mathematics

MDPI

Article

Point-Rich: Enriching Sparse Light Detection and Ranging Point Clouds for Accurate Three-Dimensional Object Detection

Yanchao Zhang [1], Yinuo Zheng [2], Dingkun Zhu [3,*], Qiaoyun Wu [4], Hansheng Zeng [1], Lipeng Gu [2] and Xiangping Bryce Zhai [2]

[1] Special Equipment Safety Supervision Inspection Institute of Jiangsu Province, Nanjing 210036, China; yanchaozhang_l@hotmail.com (Y.Z.); hansheng_zeng@hotmail.com (H.Z.)
[2] School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210001, China; zhyinuo@nuaa.edu.cn (Y.Z.); gulp1224@nuaa.edu.cn (L.G.); blueicezhaixp@nuaa.edu.cn (X.B.Z.)
[3] School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China
[4] School of Artificial Intelligence, Anhui University, Hefei 230601, China; wuqiaoyun@ahu.edu.cn
* Correspondence: zhudingkun@jsut.edu.cn

Abstract: LiDAR point clouds often suffer from sparsity and uneven distributions in outdoor scenes, leading to the poor performance of cutting-edge 3D object detectors. In this paper, we propose Point-Rich, which is designed to improve the performance of 3D object detection. Point-Rich consists of two key modules: HighDensity and HighLight. The HighDensity module addresses the issue of density imbalance by enhancing the point cloud density. The HighLight module leverages image semantic features to enrich the point clouds. Importantly, Point-Rich imposes no restrictions on the 3D object detection architecture and remains unaffected by feature or depth blur. The experimental results show that compared with the Pointpillars on the KITTI dataset, the mAP of Point-Rich under the bird's eyes view improves by 5.53% on average.

Keywords: Point-Rich; 3D object detection; lightweight network

MSC: 68-04

check for updates

Citation: Zhang, Y.; Zheng, Y.; Zhu, D.; Wu, Q.; Zeng, H.; Gu, L.; Zhai, X.B. Point-Rich: Enriching Sparse Light Detection and Ranging Point Clouds for Accurate Three-Dimensional Object Detection. *Mathematics* **2023**, *11*, 4809. https://doi.org/10.3390/math11234809

Academic Editors: Manning Wang and Konstantin Kozlov

Received: 24 October 2023
Revised: 13 November 2023
Accepted: 20 November 2023
Published: 28 November 2023

1. Introduction

In autonomous driving scenarios, 3D object detection is crucial for understanding the surroundings. Cameras and LiDAR are two fundamental sensors used by autonomous vehicles. LiDAR provides precise 3D localization and velocity estimation, unaffected by weather or lighting conditions. However, LiDAR measurements are sparse, posing challenges for 3D perception. Cameras provide rich visual information, including color, texture, and pixel-level details. To overcome limitations, multi-modal data fusion has emerged as an effective approach, combining the strengths of different sensor modalities. In this paper, we present a simple and effective framework to fuse 3D LiDAR and high-resolution color measurements.

Existing fusion methods for multi-modal 3D point cloud object detection can be broadly categorized into three groups: point-level fusion, proposal-level fusion, and feature-level fusion. Point-level fusion methods, like Pointpainting [1] and Pointaugmenting [2], integrate information at the point level by projecting or incorporating additional data onto the point cloud, such as RGB images or semantic segmentation maps, to enhance the discriminative representation. Proposal-level fusion methods, including F-PointNets [3], F-ConvNet [4], MV3D [5], and AVOD [6], operate at the proposal level and combine images and point clouds to generate regional proposals, improving the proposal quality. Feature-level fusion methods, such as EPNet [7] and EPNet++ [8], fuse features extracted from different modalities, aiming to learn feature representations from both LiDAR point clouds and images.

Nevertheless, cutting-edge 3D object detectors often face challenges due to the sparsity and uneven distribution of data in outdoor scenes. This sparsity and uneven distribution negatively impact the performance of these detectors. To address these limitations, this paper introduces **Point-Rich**, a lightweight, flexible, and efficient method for multi-modal fusion. Point-Rich enhances various 3D object detection networks by supplementing the semantic characteristics of scene images assembled using LiDAR point clouds. This novel approach comprises two essential modules: the HD module and the HL module. The HD module generates a high-resolution point cloud in proximity to the 3D object, while the HL module distinguishes foreground points from background points by colorizing the point cloud in the vicinity of the object. The application of this method makes the measurement of LiDAR point clouds more accurate and robust and improves the accuracy and reliability of 3D object detection.

