JPV-Net: Joint Point-Voxel Representations for Accurate 3D Object Detection

Nan Song[1], **Tianyuan Jiang**[1], **Jian Yao**[1,2*]

[1]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, P.R. China
[2]AI Application and Innovation Research Center, The Open University of Guangdong, Guangzhou, Guangdong, P.R. China
{nanaoisong, tianyuan, jian.yao}@whu.edu.cn

Abstract

Voxel and point representations are widely applied in recent 3D object detection tasks from LiDAR point clouds. Voxel representations contribute to efficiently and rapidly locating objects, whereas point representations are capable of describing intra-object spatial relationship for detection refinement. In this work, we aim to exploit the strengths of both two representations, and present a novel two-stage detector, named Joint Point-Voxel Network (JPV-Net). Specifically, our framework is equipped with a Dual Encoders-Fusion Decoder, which consists of the dual encoders to extract voxel features of sketchy 3D scenes and point features rich in geometric context, respectively, and the Feature Propagation Fusion (FP-Fusion) decoder to attentively fuse them from coarse to fine. By making use of the advantages of these features, the refinement network can effectively eliminate false detection and achieve better accuracy. Besides, to further develop the perception characteristics of voxel CNN and point backbone, we design two novel intersection-over-union (IoU) estimation modules for proposal generation and refinement, both of which can alleviate the misalignment between the localization and the classification confidence. Extensive experiments on the KITTI dataset and the ONCE dataset demonstrate that our proposed JPV-Net outperforms other state-of-the-art methods with remarkable margins.

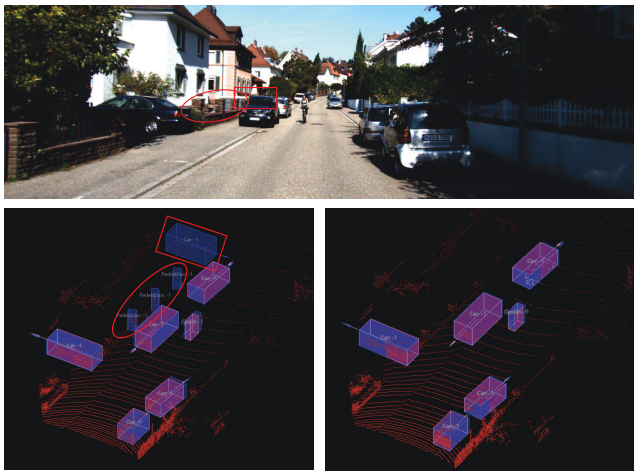

Figure 1: Illustration of detection results predicted by the detector without original point representations (bottom left) and our proposed JPV-Net (bottom right). Our JPV-Net with geometric context can eliminate the false detection marked in red, such as pillars and walls (detected as pedestrians and cars).

Introduction

Object detection is a challenging and meaningful problem in computer vision, since beneficial to many downstream vision tasks. With the rapid development of CNNs, tremendous success has been made in 2D object detection tasks. Recently, increasing attention is shifted to 3D object detection from point clouds, which is essential to many practical applications, such as autonomous driving vehicles, robotics and AR/VR. Compared with 2D images, raw point clouds are unordered, sparse and irregular, making it difficult to directly apply 2D detection methods to 3D scenes. Hence, it's crucial to explore an effective way to parse point clouds.

Existing LiDAR-based 3D object detection methods can be generally divided into two categories, i.e., voxel-based methods and point-based ones. By converting sparse point clouds into compact forms, like birds-eye-view (BEV) images (Yang, Liang, and Urtasun 2018; Engelcke et al. 2017) and voxel grids (Yan, Mao, and Li 2018; Lang et al. 2019), the voxel-based methods apply CNNs to regular 2D images or 3D voxels and efficiently extract features for detection. By contrast, the point-based methods (Qi et al. 2018; Yang et al. 2018) directly take raw point clouds as input and utilize PointNet and its variants (Qi et al. 2017a,b) to abstract point features. The former methods specialise in regressing high recall proposals, while the latter ones can avoid the information losses caused by voxelization and learn geometric features for better object parsing.

Instead of using just one specific representation, there are recent methods (He et al. 2020; Shi et al. 2020; Bhattacharyya and Czarnecki 2020) exploring to combine both of them. Despite significant breakthroughs, these methods still remain several problems restrictive to the representation capacity. Firstly, the point features derive from sketchy voxel features, which merely sketch 3D scenes and hardly compensate the lack of geometric information, leading to

*Corresponding author.

false detection shown in Figure 1. Besides, the simple concatenation of multi-scale point features is unfitted to reason about hierarchical spatial relationships and semantic context of point clouds.

Aiming at these problems above, we propose a novel two-stage 3D object detector, which takes advantages of both sketchy voxel features and geometric point features. Our critical design is the Dual Encoders-Fusion Decoder framework, composed of dual encoders and a feature fusion decoder. More specifically, voxel CNNs (Graham, Engelcke, and van der Maaten 2018; Yan, Mao, and Li 2018) and PointNet-based encoder (Qi et al. 2017b) are adopted to abstract voxel and point context, respectively, and meanwhile, the former efficiently generates accurate 3D proposals. Subsequently, the FP-Fusion modules adaptively integrate the point-wise features and corresponding voxel-to-point features in each level, and then, these fusion features can be propagated to original resolution through the decoder. Thus, as shown in Figure 1, our method effectively eliminates false detection and improves proposal refinement by incorporating the global location information and detailed intra-object context (such as tyres to vehicles and human pose to persons).

