CF-NET: COMPLEMENTARY FUSION NETWORK FOR ROTATION INVARIANT POINT CLOUD COMPLETION

Bo-Fan Chen, Yang-Ming Yeh, Yi-Chang Lu

Graduate Institute of Electronics Engineering
National Taiwan University
{r08943131, d05943006, yiclu}@ntu.edu.tw

ABSTRACT

Real-world point clouds usually have inconsistent orientations and often suffer from data missing issues. To solve this problem, we design a neural network, CF-Net, to address challenges in rotation invariant completion. In our network, we modify and integrate complementary operators to extract features that are robust against rotation and incompleteness. Our CF-Net can achieve competitive results both geometrically and semantically as demonstrated in this paper.

Index Terms— Deep learning, Point cloud completion, Rotation invariant

1. INTRODUCTION

Due to limited viewpoints and resolutions, point clouds obtained from LiDARs or depth cameras are usually occluded and thus incomplete. This urges for solutions that can recover complete point clouds from fragmentary ones, and the task is often referred as completion. Point cloud completion benefits different applications, including 3D reconstruction, robotic vision, and autonomous driving. Most completion methods adopted an encoder-decoder architecture. PCN [1] included a PointNet-like [2] encoder and a coarse-to-fine decoder. TopNet [3], using the same encoder as PCN, proposed a tree structure decoder that can hierarchically generate output points. PFNet [4] used a hierarchical PointNet-like encoder and decoder for the completion task. Although PointNet-like architectures are robust to local defects, they fail to capture local details. To preserve fine details, the authors of [5] adopted a PointNet++ [6] based encoder to better capture local information, and they proposed to utilize point features to avoid information loss after max-pooling. Recent studies also suggested that we can concatenate the calculated features or the output point clouds with original inputs to preserve more details [7, 8, 9].

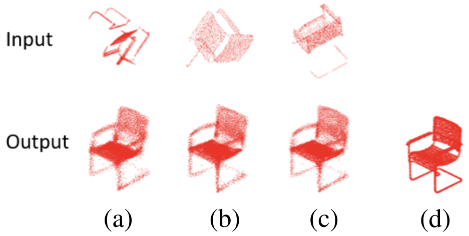


Fig. 1: Given partial point clouds with different occlusions and orientations, the completion results are consistent. (a) - (c) show the given incomplete input point clouds and the corresponding completion results by the proposed CF-Net. (d) is the ground truth.

Previous completion methods mainly focused on consistently oriented datasets, where the partial inputs and ground truth have the same orientation. In the real world, however, point clouds retrieved from different viewpoints can be rotated. Inconsistently oriented point clouds, even if completed, can degrade the performance of downstream modules. To facilitate the follow-up applications, the target orientation of the completion should be consistent.

Recently, several operators have been proposed to achieve rotation invariance on point cloud classification and segmentation. It is achieved by projecting points to a rotation invariant feature space defined by handcrafted references. RIConv [10] chose local barycenters as the reference points. AEConv [11] utilizes the local barycenter to construct a local reference frame (LRF). LGR-Net [12] combined the features extracted from two symmetric encoder branches. The local branch used local barycenters and the given surface normal as references. The global branch projected the input point cloud to the reference axes computed from SVD. These rotation invariant methods were shown robust. However, they were designed only for classification and segmentation instead of the completion tasks.

Our CF-Net combines operators with complementary advantages to achieve robustness against different perturbations. The input of our model is an arbitrarily rotated partial point cloud. The missing part is regional introduced by occlusion.

This work is partially supported by National Taiwan University under Grant number 110L890605, and the Ministry of Science and Technology, Taiwan, under Grant numbers 107-2221-E-002-123-MY2 and 110-2221-E-002-097. The authors would also like to thank National Center for High-performance Computing (NCHC) for providing computational and storage resources.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

