BILATERAL COARSE-TO-FINE NETWORK FOR POINT CLOUD COMPLETION

Tran Thanh Phong Nguyen, Son Lam Phung, Vinod Gopaldasani, Jane Whitelaw

University of Wollongong, NSW, Australia

ABSTRACT

Point cloud completion aims to accurately estimate complete point clouds from partial observations. Existing methods often directly infer the missing points from the partial shape, but they suffer from limited structural information. To address this, we propose the Bilateral Coarse-to-Fine Network (BCF-Net), which leverages 2D images as guidance to compensate for structural information loss. Our method introduces a multi-level codeword skip-connection to estimate structural details. Experimental results show that BCF-Net outperforms state-of-the-art point cloud completion networks on synthetic and real-world datasets.

***Index Terms*—** Point cloud completion, 3D object modeling, bilateral filtering, image guidance.

1. INTRODUCTION

Point cloud completion is a vital technique for improving the quality of 3D data, particularly when capturing occluded or constrained views. As 3D applications become more widespread, the need for accurate and complete 3D data is crucial. This paper focuses on point cloud completion, which predicts the complete 3D shape of an object from partial observations. Despite its significance, point cloud completion still has many challenges in improving the accuracy and efficiency, and coping with complex and diverse environments.

There are still two critical research gaps that need to be addressed. First, current approaches [1–9] predict the complete point cloud using only a partial point cloud as input. This leads to the loss of crucial information from the missing parts, making the prediction of missing points uncertain. Most methods [1–5] leverage an encoder-decoder architecture to address this issue, where an encoder maps the input point clouds into a codeword, and a decoder reconstructs a complete point cloud by decoding the codeword back to Euclidean space. However, the lack of information about the missing points in the partial point cloud remains a significant challenge. Second, the convolution operation cannot be directly applied to point clouds due to their irregularity and unorderedness, leading some methods [10–12] to convert the point cloud into voxels for 3D convolutional neural networks. However, this voxelization operation has two issues. First, it results in an irreversible loss of geometric information, and second, it can be computationally expensive.

This paper proposes a novel approach to address the gaps in existing point cloud completion methods. To compensate for the information loss, we combine *2D and 3D modules* in a non-trivial design. Specifically, we leverage guided features extracted from 2D images to complete the point cloud in a coarse-to-fine manner, thereby providing a more accurate prediction of missing points. To address the structural information problem, we propose the concatenation of multi-scale codewords/latent spaces, which performs as a skip-connection operation. This concatenation brings information from multiple levels to complete the point cloud. To enhance the 2D codeword, we leverage a variational auto-encoder to regularize the latent space distribution, improving the overall performance of the method. Additionally, our approach avoids the computational expense of voxelization by proposing a lightweight yet practical network consisting of both 2D and 3D modules. This network can handle irregular and unordered point clouds, providing more flexibility and efficiency in point cloud completion tasks.

The main contributions of this paper are threefold:

- We propose a novel auto-encoder architecture that combines 2D and 3D modules to address the structural loss of incomplete point clouds. Code is available at *https://github.com/phongnguyenai/BCF-Net*.
- We introduce a multi-level codeword combination that functions as a multi-scale skip-connection operation to predict and maintain structural details.
- We present experimental results that demonstrate improved completion outcomes compared to existing approaches on both synthetic and real-world data.

2. RELATED WORK

This section presents a brief review of algorithms for point cloud completion and view-point guidance on point clouds.

This research is supported by a joint scholarship from Safety Equipment Australia Pty. Ltd. and the University of Wollongong. The first author (T. T. P. Nguyen) is also supported by the Centre for Occupational, Public and Environmental Research in Safety and Health (COPERSH) and the Centre for Signal and Information Processing (CSIP) at the University of Wollongong.

