IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

PCPNet: An Efficient and Semantic-Enhanced Transformer Network for Point Cloud Prediction

Zhen Luo Junyi Ma Zijie Zhou Guangming Xiong*

***Abstract*— The ability to predict future structure features of environments based on past perception information is extremely needed by autonomous vehicles, which helps to make the following decision-making and path planning more reasonable. Recently, point cloud prediction (PCP) is utilized to predict and describe future environmental structures by the point cloud form. In this letter, we propose a novel efficient Transformer-based network to predict the future LiDAR point clouds exploiting the past point cloud sequences. We also design a semantic auxiliary training strategy to make the predicted LiDAR point cloud sequence semantically similar to the ground truth and thus improves the significance of the deployment for more tasks in real-vehicle applications. Our approach is completely self-supervised, which means it does not require any manual labeling and has a solid generalization ability toward different environments. The experimental results show that our method outperforms the state-of-the-art PCP methods on the prediction results and semantic similarity, and has a good real-time performance. Our open-source code and pre-trained models are available at https://github.com/Blurryface0814/PCPNet.**

***Index Terms*— point cloud prediction, semantic auxiliary training, self-supervised learning.**

I. Introduction

Sequential 3D point clouds can be used to accomplish multiple complex tasks for autonomous vehicles such as simultaneous localization and mapping (SLAM) [1], [2], place recognition [3], [4], object detection [5], [6], and semantic segmentation [7], [8]. Recently, exploiting the past 3D point cloud sequence to predict the future 3D point cloud sequence, also known as point cloud prediction (PCP), has attracted more attention in the field of point cloud processing [9], [10], [11], [12], [13], [14], [15]. The predicted point clouds can be directly utilized by the existing point cloud processing methods to further realize future object detection and semantic segmentation [12], [13]. Therefore, PCP methods deployed on autonomous vehicles can greatly improve the ability to perceive future driving conditions, thus leading to more reasonable decision-making and path planning in the following driving strategy. The point cloud sequence used by PCP methods is ordered in the time dimension but is unordered in the space dimension within each observation. Compared to the existing video prediction methods [16], [17], [18], 3D point cloud prediction needs to use LiDAR data with both ordered and unordered features to solve the sequence-to-sequence problem. In addition, the sparsity of LiDAR point clouds further increases the difficulty of this task since it is hard to capture the current structure information from discrete laser points and then predict the future point clouds.

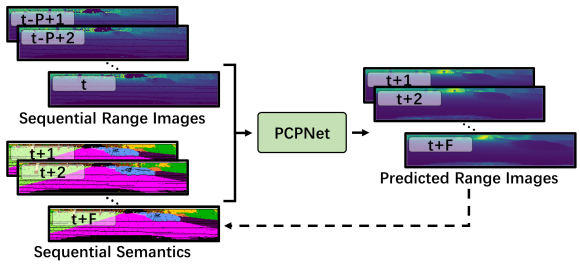

Fig. 1: PCPNet predicts future F range images based on the given past P sequential range images. The semantic information in the sequential range images is extracted for training, making the outputs of PCPNet closer to the ground truth in semantics.

In this paper, we propose an efficient point cloud prediction network named PCPNet, as shown in Fig. 1. PCPNet exploits Transformer to capture the inner correlation within each sequential LiDAR observations. Compared with the previous works that directly apply Transformer to point cloud features [19], [20], we first convert the point cloud to the range image, and then compress it along the height and width dimension respectively to generate the sentence-like features for the following Transformer. In contrast to sparse point clouds, the orderly arrangement of pixels in the range image makes the self-attention mechanism of Transformer works better. To further make the predicted point clouds semantically similar to ground truth, we also design an auxiliary training strategy that utilizes the semantics of the predicted point clouds to enhance the prediction performance. The devised semantic-based loss function helps to increase the semantic similarities between the predicted point clouds and the ground truth point clouds, leading to more significant information for other algorithms deployed on autonomous vehicles such as high-level point cloud processing and trajectory prediction. To the best of our knowledge, this is the first work that introduces Transformer and semantic enhancement into point cloud prediction. In addition, the overall system of PCPNet is entirely self-supervised since the input point clouds and ground truth ones are both from real-time sequential LiDAR observations. Note that we only use a pre-trained semantic segmentation network to generate semantic labels online for auxiliary training. This makes our proposed approach collect training data in any driving conditions automatically without intensive manual labeling.

Z. Luo, J. Ma, Z. Zhou and Guangming Xiong are with Beijing Institute of Technology.

*corresponding author email: xiongguangming@bit.edu.cn

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

To validate the good prediction performance and solid generalization ability of our proposed method, we conduct several experiments on publicly available datasets to compare our approach with the existing state-of-the-art PCP baselines. We also provide an ablation study to demonstrate that our proposed network structure is reasonable and efficient. Besides, we design an experiment to validate the effectiveness of the proposed semantic auxiliary training strategy.

Our contributions can be summarized as follows:

- A Transformer-based neural network with encoder-decoder architecture named PCPNet is proposed, which achieves state-of-the-art performance on point cloud prediction.
- A lightweight semantic segmentation network is integrated to provide semantic labels for auxiliary training, which increases the semantic accuracy of the predicted point clouds.
- The overall system is trained in a self-supervised manner to improve the generalization ability and avoid labor-intensive labeling in real-vehicle applications.