Our contributions can be summarized as follows:

- We propose Point-Rich, a lightweight plug-and-play module that encompasses two crucial components: the HighDensity module (HD) and the HighLight module (HL). By seamlessly integrating these modules into existing 3D object detection networks, the overall performance can be significantly enhanced.
- The HighDensity (HD) module aims to enhance the density of the point cloud surrounding the object, ensuring more accurate and consistent measurements.
- The HighLight (HL) module enriches the point cloud with additional semantic features from images, enabling improved discrimination between the foreground object and the background.

2. Related Work

In this section, we present an overview of LiDAR-based 3D learning and multi-modal fusion-based 3D object detection, discussing different approaches within each category.

2.1. LiDAR-Based 3D Learning

Point-based methods. Point-based approaches utilize the original point cloud as input. One example is PointNet [9], which employs a symmetric function to extract features from point clouds for classification tasks. PointNet++ [10] extends PointNet to perform 3D detection by extracting key points and their features. Other methods, such as 3DSSD [11] and Point-RCNN [12], combine PointNet with region of interest (RoI) pooling or refinement modules for object detection.

Voxel-based methods. Voxel-based methods convert point cloud data into a dense voxel grid and perform classification and localization tasks on the grid. PointPillars [13] represents the point cloud as a bird's eye view (BEV) pseudo-image. VoxelNet [14] divides the point cloud into voxels and maps them to fixed-size voxel features for detection. SECOND [15] improves upon VoxelNet by introducing a feature pyramid to capture multi-scale features. Voxel-based methods offer a higher computational efficiency compared to point-based methods, enabling faster training on large-scale datasets.

PV-based methods. PV-based methods merge the advantages of point-based and voxel-based methods, combining detailed point cloud information with the modeling capabilities of voxels. PV-RCNN [16] includes a voxel feature encoder and a point cloud feature encoder, which encode voxel and point cloud information, respectively, and combine them for detection. While PV-based methods achieve a good performance, they often require significant computational resources.

2.2. Multi-Modal Fusion-Based 3D Object Detection

Multi-modal fusion-based 3D object detection aims to leverage complementary information from images and point clouds. Existing fusion methods can be categorized into point-level, proposal-level, and feature-level fusion.

Point-level fusion. Point-level fusion methods augment the point cloud with additional pixel features from images. For example, PointPainting [1] attaches semantic scores from a pretrained semantic segmentation network to each LiDAR point. PointAugmenting [2] decorates the point cloud with depth embeddings from camera images. However, obtaining these pixel features can be challenging, and aligning LiDAR points with image pixels can introduce geometric alignment issues.

Proposal-level fusion. Proposal-level fusion fuses detection results obtained from individual sensors. F-PointNets [3] and F-ConvNet [4] generate frustum point clouds based on 2D detection results and perform 3D object detection on each frustum. MV3D [5] and AVOD [6] generate region proposals separately from images and point clouds and fuse them for refined 3D bounding box estimation. Proposal-level fusion can face challenges when dealing with occluded objects.

Feature-level fusion. Feature-level fusion combines features extracted from both sensors through domain transformation. Methods like EPNet [7], 3D-CVF [17], MMF [18], and ContFuse [19] use different techniques to fuse features at different scales. Recent works, such as DeepFusion [20], AutoAlign [21], and TransFusion [22], employ Transformer models to dynamically capture correlations between camera and LiDAR features.

Our proposed method combines proposal-level and point-level fusion approaches. In the HighDensity (HD) module, we generate dense point clouds in point cloud scenes using region proposals from images. In the HighLight (HL) module, we perform point-level fusion by projecting additional information onto point clouds to enhance their representation. This combination addresses occlusion issues in proposal-level fusion and alignment problems in point-level fusion. The HD and HL modules work in tandem to enhance the overall performance of Point-Rich in 3D object detection tasks. This combined approach improves the accuracy and reliability of object detection.

3. Methodology

The methodology of the proposed method, Point-Rich, is outlined in Figure 1. Point-Rich consists of two essential components: the HighDensity module (HD) and the HighLight module (HL). These modules work together to enhance the original sparse 3D point cloud by generating dense virtual points and decorating them with semantic segmentation information. The HD and HL modules work together to enhance Point-Rich. The HD module ensures a denser point cloud representation, capturing more detailed information about the target object. Simultaneously, the HL module decorates meaningful semantic features, enabling a more accurate and comprehensive analysis of the object. Finally, these dense semantic points are used in 3D object detection tasks to realize the accurate perception and recognition of objects in the environment.