2.1. Point cloud completion

In recent years, several methods have been proposed to address the problem of point cloud shape completion, which involves predicting complete shapes from partial observations. FoldingNet [2] and PCN [1] use folding-based decoders to create a universal 2D-to-3D mapping and generate complete point clouds in a coarse-to-fine manner. TopNet [3] utilizes a free-structured decoder to improve structure-aware point cloud generation, while GR-Net [4] uses 3D grids and skip-connections to learn context-aware and spatially-aware features. More recently, PoinTr [5] has utilized Transformers to learn structural information and global correlations for point cloud completion.

2.2. View-point guidance on point clouds

Bilateral filtering [13] uses the image as guidance for target reconstruction. Su et al. [14] apply this idea to solve the segmentation task. They use sparse bilateral convolutional layers [15] to join the 2D-3D information and enable hierarchical and spatially-aware feature learning. Our method also incorporates point-based and image-based representations to solve the point cloud completion task.

3. PROPOSED BILATERAL COARSE-TO-FINE NETWORK

This section presents the network architecture (Section 3.1), *2D modules* (Section 3.2), *3D modules* (Section 3.3), and network optimization (Section 3.4).

3.1. Network architecture

We propose the BCF-Net for point cloud completion, as shown in Fig. 1. Our network has two inputs: the input 2D image $\mathbf{I}$ and the partial point cloud $\mathbf{S}_p$. The input 2D image $\mathbf{I}$ is accepted by the *2D modules* to generate the reconstructed 2D image $\hat{\mathbf{I}}$ and the 2D-to-3D shape $\hat{\mathbf{S}}'$. The partial point cloud $\mathbf{S}_p$ is accepted by the *3D modules* to generate the reconstructed point cloud $\hat{\mathbf{S}}$.

3.2. 2D modules

Encoder2D and 2D-to-2D Decoder: The *Encoder2D* and the *2D-to-2D Decoder* are designed to form a variational auto-encoder. The *Encoder2D* is a concatenation of convolutional layers and MLP layers. The input 2D image $\mathbf{I}$, which is a $w \times h$ matrix, is fed into convolutional layers and MLP layers to generate a 2D codeword $\mathbf{z}$ with size 1-by-N. The 2D codeword $\mathbf{z}$ is passed to two modules simultaneously: *2D-to-2D Decoder* and *2D-to-3D Decoder*. *2D-to-2D Decoder* is a concatenation of MLP layers and de-convolutional layers, which is a reverse operation of the *Encoder2D*. The output of *2D-to-2D Decoder* is a reconstructed 2D image $\hat{\mathbf{I}}$, which is utilized to compute the 2D reconstruction loss.

2D-to-3D Decoder: *2D-to-3D Decoder* is the concatenation of MLP layers, de-convoluational layers, and convolutional layers. The input of this module is the 2D codeword $\mathbf{z}$, which is generated by the *Encoder2D* module. The output of this module is a 2D-to-3D shape $\hat{\mathbf{S}}'$ with size $p \times 3$, each row of which represents the Cartesian coordinates of a point.

3.3. 3D modules

Coarse shape concatenation: Suppose the 2D-to-3D shape is $\hat{\mathbf{S}}'$ with size $p \times 3$. The partial point cloud is $\mathbf{S}_p$ with size $n \times 3$. First, we concatenate the $\hat{\mathbf{S}}'$ and $\mathbf{S}_p$ into a matrix $\mathbf{S}_c$ with size $(n+p) \times 3$. Then, we apply the farthest point sampling (FPS) technique to sample the $\mathbf{S}_c$ to the $\mathbf{S}_f$ with size $q \times 3$. The operation can be described as

$$\mathbf{S}_f = F_{\text{FPS}}(cat(\hat{\mathbf{S}}', \mathbf{S}_p)), \tag{1}$$

where *cat* is the concatenation operation and F_{FPS} is the FPS operation.

