remote sensing

MDPI

Article

CFFCNet: Center-Guided Feature Fusion Completion for Accurate Vehicle Localization and Dimension Estimation from Lidar Point Clouds

Xiaoyi Chen [1,†], Xiao Feng [1,†], Shichen Zhang [2], Wen Xiao [1,3,*], Miao Tang [1] and Kun Sun [4]

[1] School of Geography and Information Engineering, China University of Geosciences, Wuhan 430074, China; 1202121024@cug.edu.cn (X.C.); fx@cug.edu.cn (X.F.); miao.tang@cug.edu.cn (M.T.)
[2] College for Elite Engineers, China University of Geosciences, Wuhan 430074, China; shichen.zhang@cug.edu.cn
[3] National Engineering Research Center of Geographic Information System, China University of Geosciences, Wuhan 430074, China
[4] School of Computing, China University of Geosciences, Wuhan 430074, China; sunkun@cug.edu.cn
* Correspondence: wen.xiao@cug.edu.cn
† These authors contributed equally to this work.

Highlights

What are the main findings?

- CFFCNet achieves substantial improvements in vehicle localization and dimension estimation from incomplete lidar point clouds, reducing localization MAE to 0.0928 m and length MAE to 0.085 m on the KITTI dataset.
- The proposed Branch-assisted Center Perception (BCP) module effectively addresses the misalignment between detection centers and true geometric centers in real-world scenarios, while the Multi-scale Feature Blending Upsampling (MFBU) module enhances geometric reconstruction through hierarchical feature fusion.

What is the implication of the main finding?

- The method demonstrates strong generalization capability from vehicle-mounted to roadside lidar data without fine-tuning, achieving localization MAE of 0.051 m on the CUG-Roadside dataset, enabling practical deployment in infrastructure-based traffic monitoring systems.
- Center-guided point cloud completion provides a new paradigm for precise 3D perception in intelligent transportation systems, contributing to safer autonomous navigation and more accurate traffic flow analysis in urban environments.

Abstract

Accurate scene understanding from 3D point cloud data is fundamental to intelligent transportation systems and geospatial digital twins. However, point clouds acquired from lidar sensors in urban environments suffer from incompleteness due to occlusions and limited sensor resolution, presenting significant challenges for precise object localization and geometric reconstruction—critical requirements for traffic safety monitoring and autonomous navigation. To address these point cloud processing challenges, we propose a Center-guided Feature Fusion Completion Network (CFFCNet) that enhances vehicle representation through geometry-aware point cloud completion. The network incorporates a Branch-assisted Center Perception (BCP) module that learns to predict geometric centers while extracting multi-scale spatial features, generating initial coarse completions that account for the misalignment between detection centers and true geometric centers in

Check for updates

Academic Editors: Tarig Ali, Ruijin Ma and Maolin Chen

Received: 14 November 2025
Revised: 16 December 2025
Accepted: 19 December 2025
Published: 23 December 2025

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

real-world data. Subsequently, a Multi-scale Feature Blending Upsampling (MFBU) module progressively refines these completions by fusing hierarchical features across multiple stages, producing accurate and complete vehicle point clouds. Comprehensive evaluations on the KITTI dataset demonstrate substantial improvements in geometric accuracy, with localization mean absolute error (MAE) reduced to 0.0928 m and length MAE to 0.085 m. The method's generalization capability is further validated on a real-world roadside lidar dataset (CUG-Roadside) without fine-tuning, achieving localization MAE of 0.051 m and length MAE of 0.051 m. These results demonstrate the effectiveness of geometry-guided completion for point cloud scene understanding in infrastructure-based traffic monitoring applications, contributing to the development of robust 3D perception systems for urban geospatial environments.

Keywords: point cloud completion; vehicle localization; deep learning; multi-feature fusion; center prediction; point clouds processing

1. Introduction

With the development of lidar technology, the field of 3D vision has witnessed rapid advancements [1–4]. Object detection (OD) using lidar data has become an important research topic [5,6], with wide-ranging applications, such as autonomous driving [7], collision detection [8], collision avoidance [9], and traffic regulation [10]. At the vehicle end, environment information is automatically obtained through perception algorithms using lidar point clouds, enabling obstacle recognition and localization [11]. On the roadside, lidar perception devices have been deployed to gather road user information, providing comprehensive environmental data for all vehicles, autonomous or not, and enabling highly accurate and real-time traffic flow information [12,13].

As a fundamental task in perception, object detection (OD) using lidar data has become an important research topic [5,6], with wide-ranging applications, such as anomaly detection [8], collision avoidance [9], and traffic regulation [10]. Traditional 2D image OD algorithms have been migrated to the 3D domain with the emergence of 3D deep learning methods based on point clouds [14,15]. 3D OD algorithms are used to recognize and localize objects within point clouds, describing the results using bounding boxes that encompass the detected objects, while the shape and posture information of detected objects is conveyed through bounding boxes and orientation angles. Through continuous advances in point cloud deep learning networks and integration with other types of perceptual data, significant progress in on-road OD has been achieved [16–20].

Nevertheless, in real-world applications, a compelling need exists for precise positioning and dimension estimation, where OD bounding boxes cannot suffice [21]. Precise object positioning enables accurate motion calculation and determination of relative distances between objects. This not only helps prevent collisions but also facilitates high-precision and detailed traffic flow analysis. Moreover, accurate positioning using roadside lidar could mitigate the issue of insufficient precision of Global Navigation Satellite Systems (GNSSs) in urban valleys. However, due to the limited number of beams in the lidar sensor and restricted point of view, lidar points are often sparse and occluded during data collection [22]. This lack of geometric information has several consequences. On the one hand, it can affect the accuracy of detection, preventing the attainment of precise localization [23]. On the other hand, the point cloud within the detected bounding box is incomplete, resulting in inaccurate dimensions.

Over the past few years, with the success of deep learning for 3D point clouds [24,25], many shape completion techniques based on points have emerged [26–30]. While these algorithms enhance the details of the completed point cloud, their impact on absolute positioning accuracy remains unquantified. Motivated by this gap, we systematically investigate whether integrating shape completion can enhance the localization and dimension-estimation accuracy of detected vehicles in real-world scenarios.

In addressing this, there are two main difficulties: (1) Existing point cloud completion models are predominantly trained on simulated datasets, where the geometric center of incomplete point clouds is known and located at the coordinate origin [31]. Nevertheless, in real-world datasets like KITTI [32], the geometric centers of vehicles remain unknown, which leads to inaccurate point cloud completion results as depicted in Figure 1.

(2) During the completion process, it is common to generate a coarse point cloud directly from global features, leading to the neglect of the local contextual information of the point cloud [33].

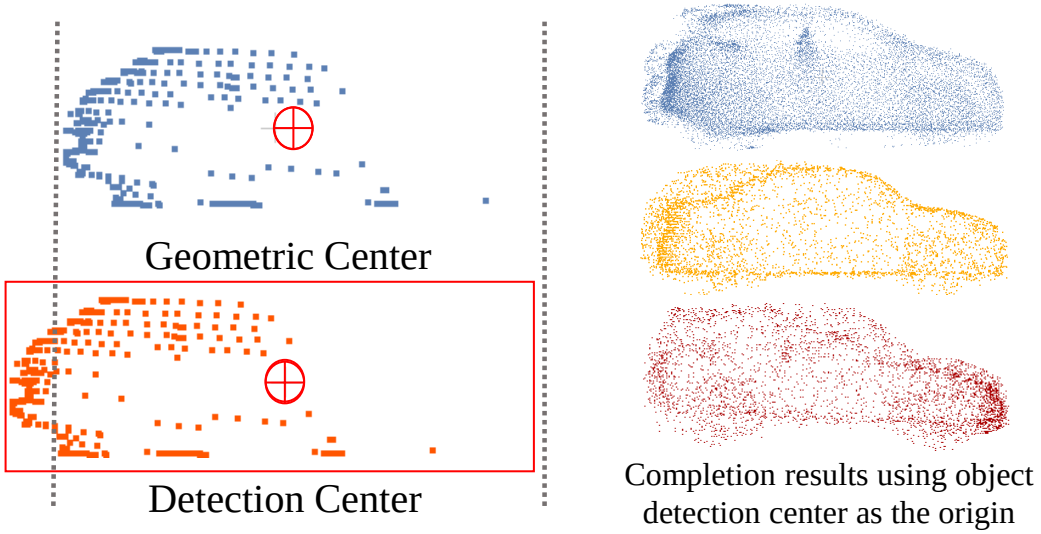


Figure 1. Impact of center misalignment. (**Left**) Difference between geometric and detection centers. (**Right**) Completion offsets on KITTI caused by misalignment.