II. Related Work

To capture future features of the environments using past information, video prediction [16], [17], [18] has been well investigated in the field of computer vision. In contrast, point cloud prediction is a novel topic in robotics and only a few studies have been working on it. PointRNN proposed by Fan et al. [9] adopts a point-based spatiotemporally-local correlation to aggregate point features and states according to point coordinates to model and predict point cloud sequences. Zhang et al. [10] design a dynamic convolution operator which allows performing convolution operations directly over point clouds. Deng et al. [11] propose a method based on FlowNet3D and Dynamic Graph CNN to predict future LiDAR frames. They use the point-based feature extractor and furthest point sampling to generate end-to-end architectures that operate directly on point clouds. Lu et al. [12] propose a motion-based neural network named MoNet which integrates the motion features between two consecutive point clouds into the RNN prediction pipeline. Different from the above-mentioned methods that only use raw point clouds as input, several existing methods also utilize sequential range images as input to further improve the operation efficiency. SPFNet by Weng et al. [13] uses a shared encoder and an LSTM to extract features from every past scan and model temporal dynamics respectively. These features are then fed to a shared decoder to predict future scans. Based on SPFNet, Weng et al. [14] propose a stochastic SPFNet named S2Net, which can sample sequences of latent variables to tackle future uncertainty. Mersch et al. [15] propose a 3D range-image-based method for predicting future point clouds. They project past point cloud sequences as 2D range images and then concatenate them to generate 3D tensors. A 3D spatio-temporal CNN with encoder-decoder architecture is designed to predict future point clouds based on these tensors.

The main challenge in point cloud prediction is how to extract significant spatio-temporal features from the sparse and disordered point cloud sequences. The existing trajectory tracking and moving object segmentation approaches have designed diverse networks to extract spatio-temporal features in a learning-based manner. Alahi et al. [21] use LSTM to extract spatio-temporal information to predict pedestrian trajectories. Huang et al. [22] model the interaction of tracking objects at each time step by a graph neural network. Compared to the RNN-based methods, more and more CNN-based methods have been proposed to achieve state-of-the-art performance in recent years. For example, Chen et al. [23] design an encoder-decoder network with combined residual images as input to improve the performance of moving object segmentation. Sun et al. [24] use a range-image-based dual-branch structure to process spatial and temporal information respectively, and then combine them with motion-guided attention modules. In recent years, Transformer [25] became a more trendy way to extract spatial features within one single LiDAR scan, but few works use it to extract temporal features from sequential LiDAR observations. Fan et al. [20] apply a point-based Transformer for spatio-temporal modeling of small-scale point cloud video. Ma et al. [4] utilize sequential range images as network input and extract significant features using Transformer to recognize previously seen places.

In this paper, we propose PCPNet which first uses Transformer to aggregate spatial and temporal information from sequential range images to forecast point clouds. Besides, compared to the existing PCP methods which only focus on low-level structure features, PCPNet also uses high-level semantic information for auxiliary training to further improve the prediction performance.

III. Our Approach

A. Overall Architecture

A point cloud sequence with the length of T can be described as $S = \{S_1, S_2, ..., S_t, ..., S_T\}$, where S_t represents the point cloud frame at time t. Further, $S_t = \{p_i^t \mid i = 1, 2, ..., N_t\}$ contains N_t unordered points, where $p_i^t \in \mathbb{R}^3$ represents a three-dimensional coordinate vector. Point cloud prediction is to forecast the future point cloud sequence $\{S_{t+1}, S_{t+2}, ..., S_{t+F}\}$ with the length of F based on the given past point cloud sequence $\{S_{t-P+1}, S_{t-P+2}, ..., S_t\}$ with the length of P.

To solve this problem, we propose a Transformer-based network named PCPNet. Considering the advantages of Transformer in processing sequential data, we first convert 3D LiDAR point clouds to 2D range images by spherical projection to obtain dense and ordered input data. Specifically, we project a laser point $p_i = (x, y, z) \in \mathbb{R}^3$ to spherical coordinates and finally to image coordinates $(u, v) \in \mathbb{R}^2$ via a mapping $\Pi: \mathbb{R}^3 \mapsto \mathbb{R}^2$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\left[1 - \arctan(y, x)\pi^{-1}\right] w \\ \left[1 - \left(\arcsin(zr^{-1}) + \mathrm{f}_{\mathrm{up}}\right)\mathrm{f}^{-1}\right] h \end{pmatrix}, \quad (1)$$

where (h, w) represents the height and width of the range image, $\mathrm{f} = \mathrm{f}_{\mathrm{up}} + \mathrm{f}_{\mathrm{down}}$ is the vertical field-of-view of the sensor, and $r = \|p_i\|_2$ represents the range of each laser point, which is the distance to the sensor origin. We