Shared Encoder3D: The *Encoder3D* consists of PointNet++ [16] layers to map the input 3D shapes into the codewords. The *Encoder3D* is trained to handle two tasks at the same time. First, it takes the coarse shape $\mathbf{S}_f$ as the input and generates the 3D fine codeword $\mathbf{z}'$ $1 \times N$. Second, it accepts partial point cloud $\mathbf{S}_p$ as the input and produces the 3D coarse codeword $\mathbf{z}''$ as the output. Finally, we perform the concatenation operation on $\mathbf{z}$, $\mathbf{z}'$, and $\mathbf{z}''$ to generate the concatenated codeword $\mathbf{z}_c$ with size $1 \times (N+N+N)$ as follows:

$$\mathbf{z}_c = cat(cat(\mathbf{z}, \mathbf{z}'), \mathbf{z}''). \tag{2}$$

3D-to-3D Decoder: *3D-to-3D Decoder* adopts the folding-based decoder architecture from [2], which uses two consecutive 3-layer perceptrons to warp a fixed 2D grid into the shape of the input point cloud. We "fold" the codeword $\mathbf{z}_c$ twice to generate the reconstructed point cloud $\hat{\mathbf{S}}$ with size $m \times 3$.

3.4. Network optimization

Let $\mathbf{S}$ be the ground truth point cloud, we compute three reconstruction loss functions: the 2D reconstruction loss $L_{2D}(\mathbf{I}, \hat{\mathbf{I}})$, the Chamfer Distance (CD) for 2D-to-3D shape $L_{coarse}(\mathbf{S}, \hat{\mathbf{S}}')$, and the CD for fine shape $L_{fine}(\mathbf{S}, \hat{\mathbf{S}})$.

First, the 2D reconstruction loss function is computed as

$$L_{2D}(\mathbf{I}, \hat{\mathbf{I}}) = \sum(\mathbf{I} - \hat{\mathbf{I}})^2 + L_{\text{KL}}, \tag{3}$$

where L_{KL} is the Kullback–Leibler (KL) loss. Second, we compute the errors between the ground truth point cloud $\mathbf{S}$ and the 2D-to-3D shape $\hat{\mathbf{S}}'$ as

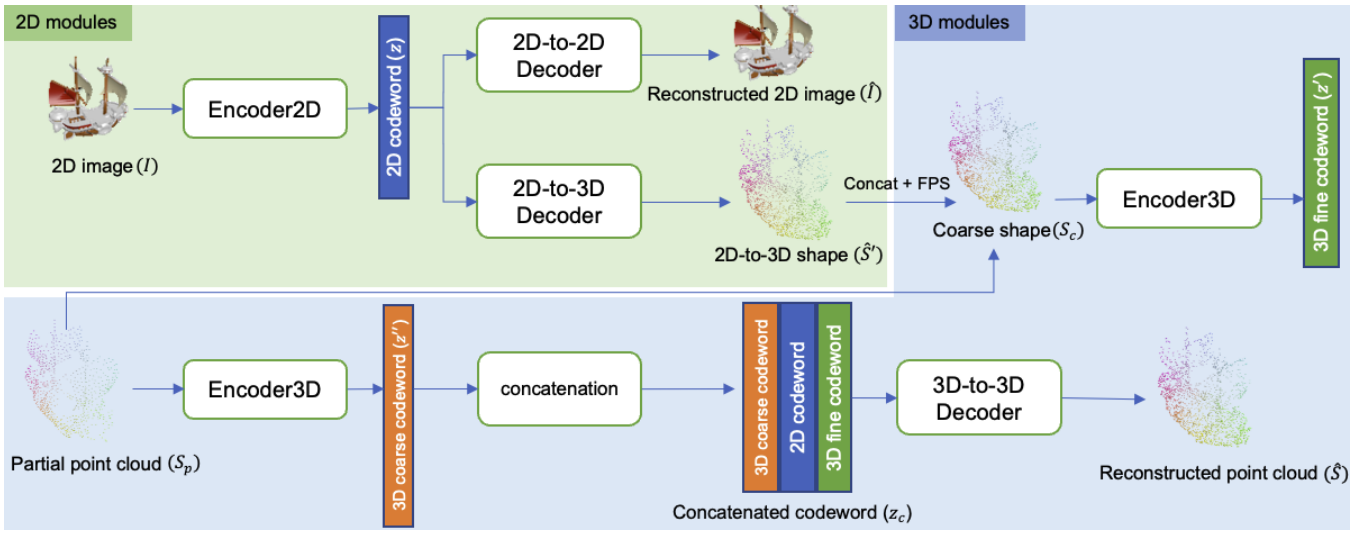

Fig. 1: The network architecture of BCF-Net for point cloud completion.