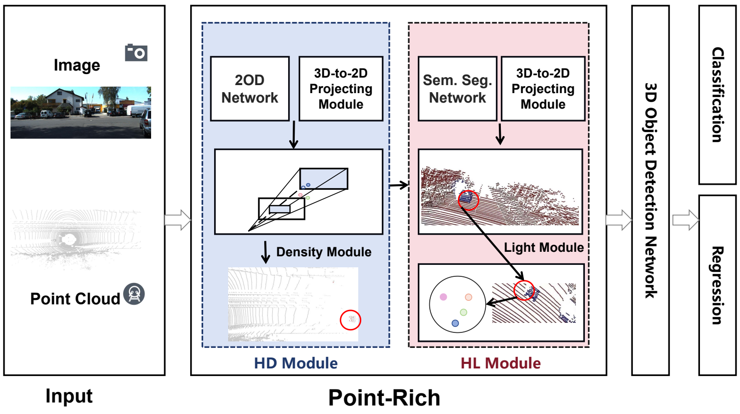

Figure 1. Overview of **Point-Rich**. Point-Rich comprises two essential components: (1) **H**igh**D**ensity module; (2) **H**igh**L**ight module. The HD module enhances point cloud density around the instance, providing a more robust and advanced representation. The HL module uses segmentation scores to decorate point clouds and generate point clouds containing semantic segmentation information.

In summary, Point-Rich's HD module enhances the point cloud density, while the HL module decorates valuable semantic features. These two components work in tandem to improve the overall performance of Point-Rich in 3D object detection tasks. This combined approach improves the accuracy and reliability of object detection.

3.1. HighDensity Module

Generate dense semantic virtual points. The density imbalance in distant regions affects the accuracy and stability of 3D object detection. To overcome this issue, the HD module is employed to generate virtual point clouds related to the target object, enhancing the point cloud density and ensuring consistent measurements. By leveraging the mature stage of 2D object detection, instance frustums can be generated, serving as a basis for creating virtual point clouds. This approach addresses the density imbalance between objects by adjusting the density of virtual point clouds as needed, providing a more balanced representation across distances. Virtual points enable consistent descriptions of object shapes and features at different distances. Combining virtual points with the original point cloud enhances the initial measurement, supporting subsequent 3D object detection algorithms. This module improves the accuracy and reliability of 3D object detection by making LiDAR point cloud measurements more accurate and robust.

Specifically, given a set of 2D detection results, we extend the original LiDAR points and initially generate enhanced LiDAR points $P = (x, y, z, r)$, where (x, y, z) is the 3D position and r is the semantic feature from the 2D detector. For simplicity, we use 2D detector class scores as semantic features.

We initiate the process by projecting the point cloud onto our detection. This involves converting each LiDAR point $(x, y, z, r)_i$ into the reference frame of the RGB camera. Subsequently, we project these points into image coordinates p, accompanied by their associated depth d, utilizing a perspective projection. For a specific 2D detection j, the assembly of all projected points and depth values forms the object frustum $F_j = (\overrightarrow{p}, d)$. The frustum only considers projected 3D points pi that fall within the detection mask m_j. Any LiDAR measurement falling outside detection masks is disregarded. Following this, we proceed to generate virtual points from each frustum F_j.

Then, we randomly sample 2D points $s \in m$ from each instance mask m_j. For each sampled point s_k, a depth estimate d is obtained from its nearest neighbor in the frustum F_j. Upon acquiring the depth estimate, we reverse the projection to return the point to its 3D space.

For each detection with relevant instance mask m_j, we generate a fixed number of τ multi-modal virtual points, which together with the original point cloud form the preliminary enhanced LiDAR points. We provide the algorithm for this process in Algorithm 1.

3.2. HighLight Module

Recode the LiDAR point cloud. In Point-Rich, the semantic segmentation network plays a vital role in enhancing the precision and robustness of point cloud processing. The image network, which receives input images, is responsible for generating per-pixel class scores. By leveraging the 2D semantic segmentation network, accurate segmentation information can be obtained, preserving fine contour details of objects and ultimately enhancing the accuracy of object detection and segmentation.

However, the deep ambiguity that arises within the HD module can significantly impact the accuracy and robustness of point cloud processing. To address this issue, we propose an HL module where LiDAR points are projected onto the output of the image-only semantic segmentation network. Subsequently, class scores are attached to each point, enabling us to effectively handle complex scenarios involving variations in distance and environmental conditions. By adopting this approach, we can improve the overall robustness and stability of point cloud processing.