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

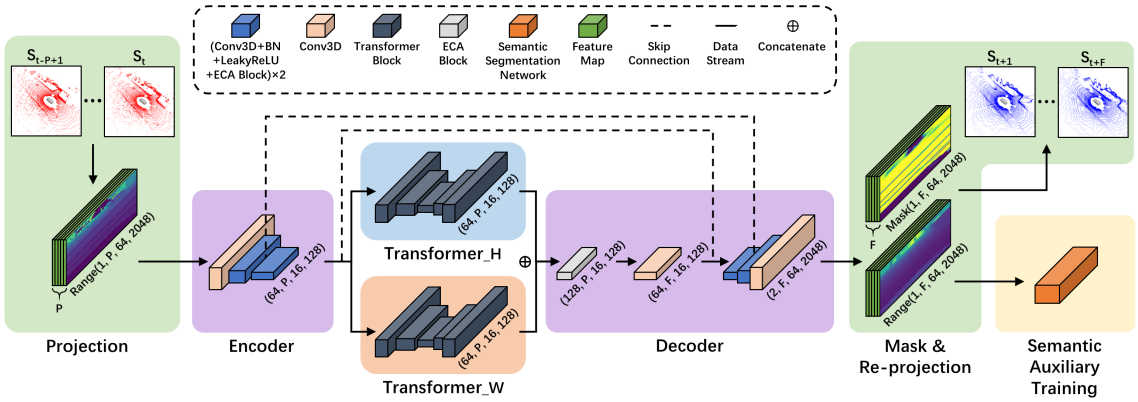

Fig. 2: The overall architecture of PCPNet. The numbers in the bracket below each cube represent the size of the feature map output from this layer. The input range images are first downsampled and compressed along the height and width dimensions respectively to generate the sentence-like features for the following Transformer blocks. The features are then combined and upsampled to the predicted range images and mask images. Semantic auxiliary training is used to enhance the practical value of point cloud prediction.

assume that the point clouds are located in the current local coordinate frame of the LiDAR sensor. If no laser point exists for a corresponding pixel, we set $r = 0$, and if a pixel corresponds to more than one laser point, the nearest one is retained. In addition, the re-projection $\Pi' : \mathbb{R}^2 \mapsto \mathbb{R}^3$ of a pixel on a range image can be described as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r\cos(\theta)\cos(\gamma) \\ r\cos(\theta)\sin(\gamma) \\ r\sin(\theta) \end{pmatrix}, \tag{2}$$

where $\theta = \arctan(y, x)$ represents the pitch angle and $\gamma = \arcsin(zr^{-1})$ represents the yaw angle.

The overall architecture of PCPNet is shown in Fig. 2. Following the operation of Mersch et al. [15], we concatenate P range images with height H_p and width W_p along the time dimension to get a 4D input tensor (C_p, P, H_p, W_p), where C_p is the number of channels. In this work, H_p is set to 64 and W_p is set to 2048. Since we only use range values as input, $C_p = 1$ holds. The encoder uses the combination of 3D convolution, 3D batch normalization, LeakyReLU, and ECA block [26] to compress the input tensor in the height and width dimensions while maintaining the size of the time dimension. ECA is a channel attention mechanism that we use to enable the network to automatically adjust the weight of each channel. The output tensor from the encoder is then fed into both Transformer_H block and Transformer_W block for self-attention. The outputs of the two branches are combined by an ECA block and a 3D convolution, ultimately being fed to the decoder that mirrors the encoder. The final output tensor size of the network is $(2, F, H_f, W_f)$, including a range image sequence and a mask image sequence both with size $(1, F, H_f, W_f)$. As done by [13], [15], the mask image sequence contains a probability for each range image pixel to be a valid point for re-projection. We combine and re-project the two image sequences to 3D coordinate system to obtain the final predicted point cloud sequence using Eq. (2). Note that we use circular padding to maintain spatial consistency on the horizontal borders of the range images in PCPNet. We also use the skip connection to maintain the details of the prediction, which is also shown in Fig. 2.

B. Transformer Block

The use of Transformer allows the network to notice the correlation between different locations throughout the input sentence [25], [3]. An important problem to be solved in applying Transformer to point clouds is to generate the input features with the sentence-like form (C_l, L), where C_l represents the number of channels and L is the length of the sentence. Therefore, we design the Transformer block which is shown in Fig. 3. The tensors generated by the encoder are fed to the Transformer_H block and the Transformer_W block respectively. The down-sampling module maintains one dimension in height or width while compressing the other to the size of 1. Then the compressed tensors are concatenated along the time dimension to obtain a continuous image sentence with a larger width. The image sentence is input into Transformer, then upsampled to the previous size through the mirror operation.

Our method is well suited for Transformer to extract spatio-temporal features. Taking the Transformer_W block as an example, the height dimension of the feature volume is compressed to size 1, and thus the channel dimension contains more distinct information. Each column of the image sentence aggregates all the information of a width slice. The concatenated image sentence can be expressed as $\{w_i \mid i = 1, 2, ..., P \cdot W_l\}$, where w_i represents the word vector at the width i and W_l represents the length of each of the P image sentences. In detail, the concatenated image sentence contains the temporal position and the spatial position of each vertical slice. Therefore, Transformer can capture the spatio-temporal correlation between vertical slices in the image sentence, and further improves the performance of point cloud prediction.

C. Semantic Auxiliary Training

In real vehicle applications, the predicted point cloud sequences can be used to serve the following downstream tasks such as future scene understanding [27] and object detection [6]. Therefore, the predicted point cloud needs to be closer to the ground truth in semantics to improve its practical value. Motivated by this, we propose semantic

