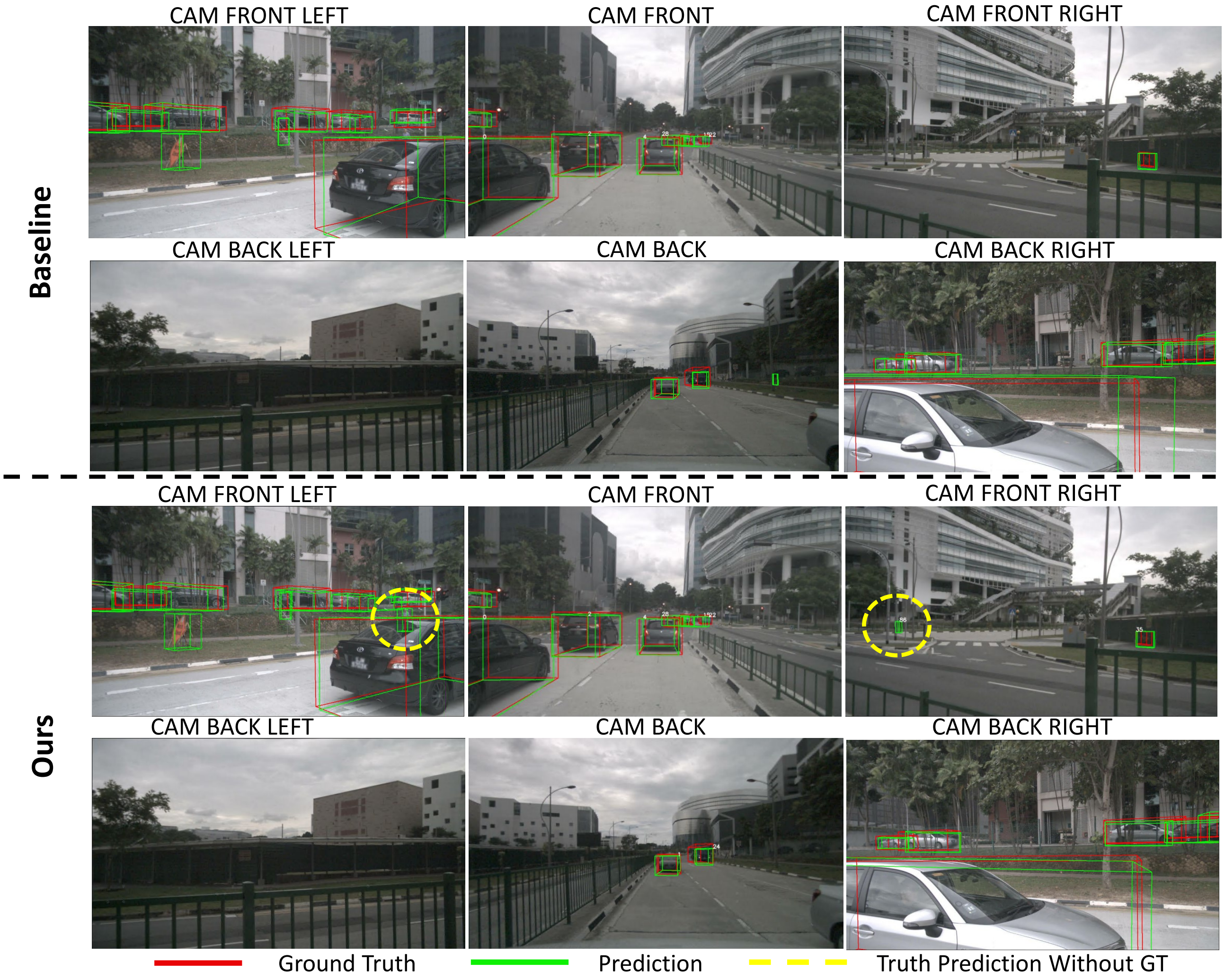

Figure 6. The visualization result of baseline and the SAN. The red box denotes the ground truth, and the green box is the prediction. In this case, our method gives a more accurate prediction and gives two correct predictions of pedestrians in the yellow circles that do not have labels.

slices. Therefore, we conduct the ablation study to evaluate the contribution of this component. For a fair comparison, we all use the global slices. As can be seen from Tab. 6, the LiDAR-guided sampling strategy can improve the NDS of average local sampling by 0.007, demonstrating the effectiveness of the proposed sampling strategy.