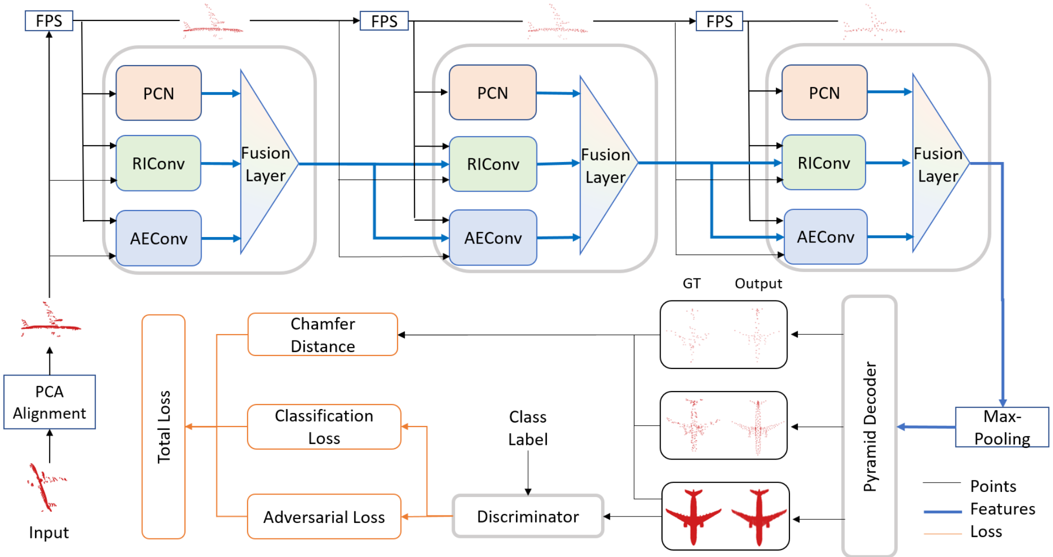


Fig. 2: Overall architecture of CF-Net. The upper half is the encoder. The decoder generates completion results based on the features learned. We also include a discriminator in our design.

The target is the completed point cloud with a unified direction. We illustrate this task in Figure 1. The outputs of our model are complete point clouds with consistent orientation, and thus can be directly fed to follow-up application modules.

2. METHODOLOGY

Our design follows an encoder-decoder structure, as illustrated in Figure 2. The proposed encoder consists of three encoder layers. The input is subsampled in each layer to extract features of different scopes. We apply max-pooling after the last encoder layer to obtain global features, and adopt the pyramid decoder proposed in [4] to transform the global features into the output point cloud. Following the decoder, we use a discriminator to improve the authenticity of completion results. Implementation details are available on our GitHub page (https://github.com/brandon9838/CFNet).

2.1. Encoder

Our encoder includes three operators, each with its strengths and weaknesses. We integrate them into our design to complement their drawbacks. We also modify the operators to make them more suitable for our task. Similar to LGR-Net [12], we use a fusion layer to integrate the point features from different sources.

PCN. The encoder proposed in PCN [1] consists of two shared MLP (multilayer perceptron) separated by a max-pooling layer. We include PCN as one of our operators. As mentioned in PCN, its architecture is robust against local defects often observed in incomplete inputs. However, PCN is not rotation invariant. To alleviate the influence of rotation, we align the partial input to its principal components at the beginning. Note that the principal components do not have a unified direction, and are sensitive to occlusions and minor structural differences. Therefore, the alignment process does not guarantee rotation invariance.

RIConv. To achieve rotation invariance, RIConv [10] projects the input points to a handcrafted rotation invariant feature space. The projection is defined on center points c_i obtained from farthest point sampling (FPS) and their kNN, $p_{(i,k)}$. The barycenter of $p_{(i,k)}$, m_i, is used as the reference point. We add $\overline{Om_i}$ in the feature space of RIConv, where O is the origin of input. This extra feature indicates the relative position of m_i. RIConv also includes a binning process, where input point features are sorted and divided into equal bins along $\overrightarrow{m_{global}m_i}$, where m_{global} is the barycenter of the input. For the partial inputs in ShapeNet, m_{global} varies according to occlusion and is not as reliable. Therefore, we replace m_{global} with O, which equals the bounding box center of ground truths in ShapeNet. It is invariant to occlusion and thus a more robust reference point.

We include RIConv as our second operator. Although RIConv is rotation invariant, it depends on local information to construct the rotation invariant feature space. Therefore, it is sensitive to local defects. To encourage our model to focus on the global feature, which is more robust to noises, we concatenate the rear part of PCN after RIConv where the global feature is concatenated with the point features.

AEConv. AEConv [11] projects the points to a rotation invariant feature space. The projection is defined on a local reference frame (LRF) using c_i, $p_{(i,k)}$, m_i, and O as reference points. We also add $\overline{Om_i}$ to the rotation invariant features.

AEConv proposed a method to align the features extracted from different LRFs before combining them. The rotational and translational differences of LRFs (R and T) are concatenated to the point features, and the following MLP performs the alignment process. We adopt AEConv as our third operator. However, AEConv also relies on local information and therefore is not robust to local defects. To improve the design, similar to our modification to RIConv, we also attach the latter part of PCN to build our modified AEConv. Note that RIConv extracts local features between neighbors in the spatial domain, while AEConv considers information from both spatial and feature domains. Their receptive fields are therefore different.

Fusion layer. Similar to LGR-Net We adopt an attention-based fusion layer to combine the features from different operators.