Specifically, the algorithm projects LiDAR points into the image, attaching the segmentation fraction of the relevant pixels to the LiDAR points to create a drawn LiDAR point. For the KITTI dataset, the output of the semantic segmentation network is a class score with

C channels, where $C = 4$ (car, pedestrian, bicycle, background). This projection method can better maintain the shape information of the object and can better deal with complex situations, such as environmental changes and distance changes. We provide pseudo-code for this process in Algorithm 2.

Algorithm 1: Generate dense semantic virtual points to initially enhance the point cloud

```
Input: LiDAR point $P = (x, y, z, r)$; Instance mask $\{m_1, \ldots, m_n\}$; Homogeneous
       transformation matrix $T \in R^{4\times4}$; Camera matrix $C \in R^{3\times4}$.
Output: Preliminary enhanced LiDAR point $P$.
for $p = (x, y, z, r) \in P$ do
    /* Perspective projection to 2D point p depth d                 */
    $\overrightarrow{p}, d \leftarrow PROJRCT(C, T, p)$
    for $j \in \{1, \ldots n\}$ do
        if $\overrightarrow{p} \in m_j$ then
            $F_j \leftarrow F_j \cup \{(\overrightarrow{p}, d)\}$ // Add point to frustum
        end
    end
end
for $j \in \{1, \ldots n\}$ do
    $S \leftarrow Sample_\tau(m_j)$ for $k \in \{1, \ldots \tau\}$ do
        $(\overrightarrow{p}, d) \leftarrow NN(S_k, F_j)$ // Find closest projected 3D point
        /* Unproject the 2D point using the nearest neighbors depth  */
        $q \leftarrow Unproject(C, T, (S_k, d))$
        Add $(q, e)$ to $P$
    end
end
```

Algorithm 2: Recode the LiDAR point cloud

```
Input: LiDAR point $P = (x, y, z, r)$; Instance mask $\{m_1, \ldots, m_n\}$; Homogeneous
       transformation matrix $T \in R^{4\times4}$; Camera matrix $C \in R^{3\times4}$.
Output: Preliminary enhanced LiDAR point $P$.
for $p = (x, y, z, r) \in P$ do
    $\overrightarrow{p} \leftarrow PROJRCT(C, T, p)$
    $\overrightarrow{s} = S[\overrightarrow{p}[0], \overrightarrow{p}[1], :]$
    $p = Concatenate(p, \overrightarrow{s})$
end
```

3.3. Three-Dimensional Object Detection

Two-dimensional object detection networks often encounter challenges such as imprecise 2D bounding box regression, and the HD module's performance is influenced by the output of any 2OD network. To address these issues, we draw inspiration from the data augmentation techniques used in 2OD networks and propose a simple yet effective training strategy for the 3D object detection network during both the training and inference stages.

During the training stage, we introduce a strategy to handle the variability in 2D bounding box sizes. We randomly increase the length and width of the 2D bounding box by 0% to 10%. This augmentation ensures that the 3D object detection network can handle larger bounding boxes accurately. In the inference stage, we address the problem of small-sized 2D bounding boxes by enlarging their dimensions by 5%. This expansion enhances the receptive field of the frustum point cloud. Furthermore, we apply a confidence threshold of 0.1 to filter out 2D bounding boxes with low confidence scores in the KiTTI dataset. This filtering step helps to improve the overall detection accuracy by removing uncertain or unreliable detection results. By incorporating these modifications, we aim to enhance the performance and robustness of the 3D object detection network, enabling more accurate and reliable object detection in various scenarios.

The decorated point cloud can be used by LiDAR detectors that learn an encoder since Point-Rich simply changes the input dimension and number of the LiDAR points. In this paper, we prove that Point-Rich is suitable for different LiDAR detectors: PointPillars [13], PointRCNN [12], and VirConv [23]. Despite these different design options, all LiDAR detectors benefit from Point-Rich.

4. Experiments

4.1. Dataset

We validate our proposed Point-Rich 3D detector on KITTI [24].

KITTI. KITTI is a popular benchmark and evaluation framework for 3D object detection in autonomous driving scenarios. The dataset was collected with a vehicle equipped with a 64-beam Velodyne LiDAR point cloud and a single PointGrey camera. Both the training and testing datasets in this paper are derived from the KITTI dataset. It contains 7481 LiDAR and image frames for training and 7518 for testing. The KITTI object detection benchmark requires the detection of cars, pedestrians, and cyclists. Following the commonly applied setting, training data are divided into 3712 samples and 3769 samples for train split and val split. The KITTI datasets are divided into easy, medium, and hard difficulties, and the official KITTI leaderboard ranks performance based on medium average accuracy. In the evaluation, we used the mean average precisionn(mAP) as a primary metric. The mAP was calculated using recall 40-point precision (R40) and 11-point precision. For object orientation detection, the average orientation similarity (AOS) is used to measure the degree of orientation similarity between detection results and real labels. Visualization of detection results in Figure 2 prove our method effective.