To address this issue, we propose a Center-guided Feature Fusion Completion Network (CFFCNet), which consists primarily of a Branch-assisted Center Perception (BCP) module and a Multi-scale Feature Blending Upsampling (MFBU) module. Experiments on the KITTI dataset and a roadside lidar dataset without parameter fine-tuning demonstrate the effectiveness and generalization capability of the proposed network. The main contributions of this paper are as follows:

1. We propose CFFCNet, a novel geometry-aware shape completion network explicitly designed to enhance vehicle localization and dimension estimation. By integrating center guidance into the completion process, our method effectively addresses the critical misalignment issue between detection centers and true geometric centers in real-world sparse point clouds.
2. We introduce a Branch-assisted Center Perception (BCP) module and a Multi-scale Feature Blending Upsampling (MFBU) module. The BCP module provides implicit supervision for accurate center and dimension prediction, while the MFBU module employs a cross-attention mechanism to fuse multi-level local and global features. This combination significantly improves the recovery of fine-grained geometric details and ensures high-fidelity completion.

The rest of the paper is organized as follows. Section 2 provides background of the closely related work. Section 3 introduces the proposed CFFCNet in detail. Section 4 presents training, validation, and ablation experiments. Section 5 presents and evaluates the results. Section 6 discusses the performance of CFFCNet and outlines future work. Finally, Section 7 concludes the paper with a summary of the key findings.

2. Related Work

2.1. 3D Object Detection

With the emergence of PointNet [24] and PointNet++ [25], direct feature extraction from point clouds has become the primary approach in 3D neural networks. PointR-CNN [34] was the first to migrate 2D OD algorithms to the 3D domain, directly achieving 3D OD through point-based methods. Subsequently, VoteNet [35] introduced a vote-based 3D OD model, replacing the Region Proposal Network (RPN) approach to reduce computational overhead. 3DSSD [36] replaced downsampling layers with the farthest-point sampling method, reducing computational complexity while obtaining richer 3D semantic information. Pointformer [37] applied attention mechanisms to 3D OD, using both local and global attention mechanisms to extract point cloud features. CT3D [38] utilized channel attention to enhance feature learning within detection boxes, refining the second-stage detection boxes. Focal Transformer [39] introduced a shape feature module to enhance the perception of 3D objects, leading to improved detection accuracy. DI-V2X [40] aimed to learn domain-invariant representations through a new distillation framework to mitigate the domain discrepancy in the context of 3D object detection.

However, due to the characteristics of lidar, the collected point clouds often exhibit incompleteness caused by occlusion and density issues. To alleviate these challenges in detection and localization, PC-RGNN [41] utilized a point cloud completion module to address uneven point cloud distribution and enhance OD accuracy. Associate-3Ddet [42] employed learned incomplete feature matching to restore point clouds. SIENet [43] proposed using a point information enhancement module to increase the number of points in proposed boxes, strengthening their feature representation. BtcDet [44] predicted incomplete point clouds prior to learning information through symmetrical matching and other predictive strategies to enrich the original data. PA-RCNN [45] introduced a point augmentation module in the Region of Interest (ROI) stage, fusing original inputs and related features, and completing incomplete point clouds through symmetry. Nevertheless, previous work has primarily employed point cloud completion as intermediate processes to fill in features for OD.

2.2. Point Cloud Completion

In the early stages of point cloud completion tasks, voxel-based methods were commonly used [46–50]. Point clouds were voxelized to create regularized data, and 3D CNN structures were employed to analyze features, achieving superior results compared to traditional methods. However, the voxelization process led to information loss in point clouds and incurred high computational costs associated with 3D convolution networks. Such methods have become obsolete with the emergence of point-based approaches, inspired by PointNet [24] and PointNet++ [25].

Motivated by point-based learning, approaches like PCN [51], TopNet [52] and SA·Net [53] started using point-based methods to complete point clouds. PCN first encoded features through PointNet and PointNet++ and performed upsampling using the FoldingNet method [54] to obtain complete point clouds. TopNet considered the generation process as a tree-like structure, generating point clouds by producing child nodes from

parent nodes. SA·Net improved feature completeness in point clouds by incorporating skip connections into the completion model.

Subsequent work divided the generation and decoding task into multiple stages [22,52,53], progressively upsampling point clouds from coarse to fine to refine the shape of the completed point cloud. In recent years, with the rise of Transformer structures, Transformers such as PCT [2], Pointformer [37], and Point Transformer [55] have begun to make their way into point cloud processing. Utilizing the advantages of Transformer's representation learning capabilities, PoinTr [56] framed point cloud completion as a set-to-set transformation problem and proposed a Transformer encoder structure for point cloud completion. By representing point clouds as a set of unordered points with positional embeddings, point clouds can be transformed into a sequence of point proxies. AdaPoinTr [57] extended PoinTr by using an adaptive query mechanism and introducing a denoising module during the completion process. SnowflakeNet [27], with Snowflake Point Deconvolutions (SPDs), applied a Transformer-based structure to the decoding process and modeled the generation of completed point clouds as a snowflake-like growth of points in 3D space. PMP-Net++ [58] used a path prediction approach to complete point clouds by predicting the generation path of point clouds. ODGNet [59] consists of a Seed Generation U-Net, which leverages multi-level feature extraction and concatenation to enhance the representation capability of seed points, and orthogonal dictionaries that can learn shape priors from training samples to compensate for the missing portions during inference. However, these tasks typically did not fully consider the gap between real-world data and completion using simulated data.

3. Method

In this section, the CFFCNet is introduced in detail, with the overall framework illustrated in Figure 2. Firstly, the extraction of input point features is initiated by a multi-level attention network, using SnowflakeNet [27] as the backbone. Following this, the BCP module is employed to guide the completion stage by predicting the center positions, enabling the derivation of both local and global features associated with the incomplete point cloud's center. Subsequently, the coarse point cloud generation is facilitated by combining the original input partial point cloud with the result of interpolated global information to establish the fundamental geometric skeleton. While this coarse completion focuses on the global shape, the subsequent refinement phase employs the MFBU module to integrate multi-scale local features, further refining and completing the coarse point cloud with fine details. Ultimately, the complete point cloud is progressively generated through multiple refinement modules. To effectively drive this architecture, we employ a multi-task loss strategy that jointly optimizes these components. Specifically, the BCP module is supervised by a geometric regression loss to ensure accurate center and dimension prediction, while the MFBU and refinement stages are guided by a multi-stage completion loss. This unified training objective ensures that the network simultaneously learns global geometric structure and local fine-grained details.

3.1. Branch-Assisted Center Perception (BCP) Module

As shown in Figure 2, a BCP module is incorporated in the feature extraction stage. The branch supervises the model parameters during training, allowing the network to implicitly learn its predictions of the center position. The detailed branch module is illustrated in Figure 3. The design of this branch module primarily involves two components: (1) branch feature extraction and (2) encoding loss regression.

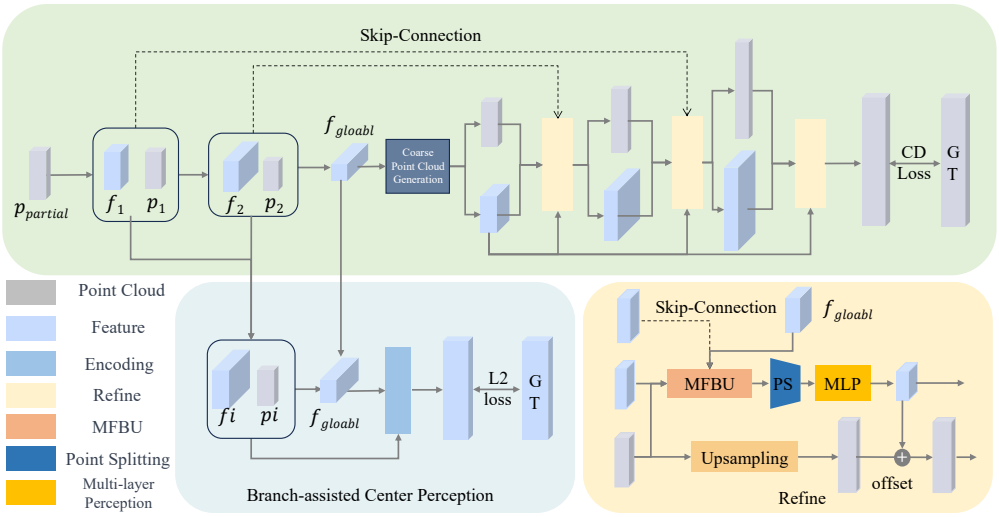


Figure 2. The overall architecture of CFFCNet. The network integrate the Branch-assisted Center Perception (BCP) and Multi-scale Feature Blending Upsampling (MFBU) modules into an encoder–decoder framework.