2.2. Decoder and discriminator

We adopt the Pyramid Decoder proposed in PF-Net [4], which hierarchically generates outputs of different resolutions. We take Chamfer Distance (CD) proposed in [13] as the completion loss. We also include the conditional discriminator proposed in ACGAN [14] to improve the output authenticity. While the model could generate an object from a random category with high authenticity, we utilize the classification loss in ACGAN to suppress this situation and further improve output quality.

3. EXPERIMENTS

3.1. Experimental setup

We choose the ShapeNet dataset provided by PCN as our testbench [1]. Each pair of the data in ShapeNet contains a complete ground truth and its partial input retrieved from a random viewpoint. Both the inputs and ground truths are diagonally normalized according to the bounding box of ground truths. The ground truths contain 16,384 points, and we resample the input to 2,048 points following the steps in [5]. We use the same train/validation/test split as PCN. Since ShapeNet is a consistently oriented dataset, for rotation invariant completion experiments, we apply SO3 augmentation to the partial inputs in ShapeNet, where the point clouds are rotated by a random angle along each of the three axes in the Cartesian coordinate system. We followed PCN [1] and trained the models for 300,000 training steps with batch size set to 32. We perform validation every 5,000 steps, and select the model with the lowest Chamfer Distance on the validation set for testing. For each experiment, we perform testing three times and report the average score.

Table 1: Completion results on the rotated (SO3) ShapeNet dataset.

Methods	CD	F-score	Acc. (%) PointNet	Acc. (%) DGCNN
PCN [1]	18.48	0.4175	67.78	68.72
TopNet [3]	30.55	0.2261	29.53	33.00
PFNet [4]	30.42	0.1840	20.11	21.92
RFA [5]	21.20	0.3754	61.17	63.28
GLFA [5]	22.78	0.3415	57.11	57.83
GRNet [16]	29.63	0.2810	50.25	52.17
Ours	**16.30**	**0.5190**	**90.47**	**88.72**
Ground Truth	-	-	95.83	96.92

3.2. Evaluation metrics

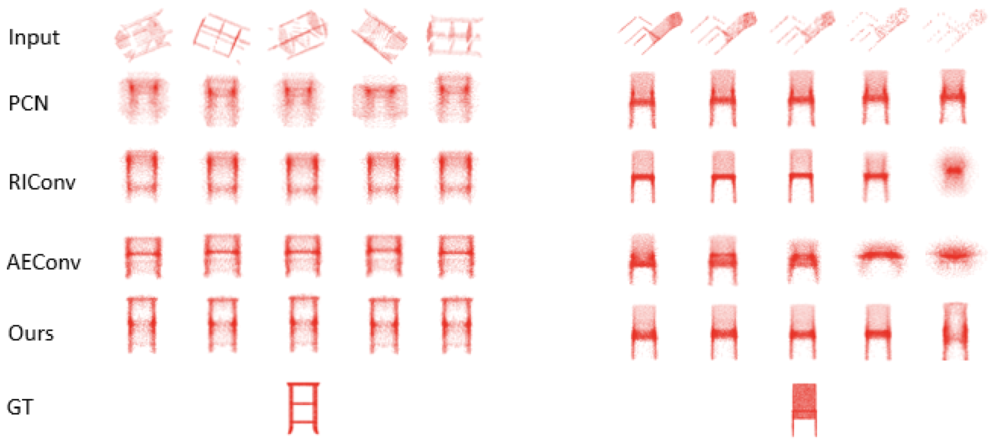
Similar to other completion researches, we use Chamfer Distance (CD) [1, 3] and F-score [8] to evaluate the completion results. The reported Chamfer Distance is multiplied by 10^3, and we set the threshold of the F-score to 0.01. These metrics measure the geometric similarities between outputs and ground truths but do not reflect the semantic differences.

To see the quality of our completion results, we use two different classification networks to semantically evaluate the model performance. One adopts a PointNet-like [2] architecture. The other one is based on DGCNN [15], which considers the local information in the feature space. These classification network are first trained on the ground truths of ShapeNet. We then feed our completion results to this pretrained classification network. We adopt classification accuracy as the evaluation metric. Higher classification accuracy indicates that the completion result is perceived as the correct object, preserving semantic meanings. This metric also demonstrates that our completion method benefits downstream modules, in this case, the classification network. We also report the classification accuracy on the ground truths as the upper bound of the metric.

3.3. Result comparison

We compare our design with other completion methods. These methods are developed initially for completion on consistently oriented datasets. Therefore, we retrained their models following our experimental settings. Table 1 shows the quantitative results. These completion methods are not rotation invariant, and they do not perform well under this setting. The proposed CF-Net achieves significant improvements in all of the metrics.

In ShapeNet, O equals the bounding box center of the ground truths. This characteristic may not hold in real-world data. To evaluate model performance under such cases, we conduct an additional experiment where O is shifted to a reference point obtainable from the input. In our experiments,



(a) Influence of rotation.