$$L_{coarse}(\mathbf{S},\hat{\mathbf{S}}') = \frac{1}{|\mathbf{S}|}\sum_{\mathbf{x}\in\mathbf{S}} min||\mathbf{x}-\hat{\mathbf{x}}||_2^2 + \frac{1}{|\hat{\mathbf{S}}'|}\sum_{\hat{\mathbf{x}}\in\hat{\mathbf{S}}'} min||\hat{\mathbf{x}}-\mathbf{x}||_2^2. \tag{4}$$

Third, we utilize the same formula as Equation (4) to compute the errors $L_{fine}(\mathbf{S},\hat{\mathbf{S}})$ between the ground truth point cloud $\mathbf{S}$ and the reconstructed point cloud $\hat{\mathbf{S}}$.

The total training loss function is computed as follows:

$$L_{training} = \alpha L_{CD}(\mathbf{S},\hat{\mathbf{S}}) + \beta L_{CD}(\mathbf{S},\hat{\mathbf{S}}') + \gamma L_{2D}(\mathbf{I},\hat{\mathbf{I}}). \tag{5}$$

In Equation (5), we set $\alpha = 1$ to show the importance of the fine shape loss L_{fine}. We set $\beta = 0.5$ to optimize the coarse shape loss L_{coarse}. We set $\gamma = 10^{-4}$ because the value scale of the L_{2D} loss is much larger than the L_{fine} and the L_{coarse}. These hyper-parameters α, β, and γ are fine-tuned carefully during the experiments. In the validation and test stages, we only utilize L_{fine}.

4. EXPERIMENTS AND ANALYSIS

This section presents the datasets and experimental methods (Section 4.1), evaluations on ShapeNet (Section 4.2), evaluations on KITTI (Section 4.3), and ablation studies (Section 4.4).

4.1. Datasets and experimental methods

For comprehensive comparisons, we conduct experiments on synthetic and real-world datasets. First, we evaluate our method on the synthetic dataset ShapeNet [17], which consists of objects from eight categories including airplane, cabinet, car, chair, lamp, sofa, table, and watercraft. For the real-world scans, we evaluate real cars extracted from the KITTI dataset [18, 19]. Our method is compared to five state-of-the-art (SOTA) point cloud completion methods: PCN [1], FoldingNet [2], TopNet [3], GRNet [4], and PoinTr [5]. We utilize their open-source code and hyper-parameters for the comparisons.

4.2. Evaluations on ShapeNet

4.2.1. *Quantitative results*

We evaluate the methods utilizing the Chamfer Distance on the 16,384 points of each shape. The results on each category and the average results are summarized in Table 1. The results show that our method outperforms other methods in most categories on the Chamfer Distance metric.

Table 1: Quantitative results on ShapeNet [17] using the Chamfer Distance metric. The best results are highlighted in bold.