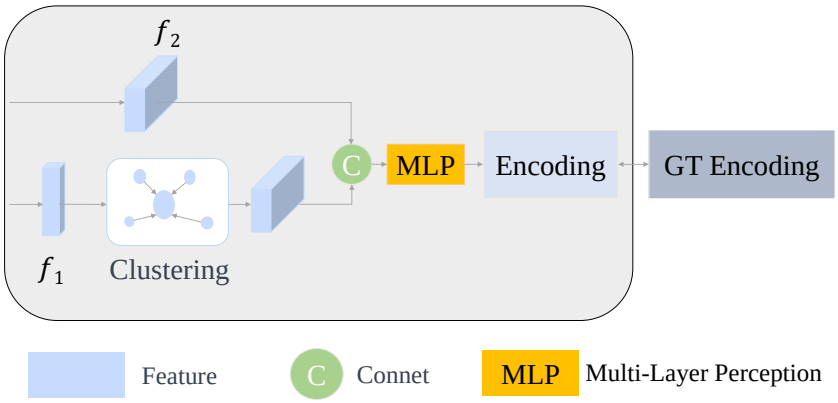


Figure 3. Structure of the Branch-assisted Center Perception (BCP) module. The module extracts multi-level features to predict center points and dimensions.

3.1.1. Branch Feature Extraction

Branch feature extraction takes the hierarchical features extracted at different levels (ranging from local geometric details to global semantic information) in the main feature extraction stage and uses them as inputs to the branch module. It uses attention mechanisms to blend multi-scale features, enabling the learning of local features within different ranges. This helps predict the center position and geometric information such as length, width, and height for the completed point cloud. The process is expressed as:

$$F^i = MLP_g\left(G(\text{Concat}[F^{(i-1)}, f^i])\right), \tag{1}$$

where f^i represents the i-th level of the point cloud feature extraction network, F^i denotes the features combined from the previous i layers, and G represents the Multi-scale Ball Query operation. Specifically, G groups local features by querying neighbors within multiple radii around the center, allowing the model to aggregate multi-scale context. The final output is a feature vector with a channel dimension of 6, corresponding to the predicted values of length, width, height (LWH), and the center's xyz coordinates.

3.1.2. Encoding Loss Regression

Due to the wide range of values corresponding to length, width and height, the regression values are projected to [0, 1] to improve convergence. Inspired by traditional OD algorithms, the equations for calculating the center offset loss are as follows:

$$\text{diagonal} = \sqrt{L^2 + W^2}, \tag{2}$$

$$dxt = \log(l/L), \quad dyt = \log(w/W), \quad dzt = \log(h/H), \tag{3}$$

$$xt = x/\text{diagonal}, \quad yt = y/\text{diagonal}, \quad zt = z/H, \tag{4}$$

$$\text{Box} = \{xt, yt, zt, dxt, dyt, dzt\}, \tag{5}$$

$$L_{box} = \sum_{i=1}^{n} (y^i - gt^i)/n, \; y^i \in \text{Box}, \; gt^i \in GT, \tag{6}$$

where L, W, and H stand for the average bounding box size. Equations (3) and (4) encode the dimensions and center point position. GT denotes the ground truth values.

3.2. Multi-Scale Feature Blending Upsampling (MFBU) Module

Previous point cloud completion networks heavily rely on the global features extracted during the feature extraction stage. They use these global features to directly generate rough point clouds as a basis for the subsequent completion task. However, in situations where global features or linear interpolation are not ideal, they may lead to a significant reduction in the model's robustness.

To enhance the model's robustness and improve completion results, the MFBU is proposed, as illustrated in Figure 4. This module combines the multi-scale features extracted during the feature extraction stage with the refined features in the refinement stage, enhancing the model's ability to learn local features.

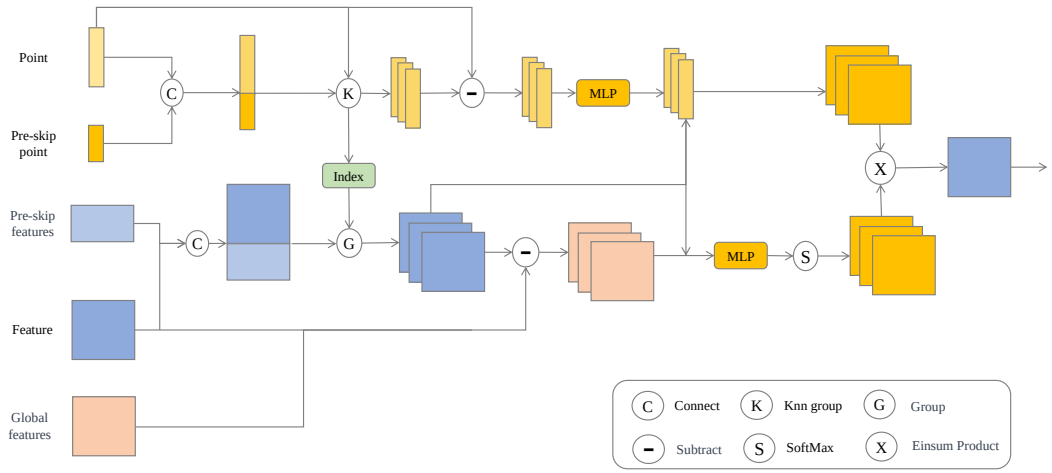


Figure 4. Structure of the Multi-scale Feature Blending Upsampling (MFBU) module.

The module first connects the point cloud P^i_{pre} from the corresponding scale obtained through skip connections with the point cloud P^i_{now} from the refinement stage. This connection results in P^i_{concat}. Subsequently, a multi-scale aggregation operation, as mentioned in PointNet++, is applied to aggregate P^i_{concat} and P^i_{now}, resulting in the aggregated point cloud P^i_{group}, which aligns with the point scale of P^i_{now}. Then, by indexing, the feature vector f^i_{pre} from the feature extraction stage is transformed from (B, C, N_i) scale to f^i_{group} (B, C, N, G)

scale to align the feature vectors, where G denotes the number of neighboring points sampled in each local group. Afterwards, these are fed into the mixed attention module.

$$f_{now}^{(i+1)} = Transformer_i(f_{group}^i, f_{now}^i, f_{global}), \quad i = 0, 1, \ldots, L-1 \tag{7}$$

in which f_{global} represents the final global features extracted during the feature extraction stage, and f_{now}^i represents the feature vectors obtained in the previous stage. The specific computational expressions for the Transformer are as follows:

$$\begin{aligned} f_{temp}^{(i+1)} =& Softmax(MLP(((f_{now}^i + f_{global})/2 + f_{group}^i) + \\ & Cov(P_{now}^i - P_{group}^i))) \odot (f_{group}^i + Cov(P_{now}^i - P_{group}^i)), \\ & i = 0, 1 \ldots L-1 \end{aligned} \tag{8}$$

The cross-attention mechanism blends global features and local features by combining multiple features, allowing it to consider multiscale information more when predicting offsets for upsampled points. This reduces its dependence on global features for generating rough point clouds, increasing the model's robustness and improving its perception of points at different scales.

Subsequently, f_{now}^{i+1} is used to generate offset predictions for the upsampled point cloud, accomplishing the point cloud completion operation.

$$f_{now}^{(i+1)} = [MLP(PS(MLP(f_{temp}^{(i+1)}))), Upsampler(f_{temp}^{(i+1)})], \tag{9}$$

$$P_{now}^{(i+1)} = Upsampler(P_{now}^i) + \tanh(MLP(\mathrm{ReLU}(f_{now}^{(i+1)}))). \tag{10}$$

in which PS represents Point-wise Splitting operation, MLP is Multi-Layer Perception, $upsampler$ denotes linear interpolation, and $relu$ is the activation function. Through these operations, the point number is increased, enabling a more detailed point cloud representation.