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

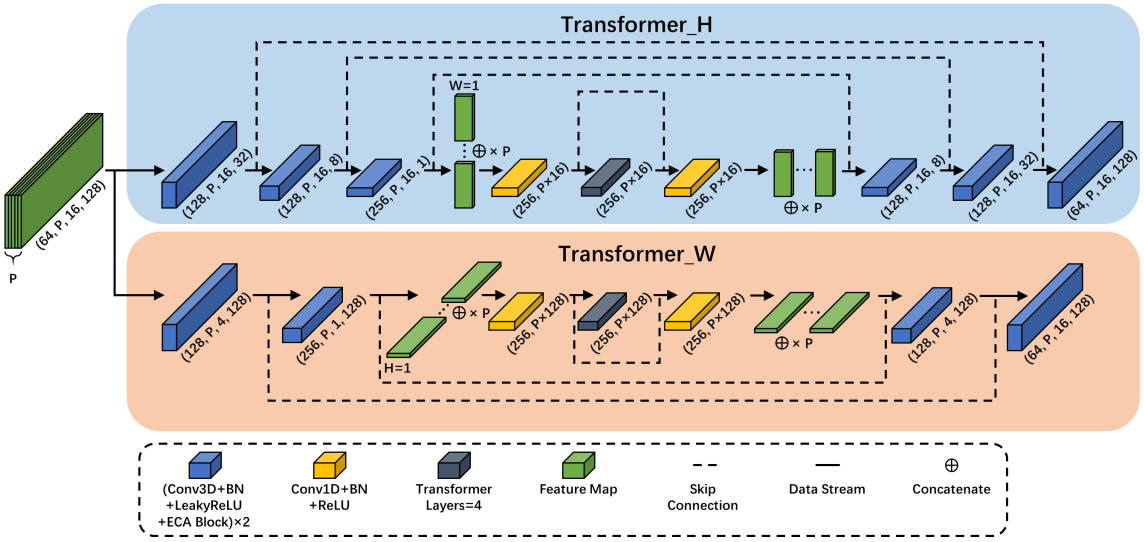

Fig. 3: The architecture of Transformer block. The feature volumes from the sequential range images are concatenated into a longer image sentence before Transformer, and then re-concatenated into the previous shape after Transformer.

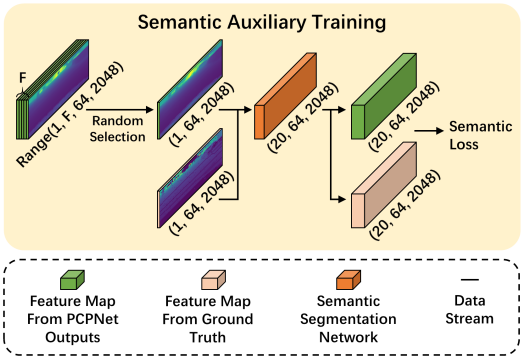

Fig. 4: The details of semantic auxiliary training. The semantic loss is calculated directly on the multi-channel feature volume output by the segmentation network.

auxiliary training to enhance the performance of PCPNet. As shown in Fig. 4, we randomly select one of the t range images from the predicted sequence generated by PCPNet for one forward propagation. The selected range image is then fed to the pre-trained RangeNet++ [7], a lightweight semantic segmentation network together with the corresponding ground truth range image. The output tensor of the segmentation network is (C_s, H_s, W_s), where C_s represents the number of classes in semantics. We calculate $L1$ loss between the semantic map from the output of PCPNet and the one from the ground truth to obtain the semantic loss. Note that the random selection in one prediction aims to reduce the calculation costs and inference time. Besides, we use the semantic output from ground truth range images as the labels to calculate the semantic loss. This helps to achieve the self-supervised training process since no manually annotated labels of semantic segmentation are needed.

We do not perform the argmax operation on channels and calculate the cross-entropy loss as most semantic segmentation methods do [28], [7], [8] because the output of the segmentation network is probability distributions for different classes, and even the wrong semantic results contain features that are worth learning. $L1$ loss can make the semantic probability distribution between the output and the ground truth closer, rather than simply making the output close to a certain class.

D. Loss Function

We use a combination of multiple losses including the average range loss $\mathcal{L}_R$, the average mask loss $\mathcal{L}_M$, the average semantic loss $\mathcal{L}_S$, and the chamfer distance loss $\mathcal{L}_C$ to train our network. The average range loss $\mathcal{L}_R$ between the predicted range images $\hat{r}_{c,i,j} \in \mathbb{R}^{F\times H_f\times W_f}$ and the ground truth range images $r_{c,i,j} \in \mathbb{R}^{F\times H_f\times W_f}$ can be formulated as

$$\mathcal{L}_R = \frac{1}{F\times H_f\times W_f}\sum_{c,i,j}\|\hat{r}_{c,i,j} - r_{c,i,j}\|_1, \tag{3}$$

where $\|\bullet\|_1$ represents $L1$ norm. Since there are invalid points on the ground truth range images, we only calculate $\mathcal{L}_R$ using the valid points. We further calculate the binary cross-entropy loss between the mask images $\hat{m}_{c,i,j} \in \mathbb{R}^{F\times H_f\times W_f}$ and the ground truth mask images $m_{c,i,j} \in \mathbb{R}^{F\times H_f\times W_f}$ to get the average mask loss $\mathcal{L}_M$ by

$$\mathcal{L}_M = \frac{1}{F\times H_f\times W_f}\sum_{c,i,j}[-m_{c,i,j}\log\hat{m}_{c,i,j} - (1-m_{c,i,j})\log(1-\hat{m}_{c,i,j})], \tag{4}$$

where $\hat{m}_{c,i,j}$ is the predicted probability to demonstrate whether (c,i,j) is a valid point. $m_{c,i,j}=1$ represents that the ground truth point (c,i,j) is valid and $m_{c,i,j}=0$ otherwise. The average semantic loss $\mathcal{L}_S$ between a predicted multi-channel semantic image $\hat{s}_{c,i,j} \in \mathbb{R}^{C_s\times H_s\times W_s}$ which is randomly selected and the corresponding ground truth multi-channel semantic map $s_{c,i,j} \in \mathbb{R}^{C_s\times H_s\times W_s}$ is given by

$$\mathcal{L}_S = \frac{1}{C_s\times H_s\times W_s}\sum_{c,i,j}\|\hat{s}_{c,i,j} - s_{c,i,j}\|_1, \tag{5}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