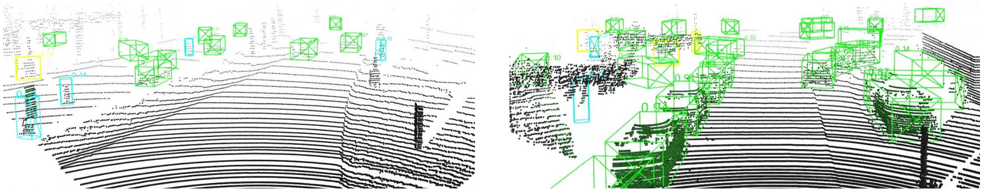

Figure 2. Qualitative analysis of KITTI results. PointRich+PointPillars predicted 3D bounding boxes are displayed on the input point cloud.

4.2. Experiment Settings

The enriched point cloud can be seamlessly integrated into various 3D object detection networks employing both pillar-based and point-based encoders. To demonstrate the effectiveness of Point-Rich across different point-based 3D object detection networks, we conduct extensive experiments on KITTI. We compare our approach against the two-stage PointRCNN and pillar-based Pointpillar, which both serve as our baseline models. We utilized publicly available implementations of PointPillars and PointRCNN algorithms and extended them to incorporate semantic segmentation scores for four classes into the point cloud. This augmentation resulted in an increase in the dimensions of the decorated point cloud from 9 to 13 for PointPillars and from 4 to 8 for PointRCNN. In the case of PointPillars, the modified encoder now has (13, 64) channels. The 8-dimensional augmented point cloud for PointRCNN is used as input for both the encoder and the region pooling layer. According to the model configuration provided in SASA [25], we train PointRCNN with the ADAM optimizer for 80 epochs. The learning rate schedule is a one-cycle schedule [26] with a peak learning rate at 0.01. We follow the original data augmentation strategies and inference settings. No additional modifications were made to the public experimental configurations. Please refer to [12,25] for more details.

In addition, we use Yolov5 [27] for 2D detection and the DeepLabv3+ [28] network [28] for semantic segmentation on the KITTI dataset.

Yolov5. For the KITTI dataset, we train 100 epochs in an end-to-end manner using the largest model, Yolov5x, with the SGD optimizer, and use an image size of 1280 pixels in the training and reasoning phases. In addition, we merge cars, trucks, and vans into the car category, leaving only the car, pedestrian, and cyclist categories for training and reasoning. We follow the original data enhancement strategy, training, and inference settings.

DeepLabv3+ [28]. For the image-based semantic segmentation, the segmentation scores for our KITTI experiments are generated from DeepLabv3+. The network is first pretrained on Mapillary [29], then finetuned on Cityscapes [30], and finally finetuned again on KITTI pixelwise semantic segmentation [31]. Note that in semantic segmentation and object detection, the class definition of a rider is different. In detection, the cyclist is defined as rider on a bike, while in semantic segmentation, the cyclist is defined as a rider only and the bike is a separate class. Therefore, there is a need to map the bikes to the cyclist class when a rider is within its 1 m radius and the rest to the background.

4.3. Main Results

We evaluate the effectiveness of our proposed point cloud preprocessing module on the KITTI datasets. To broadly validate point cloud data preprocessing, we selected three distinct and representative baselines, PointPillars, PointRCNN, and VirConv, for evaluation.

For the visualization of the HL module, the results are shown in Figure 3, where the green and red points are decorative points, and the black points are background point. For the visualization of the HD module, the results are shown in Figure 4, where the red points are newly generated near the target object and the gray points are the original point cloud. Table 1 shows the 3D object detection performance on the KITTI validation set. In the most competitive car category, adding Point-Rich to VirConv-L surpasses most existing 3D detection networks in terms of 3D detection, bird's eye view, and directional average accuracy.