3.3. Multi-Task Loss

In this paper, an end-to-end strategy is applied in the training of the network, where the loss function primarily consists of two major components: a center positioning and shape prediction loss, and a traditional multi-stage completion loss. It can be expressed as follows:

$$L_{total} = \lambda_1 L_{CD} + \lambda_2 L_{box} \tag{11}$$

where λ_1 is set to 1 and λ_2 is set to 50, to align the losses of these two components in magnitude, balancing their impacts on the model.

The center positioning and shape prediction loss L_{box} is described in Section 3.1.2. This loss function restricts the parameters of the entire network implicitly with center position and shape information by backpropagating the loss in the branch network during training. The completion loss L_{CD} can be expressed as:

$$L_{CD} = \sum_{i=1}^{3} L_{CD}^i + L_{(CD-match)} \tag{12}$$

where L_{CD}^i represents the result of calculating the CD loss between the point cloud output of each step in the refinement model and the corresponding scale of the complete point cloud. This loss function assists the model in multi-level completion learning, progressively improving the completion results step by step. $L_{CD-match}$ is the loss calculated between the final stage's completion output and the input incomplete point cloud. Through this

loss, the final output shape is constrained within a certain range of the input point cloud, ensuring a certain level of prior knowledge restriction from the original input to the model. This prevents the model from generating random completions detached from the original data.

4. Experiments

4.1. Dataset

To evaluate the proposed method, the ShapeNet dataset [31] is used for training and the KITTI dataset together with a newly collected CUG-Roadside dataset are used for validation.

4.1.1. Training Dataset

The ShapeNet completion dataset is a synthetic dataset generated by simulation, including eight categories such as vehicles and planes. In our study, the primary focus is on the vehicle category. There are 5677 vehicles in the training set, 150 in the test set, and 100 in the validation set. Each vehicle's incomplete point cloud is represented by point clouds from eight different missing angles, with each point cloud containing 500–2000 points, and one complete ground truth point cloud with 16,384 points. Geometric center refers to the actual center of a complete object, reflecting the center of the geometric shape of the object. The geometric center of the vehicles in the original ShapeNet is located at the coordinate origin.

But in the context of completing real-world datasets, the point cloud data to be completed is not necessarily perfectly centered, and the ground truth center often deviates from its geometric center.

In order to ensure that simulated data closely resembles real-world scenes, preprocessing was applied to the ShapeNet dataset. Firstly, the original input point clouds were shifted to the bounding box center of their incomplete parts, establishing it as the origin. Additionally, to facilitate the model in learning position offsets and enhance its robustness, an initial position estimation operation was performed to mitigate training difficulties caused by significant data differences. Due to the characteristics of lidar point clouds, dense points are always on one side of the vehicle. Exploiting this phenomenon, we take the denser side of the point cloud as the reference of the vehicle's boundary. Subsequently, by statistically calculating the average size of vehicles, the estimated reference boundary is aligned to the corresponding boundary of the average-size vehicle, achieving an initialization of the target vehicle's position.This ensures that the initial positions of the training dataset differ from the ground truth origin and are relatively random, providing a more realistic emulation of real-world conditions.

4.1.2. Test Dataset

KITTI dataset: The network's performance is firstly validated utilizing the KITTI dataset, widely acknowledged as one of the standard datasets in the autonomous driving field. 21 point cloud sequences of varying lengths are encompassed by the KITTI dataset, effectively covering many real-world scenarios. In this study, the vehicles are first detected from the point clouds using CasA [60]. Since there were false positives in the object detection results, which cannot be used for completion or for the following validation, the detection results are compared with actual annotations to obtain detection boxes of vehicles that form the basis of the validation dataset. Specifically, detection boxes with their center points within the ground truth bounding boxes and a rotation angle of less than ten degrees are retained. In addition, as the simulated data in the training dataset are artificially scaled to a fixed range, to match the unified scale, vehicles from the ground

truth with lengths in the range of 3.8 m to 4.15 m are selected. Ultimately, a total of 2258 vehicles were acquired as the inference dataset. Subsequently, the corresponding point cloud vehicles were proportionally scaled to align with the scale of the training set, serving as the inference dataset for validation.

CUG-Roadside dataset: An 80-line RoboSense lidar sensor, operating at a frame rate of 10 Hz, was utilized in this study to further validate the network's performance in infrastructure-based traffic flow monitoring. The sensor was positioned at corner of an intersection on the campus of China University of Geosciences, and its absolute location was determined by using Real-Time Kinematic (RTK) positioning. During the data collection process, vehicles were stopped every 2 m, allowing precise measurements of both the front and rear center positions of the vehicle using an RTK device at each stop. Over the course of ten stops, a total of 20 frames of scene data with known vehicle positions were captured. The geometric center positions and vehicle size are calculated from RTK measurements as the reference for the incomplete point clouds. Subsequently, as mentioned in Section 4.1.1, preprocessing was also applied to the roadside dataset to match the training data. This dataset is then directly fed into the completion network without fine-tuning.

4.2. Implementation Details

To ensure fair comparison and reproducibility, we detail the experimental environment and training protocols here. The proposed CFFCNet is implemented using the PyTorch framework (version 1.12.1; Meta Platforms, Inc., Menlo Park, CA, USA). All experiments were conducted on a server equipped with an Intel Xeon Gold 6248 CPU (Intel Corporation, Santa Clara, CA, USA) and two NVIDIA RTX 3090 GPUs (NVIDIA Corporation, Santa Clara, CA, USA). The study conducts end-to-end training for all the models using the same preprocessed ShapeNet dataset. The models are trained for 400 epochs with a total batch size of 64 (32 per GPU). We adopt a warm-up strategy where the learning rate gradually increases from 0 to 0.001 during the first 200 epochs. Subsequently, the learning rate is halved every 50 epochs to stabilize the convergence.

4.3. Comparative Experiments

To demonstrate the performance of our method in real-world scenarios, we conduct comparative experiments with some representative methods. Specifically, we implement GRNet [61], SnowflakeNet [27], AdaPoinTr [57], and ODGNet [59] on both the KITTI dataset and the CUG-Roadside dataset. GRNet is a classic point cloud completion algorithm that serves as a benchmark for comparison with many other point cloud completion methods. AdaPoinTr and ODGNet are recently published point cloud completion algorithms. For a fair comparison, we utilized the same preprocessed ShapeNet dataset for model training.

4.4. Ablation Study

To validate the functionality of the proposed network, two ablation experiments were tested. Firstly, the two main modules, MFBU and BCP, of the CFFCNet are tested to assess their effectiveness. The baseline model with the MFBU module only is referred to as CFFC-MFBU and that with only the BCP module is referred to as CFFC-BCP, and they are compared with the full CFFCNet. These models are all trained using the ShapeNet dataset and inferenced on the KITTI dataset. Additionally, due to the radial scanning mechanism of the lidar, the distribution of target point clouds vary at different distances. To explore the impact of point cloud sparsity and object range on the proposed model, experiments were conducted on the KITTI data at different distance intervals.

4.5. Evaluation Metric

Since the commonly used point cloud completion metrics do not evaluate the positioning accuracy of completed vehicles, to quantitatively assess the performance of different algorithms in vehicle localization and dimension estimation, we employ Mean Absolute Errors (MAEs) as the primary evaluation metric. MAE provides a robust measure of prediction accuracy by computing the average absolute deviation between predicted and ground truth values. Our evaluation framework specifically focuses on MAEs for vehicle center coordinates and bounding box dimensions (length, width, and height), with consistent application across both the KITTI dataset and the CUG-Roadside dataset. To demonstrate the efficiency of point cloud completion, we evaluate the number of parameters (Params) and frame per second (FPS) of CFFCNet, together with other algorithms on KITTI. Params represents the total number of learnable parameters in the model (such as weights and biases), which directly affects the training speed, storage requirements, and inference time. FPS denotes the number of frames processed per second, a key metric to evaluate real-time performance in applications.

5. Results

5.1. Evaluation on the KITTI Dataset

Table 1 presents the completion results based on the detection of the KITTI dataset. The results indicate that when training GRNet, SnowflakeNet, ODGNet and AdaPoinTr networks without modification using the preprocessed ShapeNet dataset as described in the experimental section, clear improvements in dimension estimation and localization are not observed. On the contrary, the results obtained by GRNet and SnowflakeNet show a decline in performance compared with the original detection in terms of both localization and dimension prediction. AdaPoinTr also exhibits inferior performance in dimension prediction, albeit slightly better in terms of localization. ODGNet demonstrates inferior performance in center prediction, while showing slight improvement in height dimension estimation. This suggests that normal shape completion networks could not improve vehicle localization and dimension estimation given the offset between the detection center of incomplete point cloud and the true geometric center.