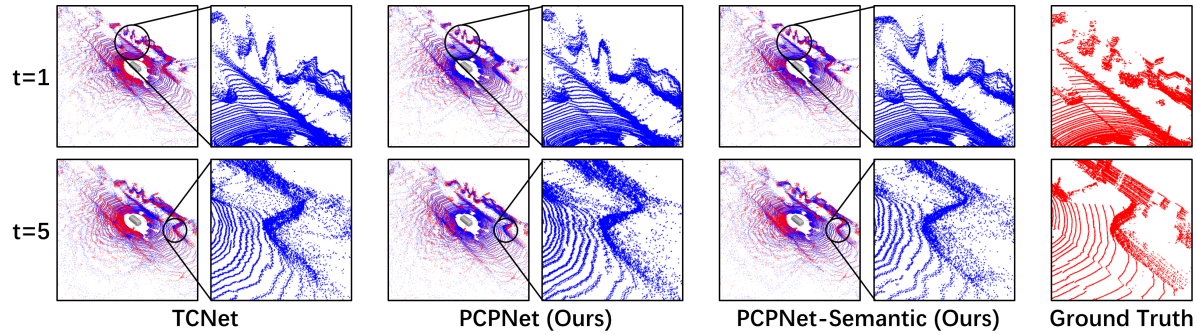

Fig. 5: Qualitative comparison conducted on sequence 08 of the KITTI dataset. The predicted points (blue) and the ground truth points (red) are combined for better visual comparison. The upper row shows the predicted step $t = 1$ and the lower row the predicted step $t = 5$. Local structures in large-scale point clouds are circled and enlarged to better observe local details.

which is also calculated only at the valid points just like $\mathcal{L}_R$.

Besides the $\mathcal{L}_R$, $\mathcal{L}_M$, and $\mathcal{L}_S$ which are range-image-based losses, we also calculate the point-based loss to further improve the accuracy of point cloud prediction. Following [9], [12], [15], we use the Chamfer Distance [29] to measure the difference between the predicted point cloud $\hat{S} = \{\hat{p}_i \in \mathbb{R}^3 \mid i = 1, 2, ..., N\}$ reprojected from the masked range image and the ground truth point cloud $S = \{p_i \in \mathbb{R}^3 \mid i = 1, 2, ..., M\}$. The chamfer distance loss $\mathcal{L}_C$ is calculated by

$$\mathcal{L}_C = \frac{1}{N} \sum_{\hat{p} \in \hat{S}} \min_{p \in S} \|\hat{p} - p\|_2^2 + \frac{1}{M} \sum_{p \in S} \min_{\hat{p} \in \hat{S}} \|\hat{p} - p\|_2^2, \quad (6)$$

where $\| \bullet \|_2$ represents $L2$ norm. Therefore, the total loss function that we ultimately use is

$$\mathcal{L} = \mathcal{L}_R + \mathcal{L}_M + \alpha_S \mathcal{L}_S + \alpha_C \mathcal{L}_C, \quad (7)$$

where α_S and α_C represent the weight coefficients to demonstrate whether using $\mathcal{L}_S$ and $\mathcal{L}_C$ for training respectively.

IV. Experiments

A. Experimental settings

We train our proposed PCPNet in a self-supervised manner since no manually annotated labels are needed. The length of the past point cloud sequence P and the future point cloud sequence F are both set to 5, which means the time step $t \in [1, 5]$. Each range image in the input and output sequences has the size 64×2048, and the probability threshold for each mask image is set to 0.5 to mask out possible invalid points in the predicted range images. We use sequences 00 to 05 of KITTI Odometry dataset [30] for training, sequences 06 and 07 for validation, and sequences 08 to 10 for testing. During the training process, we use the Adam optimizer [31] with default parameters and set the initial learning rate to 10^{-3} with an exponential decay weight 0.99. Semantic auxiliary training exploits RangeNet++ which is pre-trained on SemanticKITTI [32] as the semantic segmentation network. Our proposed network and all the baseline models are trained for 50 epochs. During the experiments, α_S is set to 1.0 for the model using semantic auxiliary training and 0.0 otherwise. In addition, we set $\alpha_C = 0.0$ for pre-training and $\alpha_C = 1.0$ for fine-tuning to realize better performance and less training time. We conduct all the following experiments on a system with an Intel i7-10875H CPU and an Nvidia RTX 3070 GPU.

B. Qualitative Evaluation

We first make a qualitative comparison between PCPNet and the range-image-based baseline method TCNet [15] to show the superiority of our method intuitively, and the results are shown in Fig. 5. PCPNet does not use semantic auxiliary training while PCPNet-Semantic does throughout the whole training process. In the predicted sequence, the $t = 1$ frame shows that both PCPNet and PCPNet-Semantic outperform TCNet in terms of the structural details of the surrounding environments. From the parts enclosed and magnified by the black circles in Fig. 5, the walls predicted by PCPNet and PCPNet-Semantic are closer to the ground truth. The distribution of point clouds predicted by PCPNet is more regular and less fluctuated compared to the results from TCNet. Besides, our proposed methods can also predict the shape of the car better than TCNet, and the PCPNet-Semantic forecasts best because our proposed semantic auxiliary training further helps to maintain the structure information in the semantic level.