Moreover, to further develop the potentials of point and voxel representations, we propose two novel IoU modules analogous to IoUNet (Jiang et al. 2018), which alleviate the misalignment problem between localization and classification confidence. Different from images, special properties of 3D scenes like sparsity and more degrees-of-freedom increase the difficulty in providing accurate 3D IoU prediction. Considering that our framework processes objects with a different perspective on each stage, we present two indirect estimation approaches, i.e., the BEV-IoU-aware proposal generation and the point-IoU-aware refinement. On the proposal stage, BEV IoU is simpler and looser to estimate than 3D counterpart owing to the object regression in BEV view, and therefore, more proposals with precise localization can be retained. For refinement, we replace 3d IoU with grid point IoU, which not only involves the sparsity of point clouds, but improves the part comprehension of objects (Shi et al. 2019) due to our substituted grid point segmentation task.

Our key contributions are three-fold: (i) We propose a JPV-Net framework with Dual Encoders-Fusion Decoder to abstract point and voxel features, respectively, and efficiently integrate them from coarse to fine, bringing about improvements of 3D object detection; (ii) In terms of representations, we propose two novel IoU estimation modules for each stage, which effectively alleviate the misalignment and achieve high recall detection performance; (iii) We evaluate our method on KITTI (Geiger, Lenz, and Urtasun 2012) and ONCE (Mao et al. 2021) object detection benchmarks, and our proposed JPV-Net outperforms previous state-of-the-art methods with large margins.

Related Work

3D Object Detection with Voxel Grids. Relying on regular grids, voxel-based methods straightforwardly apply CNNs for efficient 3d detection. At first, VoxelNet (Zhou and Tuzel 2018) and SECOND (Yan, Mao, and Li 2018) employ voxelization and 3D CNNs to perform object detection. Some other works (Yang, Luo, and Urtasun 2018; Lang et al. 2019) adopt BEV maps or pillars instead and utilize 2D CNNs for real-time detection. Besides, Shi *et al.* (Shi et al. 2019; Deng et al. 2020) introduce two-stage detection frameworks, further improving the performance of voxel-based methods. Although constrained by the losses of geometric information, these voxel-based methods are suitable to generate accurate 3D proposals with a high recall.

3D Object Detection with Raw Point Clouds. Point-based methods take raw point clouds as input and generally abstract point features with PointNet-like frameworks (Qi et al. 2017b). PointRCNN (Shi, Wang, and Li 2019) proposes a complete PointNet-based two stage detection framework, and sets a precedent for follow-up methods (Yang et al. 2019, 2020). There are some methods (Shi and Rajkumar 2020; Zhang, Huang, and Wang 2020) better modeling point relations with GNN. Powered by PointNet and its variants, these methods enrich features with geometric information and are superior to voxel-based ones.

3D Object Detection with Point-Voxel Representations. Recently, some methods strive to incorporate the advantages of point and voxel representations. SASSD (He et al. 2020) introduces an auxiliary network without extra cost, which supervises voxel context to focus on the intra-object structure. Similarly, the novel two-stage framework PV-RCNN (Shi et al. 2020) abstracts a set of keypoint features by the Voxel Set Abstraction (VSA) module and explicitly makes use of them for refinement. Our proposed JPV-Net preserves original point features rich in geometric context, and fuses them with voxel features from coarse to fine, more effective than existing methods.

Study on the Misalignment between Classification Confidence and Localization. In the detection pipeline, classification and localization are solved differently, which results in the misalignment between classification confidence and localization. IoUNet (Jiang et al. 2018) first analyzes and mitigates the aforementioned issue by means of a proposed IoU prediction branch and an IoU-aware NMS. In 3D scenes, Zhou *et al.* (Zhou et al. 2019) introduce into this methodology and improve 3D IoU and related IoU loss. In this work, our proposed BEV IoU-aware proposal generation and point-IoU-aware refinement explore the characteristics and exploit the advantages of voxel- and point-based 3D perceptions.

Joint Point-Voxel Network

In this section, we present a novel two-stage detection framework, JPV-Net, to integrate the sketchy voxel representations and geometric point representations for 3D object detection. The whole framework of JPV-Net, illustrated in Figure 2, consists of a voxel branch for efficient proposal generation and the Dual Encoders-Fusion Decoder for feature extraction and fusion. Moreover, our proposed two IoU modules exploit the potentials of BEV and point perceptions, rectifying the misalignment and promoting better localization performance.