In contrast, the CFFCNet can effectively improve the accuracy of the vehicle's length, width, height, and center position compared with the original detection box, even when the error is already small. The results show that, on average, CFFCNet reduces the length prediction error by over 20%, and it reduces the height prediction error by nearly 50%. This demonstrates the effectiveness of center-guided point cloud completion in mitigating detection center offsets and enhancing localization and dimension estimation accuracy.

Table 1. Comparison of completion MAEs on the KITTI dataset.

Model	Length (m)	Width (m)	Height (m)	Center (m)
Original detection	0.1103	0.0444	0.1469	0.0955
GRNet	0.2023	0.1542	0.1931	0.1628
SnowflakeNet	0.1855	0.0753	0.0911	0.1538
AdaPoinTr	0.1841	0.1679	0.2653	0.0940
ODGNet	0.1488	0.1343	0.1211	0.1042
CFFCNet (Ours)	0.0850	0.0594	0.0750	0.0928

Table 2 presents a comparison of number of parameters (Params) and Frames Per Second (FPS) runtime between our method and the tested models on the KITTI dataset. While our model slightly increases the number of parameters and reduces the FPS compared with the baseline model, it still delivers real-time performance comparable to other methods.

Table 2. Comparison of model size and speed on the KITTI dataset.

Model	Params (M)	FPS (f/s)
GRNet	76.70	12.31
SnowflakeNet	19.30	16.15
AdaPoinTr	30.88	14.89
ODGNet	11.41	17.38
CFFCNet (Ours)	20.02	14.30

Figure 5 displays the original point cloud detection and completion results from GRNet, SnowflakeNet, AdaPoinTr, ODGNet and CFFCNet. For the original point cloud, although the detection boxes capture the center and dimensional information of the corresponding point clouds, they lack finer details, which are crucial for a comprehensive understanding of the scene.

From the completion results, it can be observed that all comparative networks retained features of incomplete point clouds. By combining the original point cloud information with the vehicle features learned in the network, complete point clouds were inferred. However, the performance of the recent AdaPoinTr network is less refined. By examining the descent of its loss function during training, it can be observed that after approximately 70 epochs, the loss scarcely decreases further. This indicates that even the most advanced completion algorithms still face training difficulties when dealing with completion tasks with uncertain initial positions.

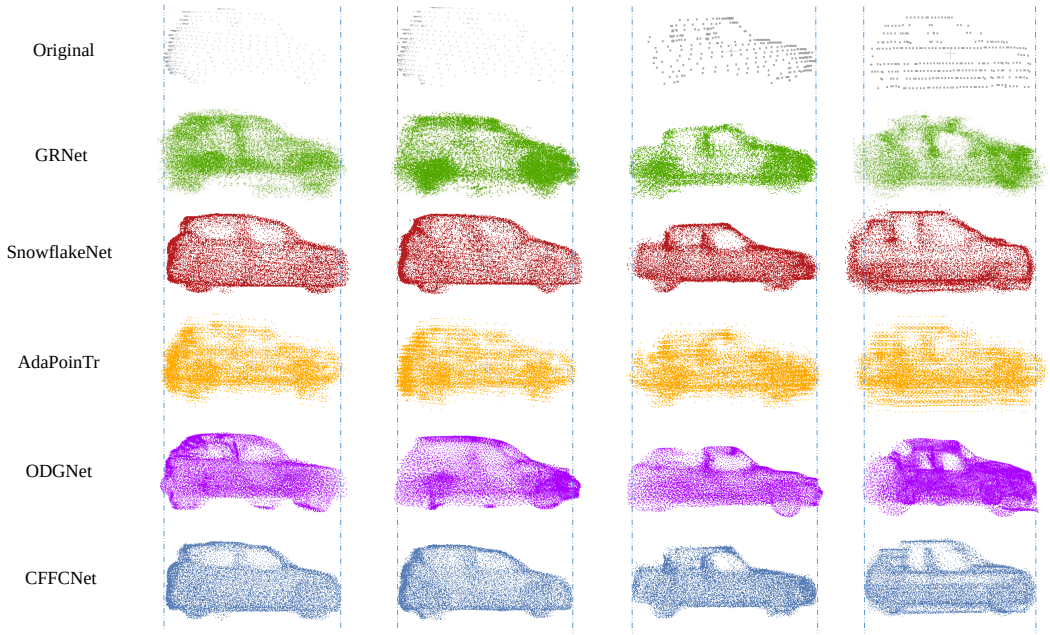


Figure 5. Original point clouds and completion results from GRNet, SnowflakeNet, AdaPoinTr, ODGNet, and CFFCNet on the KITTI dataset.

Upon closer scrutiny, the completion results of GRNet exhibit a larger dimension because of noisy point compared to other methods and inferior performance in completing wheels. It is also evident that the completion predictions of SnowflakeNet result in larger dimensions compared to the ground truth, predicting longer vehicles. Furthermore, it can be noticed that when symmetrized around the origin, the original incomplete point cloud approximates the length of the vehicle completed using the SnowflakeNet, indicating an inherent symmetric operation within the network.

Apparently, such symmetrical operations are not suitable for the task faced in this paper. The results of the AdaPoinTr network indicate that the vehicle point cloud expands evenly towards both ends from the central position, with little significant change in the center. ODGNet performs relatively better in maintaining a low noise level. However, the completion results exhibit a noticeable overestimation in shape. Therefore, its shape completion performs poorly in terms of dimension estimation. In contrast, the proposed network, as substantiated through quantitative and qualitative analyses, accurately acquires information about the center, length, width, and height through shape completion, along with additional geometric details.

5.2. Evaluation on the CUG-Roadside Dataset

Figure 6 displays the roadside lidar point clouds before and after completion, showing decent completion results along the trajectory. It demonstrates that placing lidar sensors beside the road and using CFFCNet for completion can provide enhanced positional and geometric information of the on-road targets.

As shown in Table 3, the completion results obtained with GRNet, SnowflakeNet, AdaPoinTr and ODGNet networks are similar to those achieved on the KITTI dataset, further indicating that these pure shape completion networks are not suited for completion tasks with a focus on dimension and center predictions. The results obtained using our method show that the difference in vehicle length compared to the ground truth is reduced by more than 60%, and the MAE of the center point is reduced by 37.5%. This demonstrates that, under the premise of given vehicle types, the completion model effectively restores point cloud information, proving the practicality of point cloud completion for locating incomplete objects and recovering geometric information. However, due to the characteristics of synthetic training datasets, there are still limitations in the restoration of vehicle width and height, as the dimensions of synthetic vehicles differ from the real data.

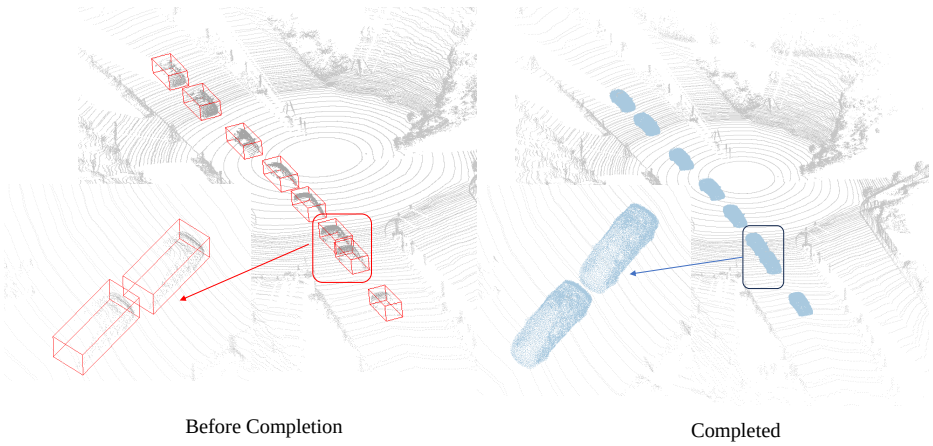


Figure 6. Visualization of vehicles before and after completion using CFFCNet on the CUG-Roadside dataset.

Table 3. Comparison of completion MAEs on the CUG-Roadside dataset.