With the increase of time steps, the prediction of point clouds becomes more difficult since there is a larger time gap between the current perception and the predicted one. At the $t = 5$ frame, the prediction of TCNet at wall corners has an obvious deviation compared to the results of PCPNet. PCPNet-Semantic outperforms all the baseline methods overall, which indicates the use of semantic information enhances the ability to predict the future shape of the perceived objects with large time gaps.

C. Quantitative Evaluation

We quantitatively compared our methods with multiple baselines including PointLSTM [9], MoNet-LSTM [12], MoNet-GRU [12], and TCNet [15] on the KITTI test set, and the results support that our proposed PCPNet achieves the state-of-the-art performance on point cloud prediction. Here we use Chamfer distance $[m^2]$ as the metric to measure the difference between the predicted point cloud and the ground truth point cloud. PointLSTM, MoNet-LSTM, and MoNet-GRU are all point-based methods, so they use down-sampled point clouds to accelerate calculation, while TCNet and our proposed PCPNet are range-image-based methods and can predict full-scale point clouds. For the point-based methods, we follow the operation reported by their authors

TABLE I: Chamfer Distance Results on the KITTI Test Set

	Sampled Point Clouds						Full-scale Point Clouds		
Prediction Step	PointLSTM [9]	MoNet-LSTM [12]	MoNet-GRU [12]	TCNet [15]	**PCPNet (Ours)**	**PCPNet-Semantic (Ours)**	TCNet [15]	**PCPNet (Ours)**	**PCPNet-Semantic (Ours)**
1	0.317	0.286	0.278	0.290	0.285	**0.280**	0.256	0.252	**0.247**
2	0.507	0.412	0.392	0.357	0.341	**0.340**	0.314	0.302	**0.298**
3	0.750	0.567	0.543	0.441	**0.411**	0.412	0.387	0.363	**0.361**
4	0.982	0.719	0.681	0.522	**0.492**	0.495	0.459	**0.436**	**0.436**
5	1.210	0.874	0.830	0.629	**0.580**	0.601	0.554	**0.514**	0.530
Mean	0.753	0.572	0.545	0.448	**0.422**	0.426	0.394	**0.373**	0.374

to downsample the input point clouds to 16384 points to save the computing cost, while the more lightweight architectures allow the range-image-based methods PCPNet and TCNet to maintain a very low computing cost on more laser points. Since our method predicts range images, direct down-sampling for the reprojected point clouds significantly affects the final prediction results. Therefore, we follow the operation of TCNet [15] to downsample the input point clouds to 65536 points to compare with the point-based baseline methods. The results of the quantitative evaluation are shown in Tab. I. In terms of the sampled point clouds, our methods produce more stable predictions than other baselines as the prediction steps increase, which is reflected in the smaller chamfer distance for the larger prediction steps. Even affected by the down-sampling operation, our methods perform better on average chamfer distance at every single step. We further provide an evaluation on full-scale point clouds in Tab. I. It can be seen that our methods also perform better than the other range-image-based method TCNet at step $1 \sim 5$ on full-scale point clouds. In general, PCPNet-Semantic only outperforms PCPNet on chamfer distance with smaller time gaps, but can forecast the future structure information with larger time gaps which further support the clarification in Sec. IV-B.

D. Generalization Study

To prove the solid generalizability of our proposed method, we also conduct a generalization study on the nuScenes dataset [33] with the training strategy similar to the experiments on the KITTI dataset. We use scenes 00 to 69 for training (70 in total), scenes 70 and 84 for validation (15 in total), and scenes 85 to 99 for testing (15 in total). Since our proposed method is self-supervised once semantic labels are available, the semantic segmentation network utilized in the auxiliary training strategy affects the generalization ability of PCPNet most. In this experiment, PCPNet is trained in a self-supervised manner on the nuScenes dataset, where the semantic labels are provided by RangeNet++ which is still pre-trained on the SemanticKITTI dataset. Since the KITTI dataset contains point clouds collected by a 64-beam LiDAR while the nuScenes dataset uses a 32-beam one, we re-train RangeNet++ on the SemanticKITTI dataset with range images of size 32×1024 to adapt to the input data form of nuScenes dataset. As shown in Tab. II, all losses of PCPNet-Semantic are less than TCNet on the nuScenes dataset, which supports that our method generalizes well into other driving environments even with semantic auxiliary training. Due to the use of a 32-beam LiDAR in the nuScenes dataset, the amount of point cloud data decreases significantly, which results in larger chamfer distance of both PCPNet and TCNet compared to the evaluation on the KITTI dataset.