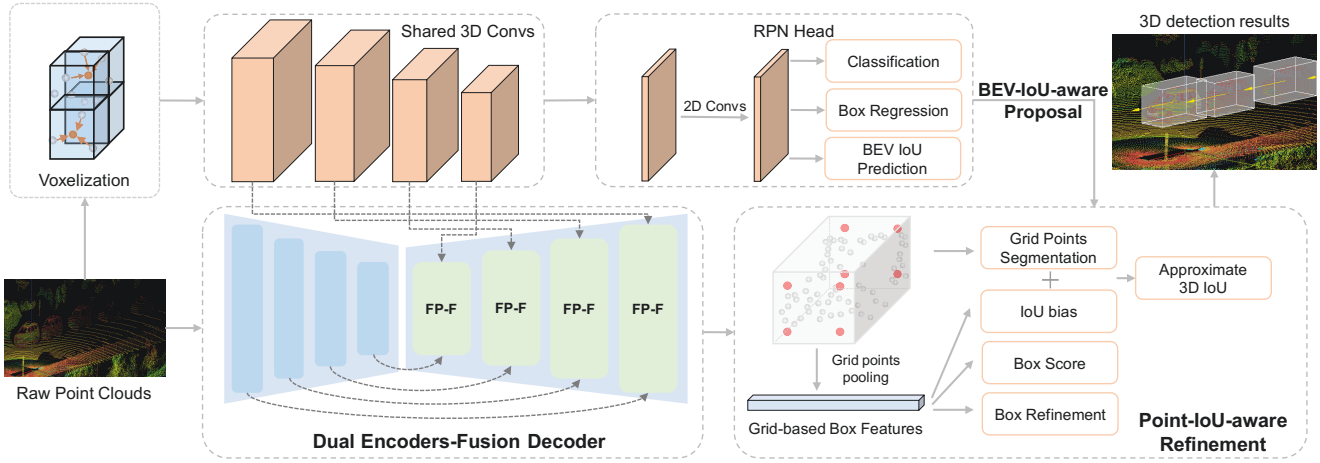

Figure 2: Illustration of the proposed JPV-Net detector. Our network is composed of two parts: the voxel network to encode multi-scale sketchy features and efficiently generate 3D proposals, and the Dual Encoders-Fusion Decoder to learn different representations, conduct feature fusion and perform proposal refinement with these representative features. Besides, two novel IoU-aware modules are applied for proposal generation and refinement, respectively.

Voxel CNNs for Proposal Generation

Following the previous works (Deng et al. 2020; Shi et al. 2020), we introduce commonly used SECOND-like voxel CNNs as the voxel backbone. Bearing in mind that the precise coordinate information is not necessary for rough localization, we take the voxel center coordinates $\{\mathbf{v}_i = (x_i, y_i, z_i)\}_{i=1}^{N}$ as original input features for simplicity. The network stacks four 3D sparse convolutional blocks to encode feature volumes, each of which consists of a series of $3\times3\times3$ sub-manifold convolutions and a sparse convolution with downsampled size of 2 (excluding the last block). Subsequently, the 3D tensor are reshaped into BEV representations in accord with autonomous driving tasks and further encoded by a 2D Region Proposal Network (RPN). Besides, anchor-based detection is adopted to achieve a high recall.

Dual Encoders-Fusion Decoder

To make up the information losses caused by voxelization, we present the Dual Encoders-Fusion Decoder, which is composed of voxel encoder shared with proposal branch, PointNet-like encoder and FP-Fusion Decoder for feature fusion. Given the fusion features of an overall 3D scene, 3D proposals are precisely refined following the grid-based approaches.

PointNet++ Backbone. To preserve significant geometric context, the point features are extracted from raw point clouds by our point backbone instead of sketchy voxel features. Specifically, we adopts four SA modules with $1\times, 4\times, 8\times$, and $16\times$ downsampled scales, consistent with the hierarchical structure of the voxel CNN. Then, multi-scale point and voxel features are aggregated via the FP-Fusion Decoder.

Trilinear SA and FP-Fusion Decoder. The sketchy voxel features and geometric point features are learned from dual encoders relying on different perceptive modes and encoding schemes. To adaptively evaluate the importance of these perceptions and conduct fusion, we propose the Trilinear SA module to project voxel features onto neighboring points and the FP-Fusion module to attentively fuse point and voxel features.

Generally, the nearest neighbor interpolation and the SA module adopted in PointNet++ (Qi et al. 2017b) are commonly used for feature projection between point sets. Existing point-voxel methods adopt and modify these modules, which serve as the efficient connection from voxel grids to points. These schemes regard the voxels as a set of points, however, ignoring the inherent spatial relationship and regular representation of convolutions. Motivated by the bilinear interpolation method, we propose the Trilinear SA module to aggregate point-wise voxel features for each point from its sparse eight neighboring voxels, briefly illustrated in Figure 3. Specifically, we denote $\{(\mathbf{v}_i^{(l)}, \mathbf{f}_i^{(l)})\}_{i=1}^{N}$ as the voxel-wise coordinates and features in the l-th level of voxel CNNs. Given each point $\mathbf{p}^{(l)}$ in the same level, we firstly sample the corresponding neighboring set $\mathcal{S}$ containing no more than eight voxels as:

$$\mathcal{S}(\mathbf{p}^{(l)}) = \left\{ [\mathbf{v}_k^{(l)} - \mathbf{v}_{\mathbf{p}}^{(l)}; \mathbf{w}_k^{(l)} \mathbf{f}_k^{(l)}] \;\middle|\; \mathbf{v}_k^{(l)} \in \mathcal{N}(\mathbf{p}^{(l)}) \right\}, \tag{1}$$

where $\mathbf{w}_k^{(l)}$ denotes the interpolation weight of $\mathbf{v}_k^{(l)}$, and $\mathbf{v}_{\mathbf{p}}^{(l)}$ and $\mathcal{N}(\mathbf{p}^{(l)})$ denote the voxel coordinate and a set of sparse neighboring voxels for the point $\mathbf{p}^{(l)}$. We take the relative voxel coordinates and voxel feature vectors as the representations of neighboring sets. In consideration of the sparse distribution of the voxel space, we utilize the max pooling operation to aggregate features instead of weighted sum, which can be formularized as follows:

$$\mathbf{F}_{\mathrm{v}}^{(l)} = \max(\mathcal{F}(\mathcal{S}(\mathbf{p}^{(l)}))), \tag{2}$$

where $\mathbf{F}_{\mathrm{v}}^{(l)}$ denotes the projected voxel feature for the point $\mathbf{p}^{(l)}$, and $\mathcal{F}$ denotes a Multi-Layer Perceptron (MLP) network to encode the set features. Our proposed Trilinear SA

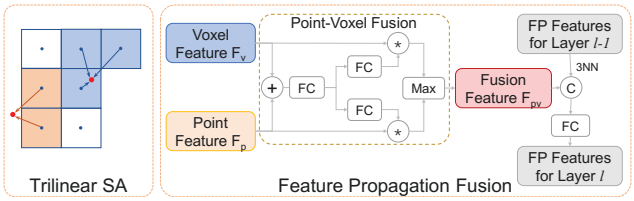

Figure 3: Illustration of our Trilinear SA module (left) and FP-Fusion module (right). The Trilinear SA module briefly aggregates non-empty voxel features of sparse eight neighbors for each point. The FP-Fusion module attentively fuses point and voxel features and passes it through the Feature Propagation decoder.