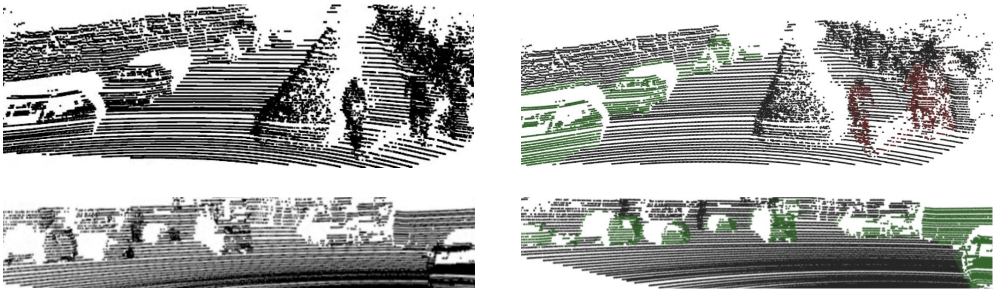

Figure 3. Visualization of HL module results. We create two different comparison figures. We show our HL module results of two different scenes. The pictures on the left are the original point cloud, and the pictures on the right are the result after HL module processing. The green and red points are decorative points, and the black points are background points.

The core components proposed in this paper are added to existing LiDAR-based object detection methods. The experimental results are shown in Tables 2 and 3. Experiments have shown that, with the addition of Point-Rich, the LiDAR method can significantly increase the average accuracy of cars, pedestrians, and bicycles in 3D object detection tasks conducted from an aerial view on the KITTI verification set. In our study, we successfully improved the detection accuracy of pedestrians and bicycles. This means that our approach is better able to distinguish and identify these two categories of targets, providing more accurate environmental awareness for autonomous driving systems. By improving pedestrian and cyclist detection, we can improve the safety and reliability of autonomous driving systems and reduce the potential risk of traffic accidents.

Table 1. Results on the car category of the KITTI val set. The evaluation metrics of 3D detection, bird's eye view, and orientation average precision are calculated on 40 recall points. "L" and "C" indicate the LiDAR and camera, respectively.

Methods	Year	Modality	3D Detection			Bird's Eye View			Orientation		
			Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointPillars [13]	2019	L	87.91	76.40	73.76	92.53	87.52	85.39	95.54	91.57	90.84
PointRCNN [12]	2019	L	92.04	82.63	80.55	93.19	90.31	88.48	95.90	94.01	91.84
3DSSD [11]	2020	L	92.37	83.11	80.13	95.22	91.31	88.72	98.63	95.07	92.50
3DSSD_SASA [25]	2022	L	92.23	85.28	82.58	95.38	91.81	89.27	98.62	95.52	94.78
SFD [32]	2022	L + C	92.73	87.82	84.59	95.78	92.68	90.19	99.05	96.72	94.95
VirConv-L [23]	2023	L + C	**93.21**	88.02	85.61	**96.16**	**93.50**	**91.39**	**99.22**	97.52	**95.05**
VirConv-L+Point-Rich	-	L + C	93.03	**88.22**	**85.75**	95.98	93.44	91.34	99.15	**97.62**	95.02

Table 2. Results on the KITTI val set under the bird's eye view. PointPillars is used as the baseline.

Methods	mAP	Car			Pedestrian			Cyclist		
	Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
Pointpillars	66.88	92.53	87.52	85.39	59.63	52.82	48.32	83.97	60.31	56.31
Pointpillars+Point-Rich	72.41	93.15	87.82	85.56	66.51	60.58	56.52	89.12	68.84	64.55
Delta	+5.53	+0.62	+0.30	+0.17	+6.88	+7.76	+8.20	+5.15	+8.53	+8.24

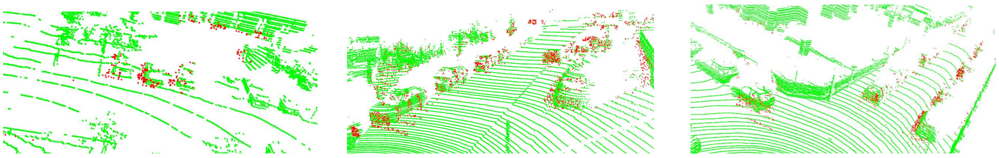

Figure 4. Visualization of HD module results. The red points are the enhancements generated near the target object, and the green points are the original point cloud.

Table 3. Results on the car category of the KITTI val set. PointRCNN is used as the baseline.

Methods	Detection			Bird's Eye View			Orientation		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointRCNN	91.45	82.28	78.33	93.08	88.73	86.34	95.78	93.19	89.57
PointRCNN+Point-Rich	92.04	82.63	80.55	93.19	90.31	88.48	95.90	94.01	91.84
Delta	+0.59	+0.35	+2.22	+0.11	+1.58	+2.14	+0.12	+0.82	+2.27

All of the experimental results above show that our method is more suitable for 3D networks and can significantly improve the performance of existing 3D object detection networks.