Table 6. Ablation Study of LiDAR-Guided Sampling.

Statistics Local	NDS	mAP
.	0.359	0.310
✓	**0.366**	**0.310**

Table 7. Ablation Study of Fusion Strategy.

Method	Voxel Range	NDS ↑	mAP ↑
SA-Mean	local Only	0.332	0.296
SA-SE	local Only	0.350	0.298
SA-SE-Mean	local + Global	0.359	0.311
SA-SE-SE	local + Global	0.361	0.310
SA-SE-Trans	local + Global	0.366	0.310

Fusion Strategy The fusion strategy also plays an important role in merging the local and global slices. In short, our fusion strategy contains two stages. The first stage merges the local and global slices, respectively. The second stage fuses the merged local and global features for task heads. In this part, we evaluate the fusion strategy based on BEVDepth with ResNet-50. Mean denotes adding the BEV features together. SE denotes the Squeeze-and-Excitation Attention residual block. Trans means the designed two

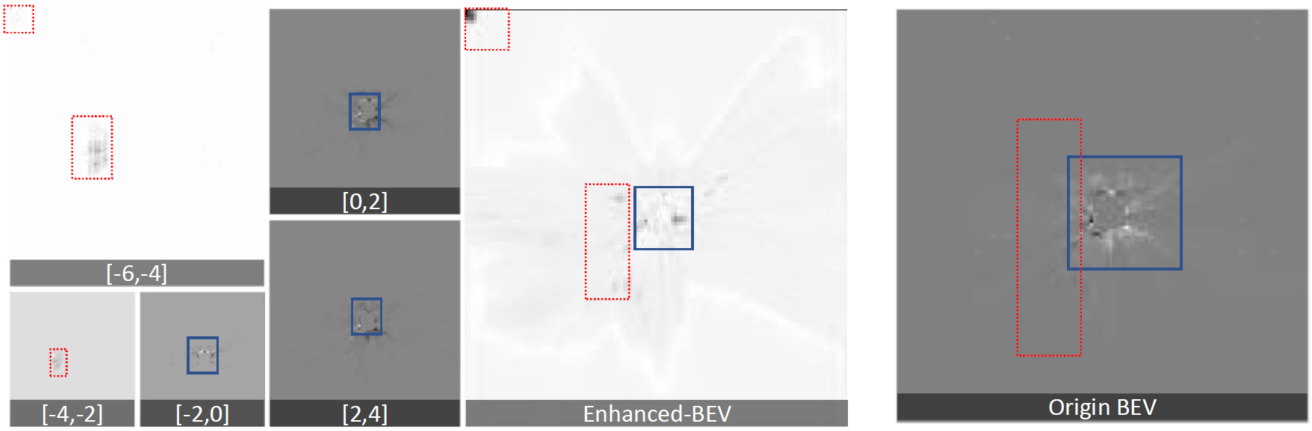

Figure 7. The visualization result of the baseline BEV feature and SAN BEV feature. As can be seen, the features of different slices can capture different objects. For example, the original feature fails to capture the top-left object, while our enhanced feature successfully captures this object.

Figure 8. Computational cost. We compare the proposed SANet with the baseline method BEVDepth with ResNet-50 and ResNet-101 as the backbones. As can be seen, our method will introduce some additional computational costs. However, this is because we simply repeat the LSS operation many times to generate the features of slices. Careful engineering optimization can significantly improve the efficiency.