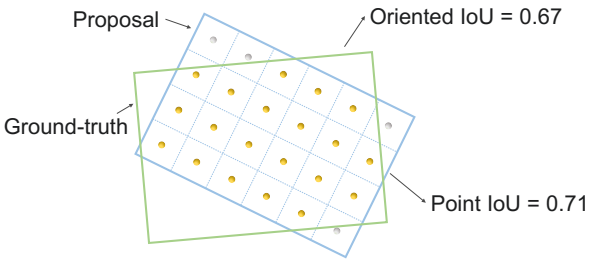

Figure 4: Illustration of an example of our point-IoU-aware refinement on the BEV map. The proposal is uniformly divided into 24 grid points, and the real IoU score and the point IoU score are 0.67 and 0.71, respectively.

module is lightweight to economize computing resource and preserves more local voxel context, beneficial to the interaction between point and voxel encoders.

Notably, the voxel and point branches summarize the 3D scene with a different emphasis. The voxel CNN roughly sketches the entire scene and objects regardless of detailed structure, while the PointNet backbone represents more discriminative details. Therefore, adaptively fusing multiple features is vital for fine-grained refinement. Given the multi-scale point features and point-wise voxel features, we introduce an extra fusion module to attentively fuse them. As shown in Figure 3, we directly add the point features $\mathbf{F}_{\mathrm{p}}^{(l)}$ and voxel features $\mathbf{F}_{\mathrm{v}}^{(l)}$ in the same l-th level, and feed it into several MLP layers to generate the normalized weight maps $\mathbf{w}_{\mathrm{p}}$ and $\mathbf{w}_{\mathrm{v}}$ with the same channel as input. Then, for simplicity, we use a max pooling operator to fuse them, each of which is weighted by channel-wise production with its weight map as:

$$\mathbf{w}_{\mathrm{p}} = \sigma(\mathcal{W}_{\mathrm{p}} \tanh(\mathcal{W}(\mathbf{F}_{\mathrm{p}}^{(l)} + \mathbf{F}_{\mathrm{v}}^{(l)}))), \tag{3}$$

$$\mathbf{w}_{\mathrm{v}} = \sigma(\mathcal{W}_{\mathrm{v}} \tanh(\mathcal{W}(\mathbf{F}_{\mathrm{p}}^{(l)} + \mathbf{F}_{\mathrm{v}}^{(l)}))), \tag{4}$$

$$\mathbf{F}_{\mathrm{pv}}^{(l)} = \max(\mathbf{w}_{\mathrm{p}} \odot \mathbf{F}_{\mathrm{p}}^{(l)},\ \mathbf{w}_{\mathrm{v}} \odot \mathbf{F}_{\mathrm{v}}^{(l)}), \tag{5}$$

where $\mathcal{W}$ is the shared MLP, $\mathcal{W}_{\mathrm{p}}$ and $\mathcal{W}_{\mathrm{v}}$ are the independent MLPs for point and voxel features, and the tanh and sigmoid activation functions are used to normalize the weight maps. Then we pass the fusion features through several Feature Propagation modules to reconstruct the original scene. Hence, the entire scene can be thoroughly encoded by the semantic fusion features, that merge global spatial perception and local geometric information. Profiting from these modules, our framework not only reduces the computation cost and maintains more representative points than PV-RCNN (Shi et al. 2020), but encodes more comprehensive point-voxel features than SA-SSD (He et al. 2020).

To handle the ambiguous boxes, we adopt the grid-based approaches for part-aware refinement following previous works (Shi et al. 2019, 2020). Besides, we boost the local sensitivity by our proposed point-IoU-aware refinement and achieve more precise localization.

IoU-aware Proposal Generation & Refinement

The misalignment between classification confidence and localization is a universal phenomenon in object detection tasks. Motivated by the methodology proposed by IoUNet (Jiang et al. 2018), we explore the characteristic for each stage in our 3D detector, i.e., proposal generation from BEV maps on the first stage and grid-based refinement on the second stage, and propose two novel approaches to handle these issues.

BEV-IoU-aware Proposal Generation. On the proposal stage, 3D bounding boxes are generated from BEV feature maps, where the details of height dimension lose partly. Thus, we utilize BEV IoU for proposal generation, which is more appropriate than its 3D counterpart. Similar to IoUNet, we introduce an additional IoU branch to estimate BEV IoU between the predictions and ground-truth bounding boxes. During the NMS procedure, the mixed scores are used for ranking, which can be written as:

$$S = S_{\mathrm{cls}} \times S_{\mathrm{IoU}}^{\mathrm{BEV}} + S_{\mathrm{IoU}}^{\mathrm{BEV}}, \tag{6}$$

where S_{cls} and $S_{\mathrm{IoU}}^{\mathrm{BEV}}$ denotes the classification confidence and the estimated BEV IoU score for predicted boxes. Compared to the production of two values, our designed score is non-parametric and more effective to cope with the situation where boxes with high IoU scores possess low even zero classification confidence due to the unbalanced number of positive/negative anchors, and hence, our proposal network can preserve more valid boxes and achieve a high recall performance. Although simple, experimental results in Section 4.3 demonstrate the effectiveness of our BEV-IoU-aware proposal generation strategy.