Model	Length (m)	Width (m)	Height (m)	Center (m)
Original detection	0.134	0.243	0.053	0.084
GRNet	0.225	0.278	0.581	0.197
SnowflakeNet	0.216	0.205	0.431	0.175
AdaPoinTr	0.278	0.160	0.572	0.066
ODGNet	0.138	0.182	0.476	0.104
CFFCNet (Ours)	0.051	0.259	0.303	0.051

5.3. Results of Ablation Experiments

5.3.1. Impact of the BCP and MFBU Modules

According to the quantitative results in Table 4, compared with CFFC-BCP for completion, CFFC-MFBU has lower MAEs in dimensions, while CFFC-BCP has lower MAE in localization. This suggests that the MFBU module is effective in recovering the dimension information of the point cloud, while the BCP module is more effective in predicting the center position of the point cloud.

Subsequently, incorporating both the BCP and MFBU modules into CFFCNet for completion, the model show superior overall performance in terms of both localization and dimension prediction accuracy. This demonstrates that the two modules can effectively complement each other without causing excessive mutual interference.

Table 4. Ablation study of the BCP and MFBU modules on the KITTI dataset.

Model	MFBU	BCP	Length (m)	Width (m)	Height (m)	Center (m)
CFFC-BCP		✓	0.1026	0.0744	0.0866	0.0927
CFFC-MFBU	✓		0.0920	0.0590	0.0710	0.0938
CFFCNet	✓	✓	0.0850	0.0590	0.0750	0.0928

From the qualitative illustration in Figure 7, it can be observed that compared with CFFC-BCP and CFFC-MFBU, CFFCNet produces better completion results for vehicles with smoother surfaces. The completion results of CFFC-BCP lack dimension completeness, while CFFC-MFBU exhibits discrete fluctuations in boundary points in incorrect directions.

However, combining both modules together effectively improves this situation. This further demonstrates that the combination of the BCP and MFBU modules can restore the dimensions of objects and assist in object localization effectively.

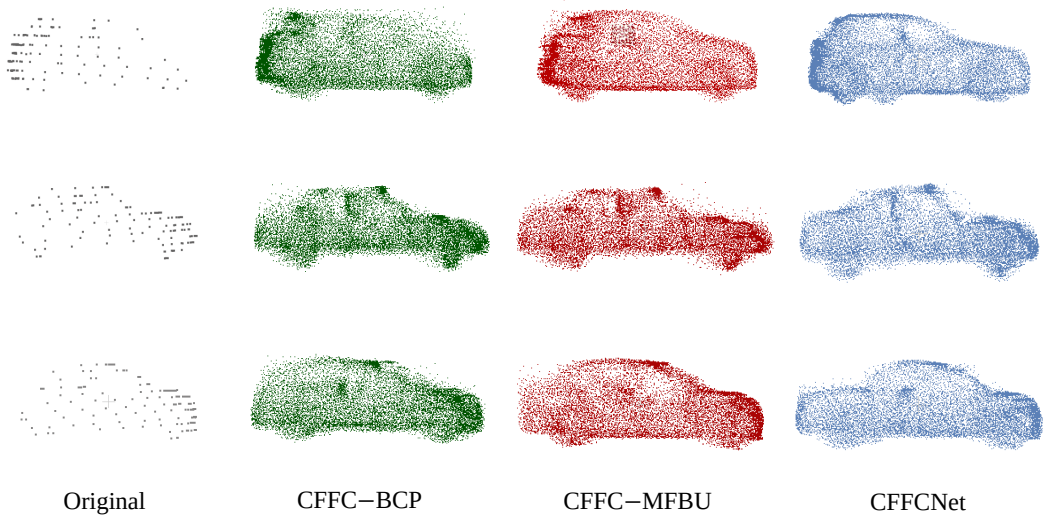


Figure 7. Visualization comparison of CFFC-BCP, CFFC-MFBU, and CFFCNet on the KITTI dataset.

5.3.2. Impact of Distance on Completion Performance

Based on Table 5 and Figure 8, it is observed that the model's completion results are consistently better than the detection results across most metrics. A notable exception is the width estimation at ranges less than ten meters, where the completion MAE (0.057 m) is slightly higher than the original detection (0.048 m). However, it is crucial to note that within this same range, the center localization accuracy improves significantly (MAE decreases from 0.146 m to 0.096 m), demonstrating the model's robustness in localization.

The specific degradation in width prediction is attributed to the sparsity of side-view point clouds caused by the sensor's vertical field-of-view (FOV) limits at close range. As visualized in Figure 8 (<10 m), the laser beams fail to sufficiently capture the vehicle's side profiles (e.g., doors and fenders), resulting in a point cloud that is sparse in these critical areas. Because the key features for determining width are sparse, the network must rely on the remaining denser data, which is predominantly from the vehicle's narrower roof. The model then struggles to infer the correct body boundaries from this incomplete representation, tending to fit the completion to the visible roof edges and thus underestimating the vehicle's true width.

Within the range of 10–25 m, although the point cloud gradually becomes sparse, it maintains a moderate level of sparsity with clear contour information and good viewing angles. Therefore, after completion, accurate positioning and dimensions can be obtained. In the range of 20 to 25 m, both detection and completion results are optimal. Visual inspection of the data reveals that the orientations of the detected vehicles in this range are mostly tangent to the scanning circle of the lidar, indicating more complete retention of vehicle information. This highlights the importance of the viewing angle for collecting point cloud information. Beyond 25 m, the point cloud generally becomes sparse, leading to degradation in detection accuracy. However, the completion can still effectively improve the locations and dimensions of vehicles even in sparse point cloud conditions at nearly 50 m.

Table 5. Performance analysis at different distance ranges on the KITTI dataset.

Range (m)	Num	Orig. Center	Comp. Center	Orig. L	Orig. W	Orig. H	Comp. L	Comp. W	Comp. H
≤10	438	0.146	0.149	0.129	0.048	0.140	0.096	0.057	0.072
10–15	466	0.090	0.088	0.114	0.040	0.142	0.089	0.052	0.045
15–20	454	0.080	0.074	0.115	0.044	0.145	0.089	0.065	0.062
20–25	436	0.079	0.073	0.101	0.043	0.146	0.077	0.062	0.076
25–50	464	0.084	0.082	0.093	0.047	0.161	0.072	0.062	0.120

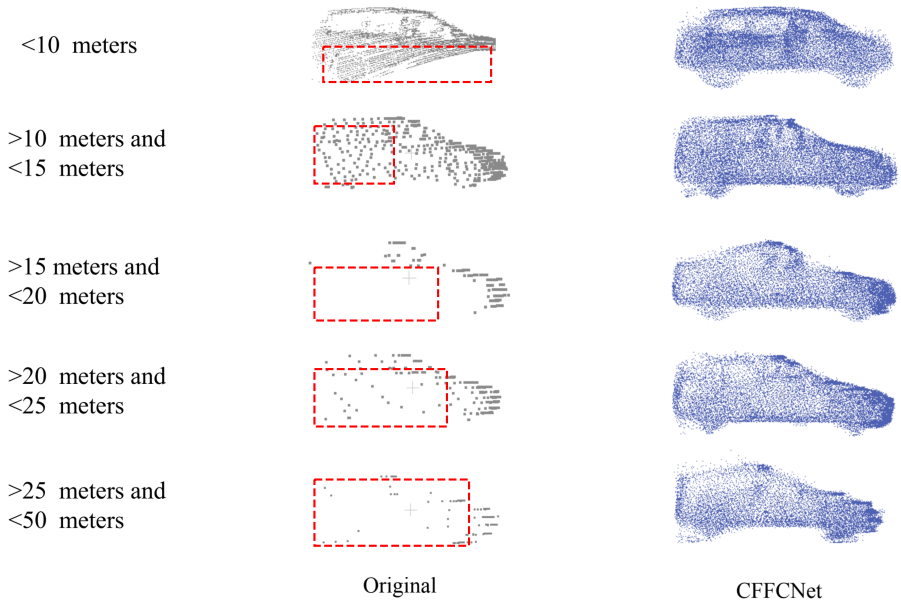


Figure 8. Comparison of point cloud completion on the KITTI dataset at different distances. Red dashed boxes highlight vehicle parts occluded or missing due to limited sensor viewing angles.

6. Discussion

This study pioneers the utilization of a post-detection point cloud completion algorithm to achieve more accurate vehicle localization and dimension estimation. Through experimental validation, the proposed CFFCNet effectively alleviates the issue of poor

completion quality caused by center position offsets in real-world scenarios, thereby encouraging the network to yield more precise geometric predictions.