TABLE II: Generalization Study on the nuScenes Test Set

Approach	$\mathcal{L}_R$	$\mathcal{L}_M$	$\mathcal{L}_S$	$\mathcal{L}_C$
TCNet	0.719	0.240	0.043	1.389
PCPNet-Semantic	**0.704**	**0.236**	**0.034**	**1.360**

TABLE III: Ablation Study on the KITTI Validation Set

Approach	$\mathcal{L}_R$	$\mathcal{L}_M$	Runtime (ms)
PCPNet-W	0.784	0.296	**5.20**
PCPNet-H	0.763	0.293	6.02
PCPNet	**0.742**	**0.288**	8.72

E. Ablation Study

In the Transformer block of PCPNet, the input features are compressed along the height and width dimensions respectively, and then enhanced by the self-attention mechanism, which is introduced in Sec. III-B. To further validate the effectiveness of the proposed Transformer block, we conduct the ablation study on the KITTI validation set with two baselines, PCPNet-W and PCPNet-H. PCPNet-W only maintains Transformer_W and discards Transformer_H, while PCPNet-H only uses Transformer_H rather than Transformer_W. Here we use full-size point clouds as inputs and use $\mathcal{L}_R$ and $\mathcal{L}_M$ as the evaluation metrics. As shown in Tab. III, PCPNet outperforms PCPNet-W and PCPNet-H on both losses, which means that the two types of Transformer enhance the performance of point cloud prediction together. Besides, PCPNet-H performs better than PCPNet-W indicating that Transformer can capture more distinct spatio-temporal information along the height dimension than the width dimension. We also show the average runtime of one prediction in Tab. III. As can be seen, PCPNet costs larger inference time due to its more complex architecture than the baselines with only one Transformer block.

F. Study on Semantic Auxiliary Training

In this experiment, we compared TCNet, PCPNet, and PCPNet-Semantic on the KITTI validation set to verify the enhancement from the proposed semantic auxiliary training. Here we propose a novel metric named as *semantic similarity* to measure the difference between the semantic map $\hat{y}_{c,i,j} \in \mathbb{R}^{C_s \times H_s \times W_s}$ output by the semantic segmentation network

TABLE IV: Comparison of Semantic Similarity on the KITTI Validation Set

Approach	Semantic Similarity
TCNet	2.789
PCPNet (Ours)	2.876
PCPNet-Semantic (Ours)	**2.913**
Ground Truth (Upper Bound)	3.877

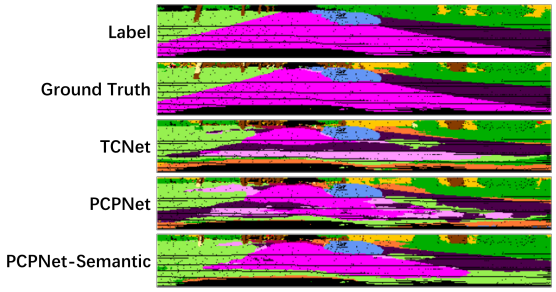

Fig. 6: Visualization of the outputs of semantic segmentation network. *Label* refers to the manual labels from SemanticKITTI, and *Ground Truth* refers to the semantic map obtained from the ground truth point clouds.

and the ground truth semantic label $y_{c,i,j} \in \mathbb{R}^{C_s \times H_s \times W_s}$, which can be formulated as

$$\textit{Semantic Similarity} = \frac{C_s \times H_s \times W_s}{\sum_{c,i,j} -y_{c,i,j} \log \hat{y}_{c,i,j}}. \tag{8}$$

According to Eq. (8), the greater the semantic similarity, the closer $\hat{y}_{c,i,j}$ is to $y_{c,i,j}$. Note that our method is completely self-supervised and we can only use the semantic map of the ground truth point clouds to train the network, but here we evaluate the effectiveness of the semantic auxiliary training using annotated semantic labels from SemanticKITTI dataset. The comparison of the semantic similarity on the KITTI validation set is shown in Tab. IV, where Ground Truth represents the semantic similarity between the semantic map from the ground truth range image and the ground truth semantic label, and is regarded as the upper bound performance. As can be seen, our methods outperform TCNet in the semantic level. Besides, the semantic similarity of PCPNet-Semantic is further improved by applying semantic auxiliary training.

The quantitative experimental results in Tab. V show that all losses of PCPNet-Semantic decrease, especially the average semantic loss $\mathcal{L}_S$ which is 23.4% lower than PCPNet. In addition, our approach still performs better than TCNet even without the enhancement from semantic auxiliary training.

The visualization of the comparison results is shown in Fig. 6. Both TCNet and PCPNet lose some semantic information, such as roads and grasses. In contrast, this situation is alleviated by PCPNet-Semantic and more laser points are correctly classified.

G. Complexity Analysis

In this experiment, we first compare the runtime of PCPNet with other baseline methods. The performance on chamfer distance v.s. runtime of networks on the KITTI test set is illustrated in Fig. 7. As can be seen, our proposed PCPNet performs best on chamfer distance while maintaining good real-time performance (8.72 ms to predict 5 future point clouds). Moreover, the complexity analysis of all the PCP methods is shown in Tab. VI. Compared with the point-based methods, the range-image-based methods have relatively lower time complexity. For example, the FLOPs of the MoNet-GRU is 239.06 billion for predicting 65536 points, while the FLOPs of our proposed PCPNet is only 54.34 billion with points predicted twice as MoNet-GRU. Although the time and space complexities of PCPNet are slightly greater than the other range-image-based method TCNet, PCPNet has better performance in predicting future point clouds due to the sophisticated network architecture.