4.4. Ablation Study

To demonstrate the effectiveness of the different modules proposed in this work, we conduct module ablation experiments and compare them according to the order of Point-Rich. These ablation module experiments are performed on the KITTI verification set, and the ablation results of each module are shown in Table 4.

Effects of HD module. Comparing the first row (without adding any modules) with the second row (adding the HD module), we can see improvements in the detection accuracy. For example, in the easy scenario, the detection accuracy for cars increased from 92.53% to 93.81%. This indicates that the HD module focuses on enhancing the point cloud density surrounding the target object, resulting in a more comprehensive and accurate representation. It effectively improves the detection performance in various scenarios.

Effects of HL module. Comparing the first row with the third row (adding the HL module), we can observe significant improvements in detection accuracy across all scenarios. By converting the point cloud data into a visual form, the HL module provides intuitive visual information that helps the algorithm better understand the scene and target. This enhancement in understanding the shape, size, and orientation of the target leads to improved accuracy and robustness in pedestrian detection.

Table 4. Ablation results of our Point-Rich in the bird's eye view (BEV) of the KITTI val set. PointPillars is used as the baseline. The 3D average precision metric is calculated on 40 recall points.

PointPillars	HD Module	HL Module	Car			Pedestrian			Cyclist		
			Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
✓			92.53	87.52	85.39	59.63	52.82	48.32	83.97	60.31	56.31
✓	✓		**93.81**	87.32	85.13	59.40	52.52	48.46	83.29	59.65	55.97
✓		✓	93.19	87.71	**86.99**	64.35	59.57	56.28	89.07	68.57	**65.03**
✓	✓	✓	93.15	**87.82**	85.56	**66.51**	**60.58**	**56.52**	**89.12**	**68.84**	64.55

After adding the HD module and HL module, the effect is better than that of adding a single module. By observing the data in Table 4, we believe that a small number of noise points may be introduced in the HD module due to the depth ambiguity, but after the HL module these noise points are distinguished and weakened, so as to improve the detection effect.

Both the HD module and the HL module play an important role in the proposed Point-Rich and contribute significantly to the performance of the overall algorithm. This further proves the effectiveness and superiority of the proposed method. By introducing the HD module and HL module, we have successfully improved the quality and availability of LiDAR point cloud data, laying a solid foundation for subsequent target detection missions. The HL module provides intuitive visual information by converting point cloud data into a visual form, which helps the algorithm better understand the scene and target. This decoration process enhances the understanding of features such as the shape, size, and orientation of the target, thus improving the accuracy and robustness of object detection.

4.5. Compared with SOTA

VirConv-S [23], the SOTA method, generates a large number of virtual points with the help of a deep completion network, which is similar to our HighDensity module. However, the number of virtual points generated by VirConv-S is much larger than the original point cloud, which greatly increases the amount of input data and model parameters. Our approach increases the point cloud density while maintaining computational efficiency. This makes our method stand out among many 3D object detection methods.

4.6. Limitations

The limitations of Point-Rich are summarized below:

Blindness to objects unrecognized by 2D networks. Point-Rich relies on the output of 2D networks to enrich the point cloud data. However, if an object is not detected by the 2D network, Point-Rich cannot compensate for this limitation. Consequently, the 3D network may remain unaware of such objects, leading to their omission in the final detection results.

Potential enhancement of false positives. In cases where the 2D object detection network mistakenly identifies an object, Point-Rich may inadvertently amplify the features of the corresponding point cloud. This may result in an increased likelihood of false positives, as the enriched features may mislead the subsequent 3D object detection network.

It is important to consider these limitations when applying Point-Rich in practical scenarios. Further research and improvements are necessary to address these challenges and enhance the overall effectiveness and robustness of Point-Rich in multi-modal 3D object detection tasks.

5. Conclusions

This paper proposed Point-Rich, a simple yet effective method for multi-modal fusion, aiming to improve the detection performance of 3D object detection networks. Our main concept is to mitigate the discrepancy between objects that are near and far in the point clouds and to distinguish the foreground points from background points. Extensive experiments conducted on the KITTI dataset demonstrate the effectiveness and superiority of Point-Rich and show the impressive improvements for the 3D object detection networks. In addition, by integrating lightweight and flexible Point-Rich modules, several 3D object detection networks are empowered to prioritize regions that are more likely to contain target objects. Our proposed method provides a promising direction for 3D object detection. Not only can it be implemented in point-based models but it is also compatible with, for example, pillar-based networks. We hope that our work encourages further advancements in this field and encourages researchers to explore innovative solutions for addressing the challenges in 3D object detection.