	Avg	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
PCN [1]	1.112	0.496	1.278	0.662	1.172	1.513	1.923	1.063	0.788
FoldingNet [2]	1.262	0.554	1.626	0.595	1.803	1.620	1.641	1.318	0.937
TopNet [3]	2.080	0.694	2.451	1.801	2.145	2.381	3.804	1.807	1.556
GRNet [4]	1.026	0.666	1.091	0.567	**1.105**	1.649	1.345	1.124	**0.665**
PoinTr [5]	5.014	3.010	5.709	4.427	6.016	4.304	9.781	4.040	2.828
Ours	**0.913**	**0.432**	**1.002**	**0.473**	1.153	**1.385**	**1.057**	**1.055**	0.745

4.2.2. *Qualitative results*

We also visualize the results produced by the compared methods. Results on the representative examples are shown in Fig. 2. FoldingNet [2], TopNet [3], and PoinTr [5] are confused between the airplane and car. PCN [1] can predict the general shape of objects, but the structural details are not captured effectively. FoldingNet [2], TopNet [3], PoinTr [5], and

PCN [1] fail to capture the curve shape of the chair. Meanwhile, GRNet [4] exhibits clearer part structures and more neatly arranged points in most categories, e.g., the chair and car. However, the points at the bottom of the airplane are missing. Our method overcomes these problems and shows a visually better performance. Because it leverages the image guidance and the non-trivial combination between multi-scale 2D and 3D features in the coarse-to-fine design.

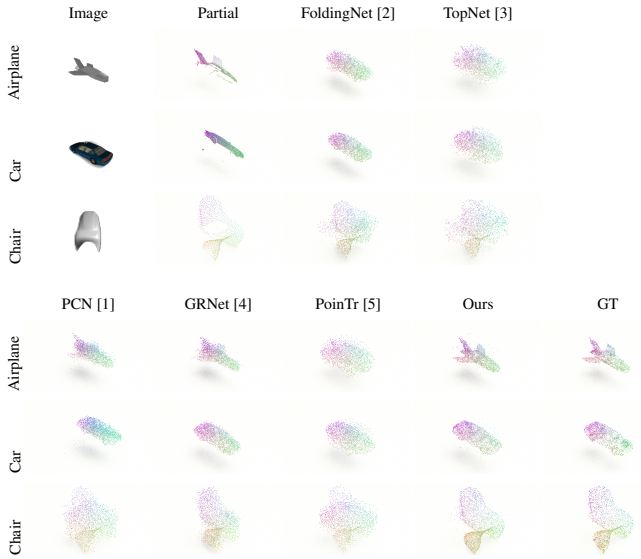

Fig. 2: Visual results of point cloud completion methods on the ShapeNet dataset [17].

4.3. Evaluations on KITTI

KITTI [18, 19] includes real-world car point clouds captured by laser scanners. Thus there is no ground truth in this dataset. We visualize the results produced by the methods in Fig. 3. It is easy to observe that FoldingNet [2] and TopNet [3] fail to predict the complete shape of the car. PCN [1], PoinTr [5], and GRNet [4] can capture the overall shape of the car. However, structural details are missing from their predictions. For example, the top of the car of GRNet [4] is incomplete, the wheels from PCN [1] are missing, and the results of PoinTr [5] are noisy and distorted. The result of our method overcomes these problems. We estimate the overall shape and structural details of the real-world car accurately.

4.4. Ablation studies

We conduct ablation experiments to study the contributions of the *2D modules* in our method. In this section, we generate a variant of our method by removing the *2D modules* from the original network. For quantitative evaluation, we compare the performance of the variant method with the original network on ShapeNet [17] on the Chamfer Distance metric. The results on each category and the average results are shown in

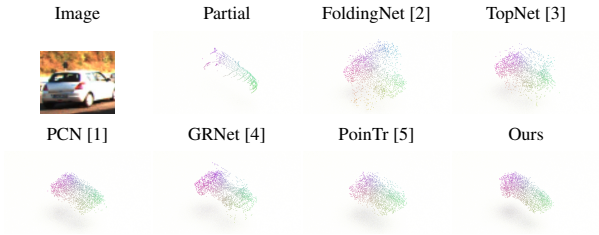

Fig. 3: Visual results of point cloud completion methods on the KITTI dataset [18, 19].