The superior performance of CFFCNet stems from the synergistic interaction between its two core modules. Specifically, the Multi-scale Feature Blending Upsampling (MFBU) module enhances coordinate representation by fusing locally extracted features from the refinement phase with global features from the extraction phase, leading to more accurate dimension prediction. Simultaneously, the Branch-assisted Center Perception (BCP) module, via implicit training, enables the model to learn center position offsets, significantly improving localization accuracy during inference. The combination of these modules allows CFFCNet to outperform state-of-the-art completion networks in recovering both the shape dimensions and geometric centers of vehicles.

However, detailed analysis reveals that improvements were not uniform across all metrics. For instance, width estimation on the KITTI dataset and height estimation on the CUG-Roadside dataset were suboptimal. We attribute these limitations to two main factors. First, the width degradation on KITTI is caused by the domain gap in scanning patterns: real-world lidar often captures only the roof due to steep viewing angles, whereas the synthetic training data lacks such specific occlusion patterns, making it challenging for the model to infer the full body width. Second, the height error on the CUG-Roadside dataset arises from a data distribution shift. The simulated training dataset scales all objects uniformly, creating a disparity in aspect ratios compared to real-world objects. Remarkably, despite this domain gap and the absence of real-world supervision, the model achieves robust performance, underscoring its strong generalization capability even without fine-tuning.

Despite these advancements, certain constraints remain. First, the method relies on traditional supervised learning, which limits its ability to correct cases where the initial position estimation has significant errors. Second, the model operates under a priori assumptions regarding vehicle size ranges. assumptions regarding vehicle size ranges. While obtaining approximate ranges in real-world scenarios is feasible, this constraint limits the algorithm's flexibility, particularly when significant discrepancies exist between the modeled vehicle sizes in the training set and the actual vehicles of different categories.

Future research should proceed along two primary avenues. First, generative approaches, particularly point cloud diffusion models [62,63], offer a promising solution to the domain gap. To address the limitations identified in our work, these models could be trained to synthesize data replicating challenging real-world scanning patterns, such as the sparse side-view point clouds caused by vertical FOV restrictions at close ranges. This would directly target the root cause of width estimation inaccuracies, though managing the inherent uncertainty of generative models remains a key challenge.

Second, the scarcity of large-scale, annotated real-world datasets remains a fundamental bottleneck. Our findings underscore the need for datasets that capture not just common scenes but also a diverse range of challenging scenarios, specifically including the close-range side-view occlusions we identified as a primary source of error. The ongoing proliferation of infrastructure-based lidar technology presents a clear opportunity to collect such comprehensive data, enabling the construction of more robust and representative benchmarks for real-world completion tasks.

7. Conclusions

In this paper, we introduced a Center-guided Feature Fusion Completion Network (CFFCNet) specifically designed for real-world point cloud completion applications, aiming to enhance accuracy in locating target vehicles and extracting their geometric information. By incorporating the proposed Branch-assisted Center Perception (BCP) module and Multi-scale Feature Blending Upsampling (MFBU) module, the network has demonstrated

significant improvements in both localization and dimension estimation. These advancements were validated through extensive experiments on the KITTI dataset and a real-world dataset collected by a roadside lidar sensor, notably achieving robust generalization without data-specific fine-tuning. These results affirm that our point cloud completion algorithm effectively assists in perceiving accurate object information, offering valuable insights for future perception systems. In future work, we plan to extend the validation of CFFCNet to a broader range of vehicle categories and sizes, further enhancing its applicability and practicality in complex, multi-class traffic environments. All authors have read and agreed to the published version of the manuscript.

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

Funding: This research was funded by the National Natural Science Foundation of China (NSFC), grant number 42201485.

Data Availability Statement: The ShapeNet completion training data are available from ShapeNet (https://shapenet.org/) (accessed on 1 April 2023). The KITTI dataset can be obtained from the KITTI website (https://www.cvlibs.net/datasets/kitti/) (accessed on 1 May 2023). The CUG-Roadside dataset is available from the Geo3DSmart repository (https://github.com/Geo3DSmart/IntDT) (accessed on 23 September 2023). Any additional processed annotations and scripts used to generate the experimental results are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive comments and suggestions that helped improve the quality of this manuscript.

Conflicts of Interest: The authors declare no conflicts 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.

Abbreviations

The following abbreviations are used in this manuscript:

3D	Three-Dimensional
BCP	Branch-assisted Center Perception
CFFCNet	Center-guided Feature Fusion Completion Network
CNN	Convolutional Neural Network
FPS	Frames Per Second
GNSSs	Global Navigation Satellite Systems
MAE	Mean Absolute Error
MFBU	Multi-scale Feature Blending Upsampling
MLP	Multi-Layer Perceptron
OD	Object Detection
RPN	Region Proposal Network
RTK	Real-Time Kinematic
ROI	Region of Interest
SPDs	Snowflake Point Deconvolutions
ROI	Region of Interest
SPDs	Snowflake Point Deconvolutions

References

1. Woo, H.; Kang, E.; Wang, S.; Lee, K.H. A New Segmentation Method for Point Cloud Data. *Int. J. Mach. Tools Manuf.* **2002**, *42*, 167–178. [CrossRef]
2. Guo, M.H.; Cai, J.X.; Liu, Z.N.; Mu, T.J.; Martin, R.R.; Hu, S.M. PCT: Point Cloud Transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [CrossRef]
3. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4338–4364. [CrossRef]
4. Han, X.F.; Jin, J.S.; Wang, M.J.; Jiang, W.; Gao, L.; Xiao, L. A Review of Algorithms for Filtering the 3D Point Cloud. *Signal Process. Image Commun.* **2017**, *57*, 103–112. [CrossRef]
5. Chen, C.; Yao, G.; Liu, L.; Pei, Q.; Song, H.; Dustdar, S. A Cooperative Vehicle-Infrastructure System for Road Hazards Detection With Edge Intelligence. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5186–5198. [CrossRef]
6. Fan, Z.; Liu, H.; He, J.; Sun, Q.; Du, X. A Graph-Based One-Shot Learning Method for Point Cloud Recognition. In *Proceedings of the Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2020; Volume 39, pp. 313–323.
7. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access* **2020**, *8*, 58443–58469. [CrossRef]
8. Radwan, M.; Ohrhallinger, S.; Wimmer, M. Efficient Collision Detection While Rendering Dynamic Point Clouds. In *Proceedings of the Graphics Interface*; AK Peters/CRC Press: Boca Raton, FL, USA, 2014; pp. 25–33.
9. Wei, P.; Cagle, L.; Reza, T.; Ball, J.; Gafford, J. LiDAR and Camera Detection Fusion in a Real-Time Industrial Multi-Sensor Collision Avoidance System. *Electronics* **2018**, *7*, 84. [CrossRef]
10. Zhang, S.; Wang, C.; Lin, L.; Wen, C.; Yang, C.; Zhang, Z.; Li, J. Automated Visual Recognizability Evaluation of Traffic Sign Based on 3D LiDAR Point Clouds. *Remote Sens.* **2019**, *11*, 1453. [CrossRef]
11. Huang, X.; Cheng, X.; Geng, Q.; Cao, B.; Zhou, D.; Wang, P.; Lin, Y.; Yang, R. The ApolloScape Dataset for Autonomous Driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 954–960.
12. Chang, Y.; Xiao, W.; Coifman, B. Using Spatiotemporal Stacks for Precise Vehicle Tracking from Roadside 3D LiDAR Data. *Transp. Res. Part C Emerg. Technol.* **2023**, *154*, 104280. [CrossRef]
13. Zhang, J.; Xiao, W.; Coifman, B.; Mills, J.P. Vehicle Tracking and Speed Estimation From Roadside LiDAR. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 5597–5608. [CrossRef]
14. He, C.; Zeng, H.; Huang, J.; Hua, X.S.; Zhang, L. Structure Aware Single-Stage 3D Object Detection from Point Cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11873–11882.
15. 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 (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4490–4499.
16. Wang, H.; Chen, Z.; Cai, Y.; Chen, L.; Li, Y.; Sotelo, M.A.; Li, Z. Voxel-RCNN-Complex: An Effective 3D Point Cloud Object Detector for Complex Traffic Conditions. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12. [CrossRef]
17. Sun, P.; Tan, M.; Wang, W.; Liu, C.; Xia, F.; Leng, Z.; Anguelov, D. SWFormer: Sparse Window Transformer for 3D Object Detection in Point Clouds. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 426–442.
18. Shi, Z.; Meng, Z.; Xing, Y.; Ma, Y.; Wattenhofer, R. 3D-RETR: End-to-End Single and Multi-View 3D Reconstruction with Transformers. *arXiv* **2021**, arXiv:2110.08861.
19. Wu, J.; Dai, G.; Zhou, W.; Zhu, X.; Wang, Z. Multi-Scale Feature Fusion with Attention Mechanism for Crowded Road Object Detection. *J. Real-Time Image Process.* **2024**, *21*, 29. [CrossRef]
20. Zhang, G.; Junnan, C.; Gao, G.; Li, J.; Hu, X. HEDNet: A Hierarchical Encoder-Decoder Network for 3D Object Detection in Point Clouds. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 12345–12358. [CrossRef]
21. Zamanakos, G.; Tsochatzidis, L.; Amanatiadis, A.; Pratikakis, I. A Comprehensive Survey of LiDAR-Based 3D Object Detection Methods with Deep Learning for Autonomous Driving. *Comput. Graph.* **2021**, *99*, 153–181. [CrossRef]
22. Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; Le, X. PF-Net: Point Fractal Network for 3D Point Cloud Completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 7662–7670.
23. Li, Y.; Li, S.; Du, H.; Chen, L.; Zhang, D.; Li, Y. YOLO-ACN: Focusing on Small Target and Occluded Object Detection. *IEEE Access* **2020**, *8*, 227288–227303. [CrossRef]
24. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
25. 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. [CrossRef]