TABLE V: Comparison of Losses on the KITTI Validation Set

Approach	$\mathcal{L}_R$	$\mathcal{L}_M$	$\mathcal{L}_S$	$\mathcal{L}_C$
TCNet	0.8143	0.2990	0.0445	0.4796
PCPNet (Ours)	0.7865	0.2932	0.0423	0.4541
PCPNet-Semantic (Ours)	**0.7694**	**0.2914**	**0.0324**	**0.4510**

TABLE VI: Complexity Analysis on the KITTI Test Set

Approach		FLOPs (billion)	Params (million)
PointLSTM	16384 pts	50.01	1.23
	65536 pts	200.04	1.23
MoNet-LSTM	16384 pts	92.76	4.00
	65536 pts	371.03	4.00
MoNet-GRU	16384 pts	59.76	3.31
	65536 pts	239.06	3.31
TCNet		30.26	17.01
PCPNet (Ours)		54.34	22.60

V. Conclusion

In this paper, we propose a self-supervised method to predict future point cloud sequences based on the given past point clouds. Benefiting from the self-attention mechanism of Transformer, our proposed network can aggregate spatio-temporal information along multiple dimensions. We also propose a semantic auxiliary training strategy to enhance the performance of forecasting more realistic point clouds for real-vehicle applications. The proposed network is evaluated on publicly available datasets using multiple metrics, and the experimental results support that our method outperforms the other state-of-the-art methods in point cloud prediction while maintaining a very fast running speed.

In the future, more types of semantic loss functions except for $L1$ norm can be adopted for an ablation study in different driving scenes. Furthermore, it may be possible to discuss whether our proposed semantic auxiliary training strategies can improve the performance of other point cloud prediction models in the future.

References

[1] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018.

[2] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5135–5142. IEEE, 2020.

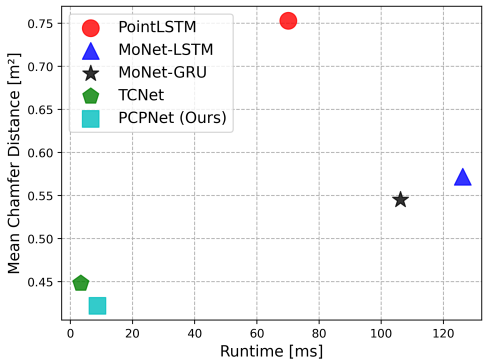

Fig. 7: The performance on chamfer distance v.s. runtime on the KITTI test set. PointLSTM, MoNet LSTM, and MoNet GRU use 16384 downsampled points as inputs to improve inference speed.

[3] Junyi Ma, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. Overlaptransformer: An efficient and yaw-angle-invariant transformer network for lidar-based place recognition. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):6958–6965, 2022.

[4] Junyi Ma, Xieyuanli Chen, Jingyi Xu, and Guangming Xiong. Seqot: A spatial-temporal transformer network for place recognition using sequential lidar data. *IEEE Transactions on Industrial Electronics*, 2022.

[5] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[6] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. *arXiv preprint arXiv:2205.05979*, 2022.

[7] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4213–4220. IEEE, 2019.

[8] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. Meteornet: Deep learning on dynamic 3d point cloud sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9246–9255, 2019.

[9] Hehe Fan and Yi Yang. Pointrnn: Point recurrent neural network for moving point cloud processing. *arXiv preprint arXiv:1910.08287*, 2019.

[10] Chaoyun Zhang, Marco Fiore, Iain Murray, and Paul Patras. Cloudlstm: A recurrent neural model for spatiotemporal point-cloud stream forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10851–10858, 2021.

[11] David Deng and Avideh Zakhor. Temporal lidar frame prediction for autonomous driving. In *2020 International Conference on 3D Vision (3DV)*, pages 829–837. IEEE, 2020.

[12] Fan Lu, Guang Chen, Zhijun Li, Lijun Zhang, Yinlong Liu, Sanqing Qu, and Alois Knoll. Monet: Motion-based point cloud prediction network. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[13] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nicholas Rhinehart. Inverting the pose forecasting pipeline with spf2: Sequential pointcloud forecasting for sequential pose forecasting. In *Conference on robot learning*, pages 11–20. PMLR, 2021.

[14] Xinshuo Weng, Junyu Nan, Kuan-Hui Lee, Rowan McAllister, Adrien Gaidon, Nicholas Rhinehart, and Kris M Kitani. S2net: Stochastic sequential pointcloud forecasting. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 549–564. Springer, 2022.

[15] Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In *Conference on Robot Learning*, pages 1444–1454. PMLR, 2022.

[16] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019.

[17] Beibei Jin, Yu Hu, Qiankun Tang, Jingyu Niu, Zhiping Shi, Yinhe Han, and Xiaowei Li. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4554–4563, 2020.

[18] Haixu Wu, Zhiyu Yao, Jianmin Wang, and Mingsheng Long. Motionrnn: A flexible model for video prediction with spacetime-varying motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15435–15444, 2021.

[19] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021.

[20] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14204–14213, 2021.

[21] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[22] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6272–6281, 2019.

[23] Xieyuanli Chen, Shijie Li, Benedikt Mersch, Louis Wiesmann, Jürgen Gall, Jens Behley, and Cyrill Stachniss. Moving object segmentation in 3d lidar data: A learning-based approach exploiting sequential data. *IEEE Robotics and Automation Letters*, 6(4):6529–6536, 2021.

[24] Jiadai Sun, Yuchao Dai, Xianjing Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. Efficient spatial-temporal information fusion for lidar-based 3d moving object segmentation. *arXiv preprint arXiv:2207.02201*, 2022.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[26] Q. Wang, B. Wu, P. Zhu, P. Li, and Q. Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[27] Lucas Nunes, Rodrigo Marcuzzi, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Segcontrast: 3d point cloud feature representation learning through self-supervised segment discrimination. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):2116–2123, 2022.

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[29] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[30] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[32] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019.

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