Table 2. The performances in all categories drop when we remove the *2D modules*, which shows the importance of the image guidance to our network. For qualitative evaluation, we visualize the results produced by the variant method for a more comprehensive evaluation. Results on the representative examples from the ShapeNet [17] and KITTI [18, 19] are shown in Fig. 4. For ShapeNet [17], the prediction from the variant method is noisy because it has no guidance from the image like the original network. For KITTI [18, 19], the variant method can capture the general shape. However, it fails to capture the structural details (e.g., the car wheel) because the 2D-3D multi-scale skip-connections are removed.

Table 2: Quantitative results for ablation study on ShapeNet [17] using CD metric. The best results are highlighted in bold.

	Avg	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
Variant	1.265	0.533	1.319	0.728	1.307	1.540	2.290	1.530	0.876
Ours	**0.913**	**0.432**	**1.002**	**0.473**	**1.153**	**1.385**	**1.057**	**1.055**	**0.745**

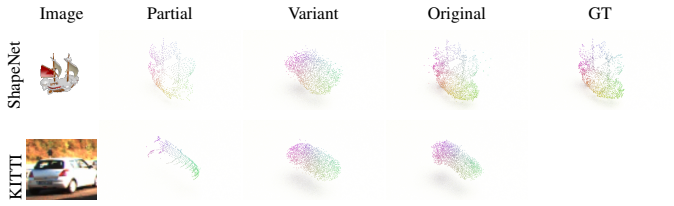

Fig. 4: Visual results of the ablation studies on ShapeNet [17] and KITTI [18, 19] datasets.

5. CONCLUSION

The proposed BCF-Net successfully addresses the structural loss problem in point cloud completion by leveraging both point-based and image-based representations. The multi-level skip-connection codeword enables the preservation of non-missing regions and the estimation of local structural details. Our BCF-Net outperforms existing SOTA methods on both synthetic and real-world datasets. A future research direction is to explore the use of multiple 2D views to further improve the performance of point cloud completion.

6. REFERENCES

[1] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *International Conference on 3D Vision*, 2018, pp. 728–737.

[2] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 206–215.

[3] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "Topnet: Structural point cloud decoder," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 383–392.

[4] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "Grnet: Gridding residual network for dense point cloud completion," in *European Conference on Computer Vision*, 2020, pp. 365–381.

[5] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12498–12507.

[6] Y. Cai, K. Y. Lin, C. Zhang, Q. Wang, X. Wang, and H. Li, "Learning a structured latent space for unsupervised point cloud completion," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5543–5553.

[7] Y. Wang, D. J. Tan, N. Navab, and F. Tombari, "Learning local displacements for point cloud completion," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1568–1577.

[8] P. Xiang, X. Wen, Y. S. Liu, Y. P. Cao, P. Wan, W. Zheng, and Z. Han, "Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5499–5509.

[9] X. Wen, Z. Han, Y. P. Cao, P. Wan, W. Zheng, and Y. S. Liu, "Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13080–13089.

[10] A. Dai, C. R. Qi, and M. Nießner, "Shape completion using 3D-encoder-predictor cnns and shape synthesis," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5868–5877.

[11] D. Stutz and A. Geiger, "Learning 3D shape completion from laser scan data with weak supervision," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1955–1964.

[12] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel CNN for efficient 3D deep learning," in *Advances in Neural Information Processing Systems*, 2019.

[13] K. He, J. Sun, and X. Tang, "Guided image filtering," in *European Conference on Computer Vision*, 2010, pp. 1–14.

[14] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M. H. Yang, and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2530–2539.

[15] V. Jampani, M. Kiefel, and P. V. Gehler, "Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4452–4461.

[16] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5105—5114.

[17] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al., "Shapenet: An information-rich 3D model repository," in *arXiv preprint*, 2015, p. arXiv:1512.03012.

[18] G. Andreas, L. Philip, and U. Raquel, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[19] G. Andreas, L. Philip, S. Christoph, and U. Raquel, "Vision meets robotics: The KITTI dataset," in *International Journal of Robotics Research*, 2013, pp. 1231–1237.