26. Wang, X.; Ang, M.H., Jr.; Lee, G.H. Cascaded Refinement Network for Point Cloud Completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 790–799.
27. Xiang, P.; Wen, X.; Liu, Y.S.; Cao, Y.P.; Wan, P.; Zheng, W.; Han, Z. SnowflakeNet: Point Cloud Completion by Snowflake Point Deconvolution with Skip-Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 5499–5509.
28. Hu, F.; Chen, H.; Lu, X.; Zhu, Z.; Wang, J.; Wang, W.; Wang, F.L.; Wei, M. SPCNet: Stepwise Point Cloud Completion Network. In *Proceedings of the Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2022; Volume 41, pp. 153–164.
29. Liu, X.; Xu, G.; Xu, K.; Wan, J.; Ma, Y. Point Cloud Completion by Dynamic Transformer with Adaptive Neighbourhood Feature Fusion. *IET Comput. Vis.* **2022**, *16*, 619–631. [CrossRef]
30. Gao, F.; Shi, P.; Wang, J.; Li, W.; Wang, Y.; Yu, J.; Li, Y.; Shuang, F. Dual Feature Fusion Network: A Dual Feature Fusion Network for Point Cloud Completion. *IET Comput. Vis.* **2022**, *16*, 541–555. [CrossRef]
31. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Hao, S.; et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012.
32. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision Meets Robotics: The KITTI Dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
33. Wang, J.; Cui, Y.; Guo, D.; Li, J.; Liu, Q.; Shen, C. PointAttn: You Only Need Attention for Point Cloud Completion. *arXiv* **2022**, arXiv:2203.08485. [CrossRef]
34. 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 (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 770–779.
35. Ding, Z.; Han, X.; Niethammer, M. VoteNet: A Deep Learning Label Fusion Method for Multi-Atlas Segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 202–210.
36. 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 (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11040–11048.
37. Pan, X.; Xia, Z.; Song, S.; Li, L.E.; Huang, G. 3D Object Detection with PointFormer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7463–7472.
38. Sheng, H.; Cai, S.; Liu, Y.; Deng, B.; Huang, J.; Hua, X.S.; Zhao, M.J. Improving 3D Object Detection with Channel-Wise Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 2743–2752.
39. Yang, J.; Li, C.; Zhang, P.; Dai, X.; Xiao, B.; Yuan, L.; Gao, J. Focal Self-Attention for Local-Global Interactions in Vision Transformers. *arXiv* **2021**, arXiv:2107.00641.
40. Li, X.; Yin, J.; Li, W.; Xu, C.; Yang, R.; Shen, J. Di-V2X: Learning Domain-Invariant Representation for Vehicle-Infrastructure Collaborative 3D Object Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–30 March 2024; Volume 38, pp. 3208–3215.
41. Zhang, Y.; Huang, D.; Wang, Y. PC-RGNN: Point Cloud Completion and Graph Neural Network for 3D Object Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 3430–3437.
42. Du, L.; Ye, X.; Tan, X.; Feng, J.; Xu, Z.; Ding, E.; Wen, S. Associate-3Ddet: Perceptual-to-Conceptual Association for 3D Point Cloud Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 13329–13338.
43. Li, Z.; Yao, Y.; Quan, Z.; Yang, W.; Xie, J. SIENet: Spatial Information Enhancement Network for 3D Object Detection from Point Cloud. *arXiv* **2021**, arXiv:2103.15396. [CrossRef]
44. Xu, Q.; Zhong, Y.; Neumann, U. Behind the Curtain: Learning Occluded Shapes for 3D Object Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 22–29 February 2022; Volume 36, pp. 2893–2901.
45. Lu, W.; Zhao, D.; Premebida, C.; Zhang, L.; Zhao, W.; Tian, D. Improving 3D Vulnerable Road User Detection with Point Augmentation. *IEEE Trans. Intell. Veh.* **2023**, *8*, 2783–2795. [CrossRef]
46. Dai, A.; Qi, C.R.; Nießner, M. Shape Completion Using 3D-Encoder-Predictor CNNs and Shape Synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5868–5877.
47. Han, X.; Li, Z.; Huang, H.; Kalogerakis, E.; Yu, Y. High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 85–93.
48. Stutz, D.; Geiger, A. Learning 3D Shape Completion from Laser Scan Data with Weak Supervision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1955–1964.
49. Le, T.; Duan, Y. PointGrid: A Deep Network for 3D Shape Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 9204–9214.

50. Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
51. Yuan, W.; Khot, T.; Held, D.; Mertz, C.; Hebert, M. PCN: Point Completion Network. In Proceedings of the International Conference on 3D Vision (3DV), Piscataway, NJ, USA, 10–12 September 2018; pp. 728–737.
52. Tchapmi, L.P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; Savarese, S. TopNet: Structural Point Cloud Decoder. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 383–392.
53. Wen, X.; Li, T.; Han, Z.; Liu, Y.S. Point Cloud Completion by Skip-Attention Network with Hierarchical Folding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1939–1948.
54. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 206–215.
55. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.S.; Koltun, V. Point Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 16259–16268.
56. Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; Zhou, J. PoinTr: Diverse Point Cloud Completion with Geometry-Aware Transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 12498–12507.
57. Yu, X.; Rao, Y.; Wang, Z.; Lu, J.; Zhou, J. AdaPoinTr: Diverse Point Cloud Completion with Adaptive Geometry-Aware Transformers. *arXiv* **2023**, arXiv:2301.04545. [CrossRef]
58. Wen, X.; Xiang, P.; Han, Z.; Cao, Y.P.; Wan, P.; Zheng, W.; Liu, Y.S. PMP-Net++: Point Cloud Completion by Transformer-Enhanced Multi-Step Point Moving Paths. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 852–867. [CrossRef]
59. Cai, P.; Scott, D.; Li, X.; Wang, S. Orthogonal Dictionary Guided Shape Completion Network for Point Cloud. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–30 March 2024; Volume 38, pp. 864–872.
60. Wu, H.; Deng, J.; Wen, C.; Li, X.; Wang, C.; Li, J. CasA: A Cascade Attention Network for 3D Object Detection From LiDAR Point Clouds. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [CrossRef]
61. Xie, H.; Yao, H.; Zhou, S.; Mao, J.; Zhang, S.; Sun, W. GRNet: Gridding Residual Network for Dense Point Cloud Completion. In Proceedings of the European Conference on Computer Vision (ECCV), Virtual Event, 23–28 August 2020; pp. 365–381.
62. Zhou, L.; Du, Y.; Wu, J. 3D Shape Generation and Completion Through Point-Voxel Diffusion. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 5826–5835.
63. Romanelis, I.; Fotis, V.; Kalogeras, A.; Alexakos, C.; Moustakas, K.; Munteanu, A. Efficient and Scalable Point Cloud Generation with Sparse Point-Voxel Diffusion Models. *arXiv* **2024**, arXiv:2408.06145. [CrossRef]

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.