Author Contributions: Methodology, Y.Z. (Yanchao Zhang); Software, Q.W.; Validation, L.G.; Formal analysis, D.Z.; Data curation, H.Z.; Writing—original draft, Y.Z. (Yinuo Zheng); Writing—review & editing, X.B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of Jiangsu Province of China under Grants No. BK20222012 and No. BK20231439, in part by the China National Key R&D Program during the 14th Five-year Plan Period under Grant No. 2022YFB2901600, in part by the Fund of Prospective Layout of Scientific Research for NUAA, and in part by the Fundamental Research Funds for the Central Universities under Grant NS2023052.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The accessed data that support the findings of this study are openly available at https://github.com/qiucun2/PointRich, accessed on 19 November 2023.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Vora, S.; Lang, A.H.; Helou, B.; Beijbom, O. Pointpainting: Sequential fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 4–19 June 2020; pp. 4604–4612.
2. Wang, C.; Ma, C.; Zhu, M.; Yang, X. Pointaugmenting: Cross-modal augmentation for 3D object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11794–11803.
3. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum pointnets for 3D object detection from rgb-d data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 918–927.
4. Wang, Z.; Jia, K. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Macau, China, 3–8 November 2019; pp. 1742–1749.
5. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3D object detection network for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1907–1915.
6. Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; Waslander, S.L. Joint 3D proposal generation and object detection from view aggregation. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–8.
7. Huang, T.; Liu, Z.; Chen, X.; Bai, X. Epnet: Enhancing point features with image semantics for 3D object detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XV 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 35–52.
8. Liu, Z.; Huang, T.; Li, B.; Chen, X.; Wang, X.; Bai, X. EPNet++: Cascade bi-directional fusion for multi-modal 3D object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 8324–8341. [CrossRef] [PubMed]
9. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
10. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5105–5114.

11. Yang, Z.; Sun, Y.; Liu, S.; Jia, J. 3dssd: Point-based 3D single stage object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 4–19 June 2020; pp. 11040–11048.
12. Shi, S.; Wang, X.; Li, H. Pointrcnn: 3D object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 770–779.
13. 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.
14. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
15. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
16. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 4–19 June 2020; pp. 10529–10538.
17. Yoo, J.H.; Kim, Y.; Kim, J.; Choi, J.W. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3D object detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXVII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 720–736.
18. Liang, M.; Yang, B.; Chen, Y.; Hu, R.; Urtasun, R. Multi-task multi-sensor fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7345–7353.
19. Liang, M.; Yang, B.; Wang, S.; Urtasun, R. Deep continuous fusion for multi-sensor 3d object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 641–656.
20. Li, Y.; Yu, A.W.; Meng, T.; Caine, B.; Ngiam, J.; Peng, D.; Shen, J.; Lu, Y.; Zhou, D.; Le, Q.V.; et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 17182–17191.
21. Chen, Z.; Li, Z.; Zhang, S.; Fang, L.; Jiang, Q.; Zhao, F.; Zhou, B.; Zhao, H. Autoalign: Pixel-instance feature aggregation for multi-modal 3D object detection. *arXiv* **2022**, arXiv:2201.06493.
22. Bai, X.; Hu, Z.; Zhu, X.; Huang, Q.; Chen, Y.; Fu, H.; Tai, C.L. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1090–1099.
23. Wu, H.; Wen, C.; Shi, S.; Li, X.; Wang, C. Virtual Sparse Convolution for Multimodal 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 21653–21662.
24. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
25. Wu, H.; Deng, J.; Wen, C.; Li, X.; Wang, C.; Li, J. CasA: A cascade attention network for 3-D object detection from LiDAR point clouds. *IEEE Trans. Geosci. Remote. Sens.* **2022**, *60*, 1–11. [CrossRef]
26. Smith, L.N.; Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. In Proceedings of the Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, Baltimore, MD, USA, 15–17 April 2019; Volume 11006, pp. 369–386.
27. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
28. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018.
29. Neuhold, G.; Ollmann, T.; Rota Bulo, S.; Kontschieder, P. The mapillary vistas dataset for semantic understanding of street scenes. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4990–4999.
30. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
31. Cheng, B.; Collins, M.D.; Zhu, Y.; Liu, T.; Huang, T.S.; Adam, H.; Chen, L.C. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 4–19 June 2020; pp. 12475–12485.
32. Wu, X.; Peng, L.; Yang, H.; Xie, L.; Huang, C.; Deng, C.; Liu, H.; Cai, D. Sparse fuse dense: Towards high quality 3d detection with depth completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5418–5427.

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.