(b) Influence of defects.

Fig. 3: Operator analyses.

Table 2: Completion results on the rotated (SO3) and shifted ShapeNet dataset.

Methods	CD	F-score	Acc. (%) PointNet	Acc. (%) DGCNN
PCN [1]	20.03	0.3808	62.89	64.59
TopNet [3]	33.85	0.2209	37.39	44.61
PFNet [4]	34.95	0.2480	14.58	12.81
RFA [5]	20.84	0.4019	69.92	70.83
GLFA [5]	23.42	0.3442	58.11	60.92
GRNet [16]	31.63	0.2469	37.78	46.05
Ours	**17.60**	**0.4683**	**87.33**	**81.81**
Ground Truth	-	-	95.83	96.92

we use m_{global} as the reference. Results are shown in Table 2. For most of the methods, including ours, the performance significantly drops compared to Table 1. The performance of RFA and GLFA, proposed in [5], is consistent. We believe the T-Net architecture included in their design alleviates the influence of translation. However, CF-Net still achieves the best results under this setting. Notice that in the following sections, we still adopt our original experimental settings.

3.4. Operator analyses

We also analyze how different operators affect the results of rotation invariant completion. In this experiment, we compare our design with the original PCN, RIConv and AEConv encoders. All the four encoders are cascaded with a fully connected decoder to produce the completion results. Quantitative results are shown in Table 3. The results show that our design indeed boosts the performance.

Table 3: Completion results of different operators on the rotated (SO3) ShapeNet dataset.

Methods	CD	F-score	Acc. (%) PointNet	Acc. (%) DGCNN
PCN*	18.85	0.3440	49.70	52.61
RIConv*	19.13	0.3563	60.78	61.33
AEConv*	18.57	0.3590	61.61	63.78
Ours*	**15.81**	**0.4651**	**82.47**	**79.25**

*: with a fully connected decoder

To show the advantages of combining three different operators in our design, we illustrate the characteristics of these operators in Figure 3. In Figure 3(a), we randomly rotate the same input and observe the difference between outputs. The results show that PCN is not rotation-robust, while the other three methods can produce consistent results. In Figure 3(b), we randomly subsample the same point cloud and produce a series of inputs with increasing defects. The results show that AEConv and RIConv are sensitive to defects, while PCN produces consistent results. Our encoder provides sufficient robustness against local defects.

4. CONCLUSION

For real-world applications, a good completion method should be able to handle input data taken from different viewpoints, and generate complete point clouds of a unified orientation. In the paper, we propose a neural network, CF-Net, for rotation invariant point cloud completion. Our CF-Net is a design of the encoder-decoder structure. By combining three operators, including PCN, modified RIConv and modified AEConv, our design can generate quality results semantically and geometrically.

5. REFERENCES

[1] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert, "Pcn: Point completion network," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 728–737.

[2] 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*, 2017, pp. 652–660.

[3] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese, "Topnet: Structural point cloud decoder," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 383–392.

[4] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le, "Pf-net: Point fractal network for 3d point cloud completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7662–7670.

[5] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao, "Detail preserved point cloud completion via separated feature aggregation," *arXiv preprint arXiv:2007.02374*, 2020.

[6] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[7] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu, "Morphing and sampling network for dense point cloud completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 11596–11603.

[8] Hyeontae Son and Young Min Kim, "Saum: Symmetry-aware upsampling module for consistent point cloud completion," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[9] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee, "Cascaded refinement network for point cloud completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 790–799.

[10] Zhiyuan Zhang, Binh-Son Hua, David W Rosen, and Sai-Kit Yeung, "Rotation invariant convolutions for 3d point clouds deep learning," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 204–213.

[11] Junming Zhang, Ming-Yuan Yu, Ram Vasudevan, and Matthew Johnson-Roberson, "Learning rotation-invariant representations of point clouds using aligned edge convolutional neural networks," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 200–209.

[12] Chen Zhao, Jiaqi Yang, Xin Xiong, Angfan Zhu, Zhiguo Cao, and Xin Li, "Rotation invariant point cloud classification: Where local geometry meets global topology," *arXiv preprint arXiv:1911.00195*, 2019.

[13] 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*, 2017, pp. 605–613.

[14] Augustus Odena, Christopher Olah, and Jonathon Shlens, "Conditional image synthesis with auxiliary classifier gans," in *International conference on machine learning*. PMLR, 2017, pp. 2642–2651.

[15] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–12, 2019.

[16] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun, "Grnet: gridding residual network for dense point cloud completion," in *European Conference on Computer Vision*. Springer, 2020, pp. 365–381.