point-IoU-aware Refinement. To tackle the misalignment on the refinement stage, an intuitive solution is to predict the 3D IoU score between refined boxes and ground-truth boxes. However, dynamic changes of IoU supervision labels and sparsity of point clouds increase the difficulty directly estimating 3D IoU scores. Noting that grid-based methods are adopted for part-sensitive refinement, grid points are uniformly sampled in each proposal and contain object features. That is to say, the overlap of boxes could be approximated by the number of grid points positioned in the overlap if there were massive points according to the Monte Carlo simulation. Based on this finding, we strive to convert the task of

IoU estimation to semantic segmentation for grid points. As illustrated in Figure 4, our proposed point IoU score is approximate to the real IoU score in the example. Given the grid points of each proposal, semantic labels are generated by checking whether each point in refined boxes is inside or outside of a ground-truth box. Subsequently, we approximately regard the number of positive semantic points as the intersection of boxes. Besides, the rectification for most proposals is normally slight, and therefore we regard that the number of grid points in ground-truth boxes is generally identical to corresponding proposals. Above all, an approximate point IoU score can be formularized as:

$$S^{\mathrm{p}}_{\mathrm{IoU}} = \frac{N_{\mathrm{pos}}}{2 \times N - N_{\mathrm{pos}}}, \tag{7}$$

where N and N_{pos} denote the numbers of grid points and positive points, respectively. To smooth the gap between the 3D IoU and our approximate point IoU, we replace the number of positive points with the sum of segmentation scores and introduce an extra bias prediction branch. Hence, the aforementioned formula is addressed as:

$$S^{\mathrm{p}}_{\mathrm{IoU}} = \frac{\sum_{i=1}^{N} c_i}{2 \times N - \sum_{i=1}^{N} c_i} + S_{\mathrm{bias}}, \tag{8}$$

where c_i is the segmentation score for each grid point and S_{bias} is the predicted tiny difference of our point IoU from real 3D IoU. Our point IoU branch is then trained to minimize the cross-entropy loss for segmentation and smooth-L1 loss for bias as:

$$L^{\mathrm{p}}_{\mathrm{IoU}} = L_{\mathrm{ce}} + \alpha L_s(\widehat{S}_{\mathrm{bias}},\ S_{\mathrm{bias}}), \tag{9}$$

where the loss weight $\alpha = 0.5$ in experiments. Our grid point segmentation strengthens the part perception for objects, and the proposed point-IoU-aware refinement outperforms the 3D IoU prediction, all which contribute to accurate refinement significantly.

Loss Functions

Our JPV-Net framework is trainable end-to-end and overall loss function consists of the region proposal loss and the refinement loss. For the proposal generation, we follow SECOND to design the region proposal loss L_{rpn}, composed of focal loss (Lin et al. 2017) with default hyper-parameters for classification, smooth-L1 loss for both box regression and our BEV IoU estimation, and cross-entropy loss for bins of direction as:

$$L_{\mathrm{rpn}} = L^{\mathrm{p}}_{\mathrm{cls}} + \beta L^{\mathrm{p}}_{\mathrm{reg}} + \gamma L^{\mathrm{BEV}}_{\mathrm{IoU}}, \tag{10}$$

$$L^{\mathrm{p}}_{\mathrm{reg}} = \sum_{r \in x,y,z,l,h,w,\theta} L_s(\Delta\hat{r},\ \Delta r) + \mu L_{\mathrm{ce}}(\hat{b}_\theta, b_\theta), \tag{11}$$

where $\Delta\hat{r}$ and $\hat{b}_\theta$ denote the predicted box residual and bins of direction, and $L^{\mathrm{p}}_{\mathrm{cls}}$, $L^{\mathrm{p}}_{\mathrm{reg}}$ and $L^{\mathrm{BEV}}_{\mathrm{IoU}}$ denote losses of classification, regression and IoU estimation for proposal, respectively. The refinement loss L_{rcnn} includes the point segmentation loss L_{seg}, the IoU-related confidence loss $L^{\mathrm{r}}_{\mathrm{cls}}$, the regression loss $L^{\mathrm{r}}_{\mathrm{reg}}$ similar to proposal and our point-IoU-aware refinement loss $L^{\mathrm{p}}_{\mathrm{IoU}}$,

$$L_{\mathrm{rcnn}} = L^{\mathrm{r}}_{\mathrm{cls}} + \beta' L^{\mathrm{r}}_{\mathrm{reg}} + \gamma' L^{\mathrm{p}}_{\mathrm{IoU}} + L_{\mathrm{seg}}. \tag{12}$$

Hence, the overall loss is calculated as the sum of these losses for two stages. Further training loss details are provided in the supplementary file.

Experiments

In this section, we evaluate our proposed JPV-Net detector on the challenging 3D object detection benchmark of KITTI dataset. In the following, we briefly introduce the dataset and the implementation details of our framework in Section 4.1. Then we illustrate the experimental results by comparing with state-of-the-art 3D detection methods on the KITTI dataset in Section 4.2. Finally, we present extensive ablation studies to analyze and validate our design in Section 4.3. Furthermore, we also evaluate our method on the larger and more diverse ONCE dataset, and experimental results are illustrated in the supplementary file.

Implementation Details

Dataset. KITTI Dataset (Geiger, Lenz, and Urtasun 2012) is a widely used benchmark dataset for autonomous driving. There are 7,481 training samples and 7,518 test samples with three categories of Car, Pedestrian and Cyclist. The training samples are generally divided into the $train$ split (3,712 samples) and the val split (3,769 samples). We use average precision (AP) as the metric to evaluate for all the three difficulty levels (Easy, Moderate and Hard). Remarkably, our models are trained on the $train$ split and evaluated on the val split for validation, while trained with all training samples and evaluated with test samples for test.