Method	Backbone	NDS	FPS	Model Size(MB)	Image backbone(ms)	pooling(ms)	Fusion(ms)
SANet	R50	0.366	15.4	911.0	0.53	23.12	0.50
SANet	R101	0.379	14.3	1128.7	0.55	26.22	0.45
BEVDepth	R50	0.330	24.3	870.0	0.54	26.23	.
BEVDepth	R101	0.371	19.6	1087.1	0.55	26.26	.

branches transformer. As can be seen in Tab. 7. Using SE in the first stage and Trans in the second stage achieves the best performance compared with the alternatives. Nevertheless, all the fusion strategies can achieve considerable improvements compared with the baseline, demonstrating the effectiveness of the proposed SAN.

4.4. Computational Cost

In this section, we report the computational cost of SAN. As shown in Tab. 8, our method introduces additional computational and storage costs to the baseline methods. To be more specific, when the backbone is ResNet-101, our method introduces 41 MB storage cost and 27% slower than the BEVDepth baseline. The most time-consuming step is building the features of global and local slices. However, this is because our current implementation simply repeats the LSS [23] operations. More careful engineering optimization can help to reduce the computational cost of SAN, which will be our future work.

5. Limitation

Although the proposed SAN is simple yet effective, our method still has some limitations. One limitation is the additional computational and storage cost as mentioned above. However, we believe careful engineering optimization can solve this problem. Besides, our method follows the BEVDepth [15] pipeline, which is sensitive to the accuracy of depth values or the depth distributions. How to apply SAN to baseline methods such as BEVFormer [16] is still a problem, which will also be our future work.

6. Conclusion

In summary, we propose a novel method named Slice Attention Network for BEV 3D object detection in this paper. Instead of summing up the frustum features that fall into the same flattened BEV grid, our method explores the benefit of different heights in BEV space. We extract the BEV features of global and local slices. The global slices aim at covering the large height ranges, while the local slices aim at emphasizing informative local height ranges. To improve the performance, we propose to sample the local slices based on the histogram of LiDAR points along the height dimension. The features of local and global slices are fused by a two-stage strategy for task heads. We use BEVDepth as the baseline method and conduct detailed experiments to demonstrate the effectiveness of BEV-SAN.

References

[1] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9287–9296, 2019. 2

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 5

[3] Shaoyu Chen, Tianheng Cheng, Xinggang Wang, Wenming Meng, Qian Zhang, and Wenyu Liu. Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer. *arXiv preprint arXiv:2206.04584*, 2022. 1

[4] Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Chang Huang, and Wenyu Liu. Polar parametrization for vision-based surround-view 3d detection. *arXiv preprint arXiv:2206.10965*, 2022. 3

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 1

[6] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12093–12102, 2020. 2

[7] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[10] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 4, 5

[11] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 1, 2

[12] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 2, 5

[13] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *European Conference on Computer Vision*, pages 644–660. Springer, 2020. 2

[14] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo. *arXiv preprint arXiv:2209.10248*, 2022. 1, 3

[15] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv:2206.10092*, 2022. 1, 2, 3, 4, 5, 6, 8

[16] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 1, 2, 8

[17] 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. 1, 2

[18] 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. 1

[19] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 996–997, 2020. 2

[20] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3111–3121, 2021. 2

[21] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4721–4730, 2021. 2

[22] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. 2

[23] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pages 194–210. Springer, 2020. 2, 8

[24] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 1

[25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1

[27] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-

voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 1

[28] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *Conference on Robot Learning*, pages 1475–1485. PMLR, 2022. 2

[29] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021. 2

[30] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022. 1, 2

[31] Zengran Wang, Chen Min, Zheng Ge, Yinhao Li, Zeming Li, Hongyu Yang, and Di Huang. Sts: Surround-view temporal stereo for multi-view 3d detection. *arXiv preprint arXiv:2208.10145*, 2022. 3

[32] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. *arXiv preprint arXiv:2207.02202*, 2022. 1

[33] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11040–11048, 2020. 1

[34] Renrui Zhang, Han Qiu, Tai Wang, Ziyu Guo, Xuanzhuo Xu, Yu Qiao, Peng Gao, and Hongsheng Li. Monodetr: Depth-guided transformer for monocular 3d object detection. *arXiv preprint arXiv:2203.13310*, 2022. 2

[35] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3289–3298, 2021. 2

[36] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 6