Network Architecture. For the 3D scenes in the KITTI dataset, the detection range is within $[0, 70.4]m$ for the X-axis, $[-40, 40]m$ for the Y-axis and $[-3, 1]m$ for the Z-axis, containing about $20K$ LiDAR points. We voxelize each scene with the voxel size $(0.05m, 0.05m, 0.1m)$ as the voxel input, and subsample 16,384 points from raw point clouds as the point input. As shown in Figure 2, our voxel CNN is composed of four 3D encoding levels and 2D convolutions for BEV maps, same as the SECOND (Yan, Mao, and Li 2018). For consistency, we also employ four levels of Set Abstraction modules to encode point features with the point sizes of 16384, 4096, 2048, 1024, respectively. Then, voxel and point features in the same level are passed to the corresponding FP-Fusion module through skip links. We adopt four FP-Fusion layers to gradually summarize the whole point scene and finally recover the original size. To achieve better refinement performance, we introduce the point-grid-based refinement proposed by PV-RCNN with its official hyper-parameters.

Training and Inference Schemes. Our JPV-Net framework is end-to-end trainable by the ADAM optimizer with an initial learning rate and a weight decay of 0.01, and the batchsize 16 on 8 GTX 1080 Ti GPUs. We train our network for 80 epochs with the cosine annealing strategy for the learning rate decay. In training, the IoU thresholds for positive and negative anchors are set to 0.6 and 0.45, respectively. The matching IoU for proposal is calculated by the horizontal rectangles in BEV maps. On the refinement

Types	Methods	Modalities	Car - 3D			Cyclist - 3D		
			Easy	Mod.	Hard	Easy	Mod.	Hard
1-stage	SECOND (Yan, Mao, and Li 2018)	LiDAR	83.34	72.55	65.82	71.33	52.08	45.83
	PointPillars (Lang et al. 2019)		82.58	74.31	68.99	77.10	58.65	51.92
	SA-SSD (He et al. 2020)		88.75	79.79	74.16	-	-	-
	Point-GNN (Shi and Rajkumar 2020)		88.33	79.47	72.29	78.60	63.48	57.08
	CIA-SSD (Wu et al. 2021)		89.59	80.28	72.87	-	-	-
2-stage	MV3D (Chen et al. 2017)	RGB + LiDAR	74.97	63.63	54.00	-	-	-
	AVOD-FPN (Ku et al. 2018)		83.07	71.76	65.73	63.76	50.55	44.93
	F-PointNet (Qi et al. 2018)		82.19	69.79	60.59	72.27	56.12	49.01
	PointPainting (Vora et al. 2020)		82.11	71.70	67.08	77.63	63.78	55.89
	3D-CVF (Yoo et al. 2020)		89.20	80.05	73.11	-	-	-
	EPNet (Huang et al. 2020)		89.81	79.28	74.59	-	-	-
	PointRCNN (Shi, Wang, and Li 2019)	LiDAR	86.96	75.64	70.70	74.96	58.82	52.53
	Fast Point R-CNN (Chen et al. 2019)		85.29	77.40	70.24	-	-	-
	STD (Yang et al. 2019)		87.95	79.71	75.09	78.69	61.59	55.30
	Part-A^2 Net (Shi et al. 2019)		87.81	78.49	73.51	79.17	63.52	56.93
	PV-RCNN (Shi et al. 2020)		**90.25**	81.43	76.82	78.60	63.71	57.65
	JPV-Net (Ours)		88.66	**81.73**	**76.94**	**80.66**	**65.41**	**59.26**

Table 1: Comparisons with state-of-the-art methods on the KITTI $test$ set. All results are evaluated by the mAP with 40 recall positions.

Methods	Easy	Mod.	Hard
SECOND (Yan, Mao, and Li 2018)	88.12	78.30	76.91
SA-SSD (He et al. 2020)	**90.15**	79.91	78.78
3DSSD (Yang et al. 2020)	89.71	79.45	78.67
3D-CVF (Yoo et al. 2020)	89.67	79.88	78.47
Part-A^2 Net (Shi et al. 2019)	89.47	79.47	78.54
PV-RCNN (Shi et al. 2020)	88.93	83.36	78.70
JPV-Net (Ours)	89.71	**84.61**	**79.09**

Table 2: Comparisons with state-of-the-art methods on the KITTI val set for the Car class. All results are evaluated by the mAP with 11 recall positions.

Methods	DE-FD	BEV IoU	Point IoU	Easy	Mod.	Hard
SECOND				90.37	81.78	78.60
PV-RCNN				91.74	84.07	82.06
(a)	✓			92.05	84.91	82.61
(b)	✓	✓		92.11	85.21	82.88
(c)	✓		✓	92.37	85.29	83.04
JPV-Net	✓	✓	✓	**92.45**	**85.60**	**83.12**

Table 3: Ablation experiments on the val split of the KITTI dataset. DE-FD stands for Dual Encoders-Fusion Decoder, and $(a), (b), (c)$ are variants of our JPV-Net.

stage, we randomly sample 128 proposals with an equal ratio of the positive and negative objects. For inference, we only keep the top-100 proposals for further refinement. After the refinement, redundant boxes are removed with a NMS threshold of 0.1.

Data Augmentation. Widely adopted data augmentation strategies are conducted in our framework following (Yan, Mao, and Li 2018; Lang et al. 2019; Shi et al. 2020). Specifically, we utilize the random flipping along the X-axis, the global scaling with a random scaling factor sampled from $[0.95, 1.05]$, and the global rotation around the Z-axis with a random angle sampled from $[-\frac{\pi}{4}, \frac{\pi}{4}]$. Besides, the ground-truth sampling strategy proposed by (Yan, Mao, and Li 2018) is also introduced to substantially increase the quantity of 3D objects.

Experiments on the KITTI Dataset

In this section, our JPV-Net is trained on the $train$+val set and evaluated by submitting the detection results to the KITTI 3D object detection benchmark. The performance on the $test$ set is calculated with 40 recall positions by the test server. Moreover, we validate our model on the val set, which is calculated with 11 recall positions to fairly compare with the previous works.

As shown in Table 1, the proposed JPV-Net surpasses other state-of-the-art object detectors in the 3D detection tasks with a remarkable margin. In particular, our method achieves 2.06%, 1.70%, 1.61% gains on three difficulty levels of the Cyclist class over PV-RCNN (Shi et al. 2020). For the Car class, our method provides better performance on the Moderate and Hard difficulty levels, while dropping a little on the Easy difficulty level.

As shown in Table 2, we also compare the performance of our JPV-Net on the val split of the Car class with some novel methods. Our method achieves better AP by 0.78%, 1.25%, 0.39% compared to PV-RCNN on three difficulty levels. Besides, we present several qualitative results on the KITTI dataset in Figure 5.

Ablation Studies

Here, we conduct extensive experiments to analyze the effectiveness of individual modules in our detector. All models are trained on the $train$ split and evaluated on the val split for the Car class with 40 recall positions.

Effects of Joint Point-Voxel Features and Dual Encoders-Fusion Decoder. Profiting from the joint point-voxel features, our proposed FP-Fusion module brings significant improvement of 3.13% over SECOND (Yan, Mao, and Li 2018) and 0.84% over PV-RCNN (Shi et al. 2020) shown in the first two rows of Table 3. To validate the effectiveness of

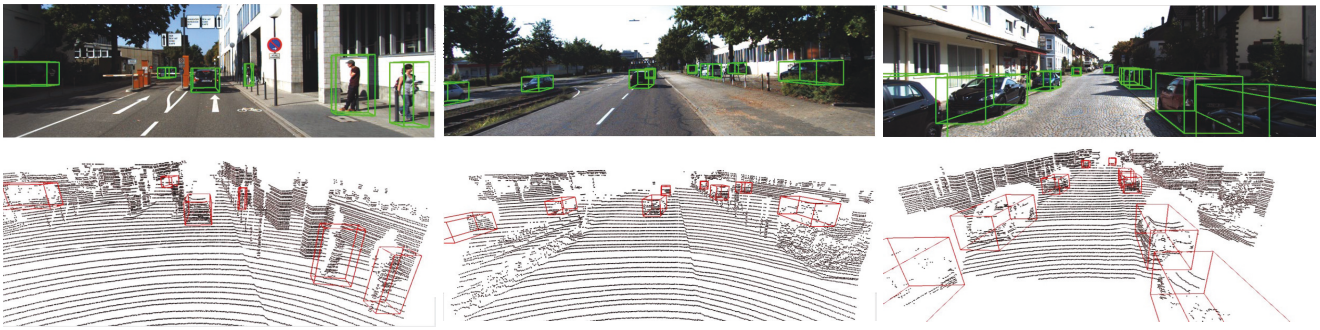

Figure 5: Illustration of qualitative results on the KITTI test set. The two rows show the camera images and corresponding front view of point clouds. The predicted bounding boxes are shown in red and green colors, respectively.

Methods	Voxel Features	Point Features	Easy	Mod.	Hard
Voxel Network	✓		91.82	84.48	82.23
Point Network		✓	91.84	84.61	82.45
DE-FD (con.)	✓	✓	91.98	84.82	82.55
DE-FD (att.)	✓	✓	**92.05**	**84.91**	**82.61**

Table 4: Effects of different representations for detection performance.

V2P Projection	Easy	Mod.	Hard
3NN	91.88	84.83	82.49
VSA	92.01	**84.94**	**82.63**
Trilinear SA	**92.05**	84.91	82.61

Table 5: Comparison between our Trilinear SA and other projection methods.

aggregating both two representations, we introduce two simplified networks with different refinement features. Specifically, the voxel network removes the point encoder, approximately regarded as the PV-RCNN equipped with a Feature Propagation decoder, whereas the point network abandons the projection from voxel grids to points, which adopts two separate branches for proposal generation and refinement. As shown in Table 4, our network of joint point-voxel features outperforms both of the simplified networks. It is worth noting that the point network performs better than the voxel network, because point features abstracted by PointNet++ can provide more fine-grained geometric information than those from sketchy voxel features. Besides, we compare our attentive fusion module with the concatenation operation, and experimental results show that our method is more effective. It is therefore that the fusion of voxel and point representations contributes remarkably to the detection performance of our framework.

Effects of the Trilinear SA Module. We investigate the effects of the Trilinear SA module by comparing with the Three Nearest Neighbor (3NN) interpolation and the Voxel Set Abstraction (VSA) module. As shown in Table 5, our Trilinear SA module achieves similar performance to the VSA module due to the fine-grained geometric context of raw point clouds, and is much superior to the 3NN interpolation. Free of the multi-scale grouping operation and limiting the number of neighbors, our method saves more memory than the VSA module.

	IoU Methods	Easy	Mod.	Hard	Recall
Proposal	without IoU	90.37	81.78	78.60	66.17
	3D IoU	**90.59**	81.87	78.74	66.77
	BEV IoU	90.44	**81.92**	**78.87**	**67.04**
Refinement	without IoU	92.05	84.91	82.61	75.15
	3D IoU	**92.42**	85.15	**83.08**	75.41
	Point IoU	92.37	**85.29**	83.04	**75.96**

Table 6: Effects of the BEV IoU-aware proposal generation and the point-IoU-aware refinement.

Effects of the BEV IoU-aware Proposal Generation and point-IoU-aware Refinement. As shown in the forth and fifth rows in Table 3, each of our proposed IoU methods can further improve the detection performance, and the combination of them leads to the substantial enhancement by 0.69%. We compare the mAP and recall among different IoU prediction methods in Table 6. Specifically, we evaluate the BEV and point IoU methods based on proposal stage and final results, respectively. All our proposed methods can partly alleviate the misalignment and improve the detection performance and recall. In details, the BEV IoU method outperforms others on most metrics, and the point IoU method is mainly effective on the Moderate difficulty and recall.

Conclusion

In this work, we have presented a novel two-stage 3D object detection framework named as JPV-Net, which incorporates the advantages of voel and point representations. Voxel features extracted by 3D CNNs sketch the entire 3D scenes, and geometric point features capture spatial information to fill in the details. Accurate proposal refinement can be performed benefiting from representative fusion features. Our two IoU-aware approaches are valid to alleviate the misalignment, leading to further promotion. Experimental results on the KITTI and ONCE datasets demonstrate that our proposed modules compensate the weaknesses of existing works and bring significant improvement of detection performance.

Acknowledgements

This work was partially supported by the Shenzhen Central Guiding the Local Science and Technology Development Program (No. 2021Szvup100).

References

Bhattacharyya, P.; and Czarnecki, K. 2020. Deformable PV-RCNN: Improving 3D Object Detection with Learned Deformations. *arXiv preprint arXiv:2008.08766*.

Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-View 3D Object Detection Network for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1907–1915.

Chen, Y.; Liu, S.; Shen, X.; and Jia, J. 2019. Fast Point R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 9775–9784.

Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2020. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. *arXiv:2012.15712*.

Engelcke, M.; Rao, D.; Wang, D. Z.; Tong, C. H.; and Posner, I. 2017. Vote3Deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1355–1361.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are We Ready for Autonomous Driving? the KITTI Vision Benchmark Suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3354–3361.

Graham, B.; Engelcke, M.; and van der Maaten, L. 2018. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9224–9232.

He, C.; Zeng, H.; Huang, J.; Hua, X.-S.; and Zhang, L. 2020. Structure Aware Single-stage 3D Object Detection from Point Cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11873–11882.

Huang, T.; Liu, Z.; Chen, X.; and Bai, X. 2020. EPNet: Enhancing Point Features with Image Semantics for 3D Object Detection. *arXiv preprint arXiv:2007.08856*.

Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; and Jiang, Y. 2018. Acquisition of Localization Confidence for Accurate Object Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 784–799.

Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; and Waslander, S. L. 2018. Joint 3D Proposal Generation and Object Detection from View Aggregation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–8.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12697–12705.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Mao, J.; Niu, M.; Jiang, C.; Liang, X.; Li, Y.; Ye, C.; Zhang, W.; Li, Z.; Yu, J.; Xu, C.; et al. 2021. One Million Scenes for Autonomous Driving: ONCE Dataset. *arXiv preprint arXiv:2106.11037*.

Qi, C. R.; Liu, W.; Wu, C.; Su, H.; and Guibas, L. J. 2018. Frustum PointNets for 3D Object Detection from RGB-D Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 918–927.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*, 5099–5108.

Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10529–10538.

Shi, S.; Wang, X.; and Li, H. 2019. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–779.

Shi, S.; Wang, Z.; Wang, X.; and Li, H. 2019. Part-A^2 Net: 3D Part-Aware and Aggregation Neural Network for Object Detection from Point Cloud. *arXiv preprint arXiv:1907.03670*.

Shi, W.; and Rajkumar, R. 2020. Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1711–1719.

Vora, S.; Lang, A. H.; Helou, B.; and Beijbom, O. 2020. PointPainting: Sequential Fusion for 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4604–4612.

Wu, Z.; Weiliang, T.; Sijin, C.; Li, J.; and Chi-Wing, F. 2021. CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud. In *AAAI*.

Yan, Y.; Mao, Y.; and Li, B. 2018. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10): 3337.

Yang, B.; Liang, M.; and Urtasun, R. 2018. HDNET: Exploiting HD Maps for 3D Object Detection. In *Conference on Robot Learning (CoRL)*, 146–155.

Yang, B.; Luo, W.; and Urtasun, R. 2018. PIXOR: Real-time 3D Object Detection from Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7652–7660.

Yang, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020. 3DSSD: Point-based 3D Single Stage Object Detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11040–11048.

Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; and Jia, J. 2018. IPOD: Intensive Point-based Object Detector for Point Cloud. *arXiv preprint arXiv:1812.05276*.

Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; and Jia, J. 2019. STD: Sparse-to-Dense 3D Object Detector for Point Cloud. In *Proceedings of the IEEE International Conference on Computer Vision*, 1951–1960.

Yoo, J. H.; Kim, Y.; Kim, J. S.; and Choi, J. W. 2020. 3D-CVF: Generating Joint Camera and LiDAR Features Using Cross-View Spatial Feature Fusion for 3D Object Detection. *arXiv preprint arXiv:2004.12636*.

Zhang, Y.; Huang, D.; and Wang, Y. 2020. PC-RGNN: Point Cloud Completion and Graph Neural Network for 3D Object Detection. *arXiv preprint arXiv:2012.10412*.

Zhou, D.; Fang, J.; Song, X.; Guan, C.; Yin, J.; Dai, Y.; and Yang, R. 2019. IoU Loss for 2D/3D Object Detection. In *International Conference on 3D Vision (3DV)*, 85–94.

Zhou, Y.; and Tuzel, O. 2018. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4490–4499.