

# Supplementary Material:

## PU-EVA: An Edge-Vector based Approximation Solution for Flexible-scale Point Cloud Upsampling

Luqing Luo<sup>1</sup>, Lulu Tang<sup>1\*</sup>, Wanyi Zhou<sup>2</sup>, Shizheng Wang<sup>3</sup>, Zhi-Xin Yang<sup>1†</sup>

<sup>1</sup>State Key Laboratory of Internet of Things for Smart City, University of Macau

<sup>2</sup>South China University of Technology

<sup>3</sup>Institute of Microelectronics Chinese Academy of Sciences

{gabrielle.tse, lulutamg}@outlook.com, 202021044254@mail.scut.edu.cn,

shizheng.wang@foxmail.com, zxyang@um.edu.mo

### Overview

In this supplementary material, we provide additional content to complement the paper, which are listed below:

- In Section **A**, we present detailed configurations of PU-EVA and discuss the influence of number of  $R$  anchor points on the performance.
- In Section **B**, we show qualitative comparison of upsampling results of EVA upsampling and NodeShuffle [3].
- In Section **C**, we show a typical failure case of PU-EVA.
- In Section **D**, we visualize and discuss PU-EVA results on real-scanned LiDAR point clouds.

### A. Configuration of PU-EVA

The network configurations of our PU-EVA are listed as follows.

1. **Dense Feature Extraction:** Four dense blocks based on EdgeConv with skip-connections are employed in this module. Within each dense block, three convolutional layers with output channels 24 are skip-connected; between each dense block, the features produced by each block are fed as input to all following blocks, resulting the output channels of four dense blocks as 120, 240, 360 and 480, sequentially.
2. **Edge-Vector based Approximation (EVA) Upsampling:** The output channels of  $1 \times 1$  convolutional layer

for high dimensional features  $g$  and  $h$  are 240, and that for  $l$  are 480.  $K$  is set as 12 to obtain the local neighborhood and  $R$  is set as 6 for anchor points.  $l$  is max-pooled into dimension of  $N \times 1 \times 480$  before tiling by upsampling rate ( $R = 6$ ) to expand as  $N \times 6 \times 480$ . The output of affine combinations  $N \times 6 \times 3$  is then concatenated with the expanded  $l$  to get features with dimension  $N \times 6 \times (480 + 3)$  for coordinates regression.

3. **Coordinate Reconstruction:** Displacement error from the second-order error term of Taylor’s Expansion is estimated in this module by utilizing three fully connect layers (MLPs) with output channels 256, 128 and 64, respectively. The upsampling coordinates are reconstructed on the last coordinate regression layer with output dimension of  $RN \times 3$ .

Besides, all the convolutional layers and fully connected layers in the network are followed by the ReLU activation function, except for the last coordinate regression layer.

### B. More qualitative comparison of upsampling results

To explore the efficacy of proposed EVA upsampling unit, we visualize the upsampling results of EVA upsampling and the best upsampling unit in PU-GCN [3], named NodeShuffle. As shown in the blown-up views in Figure 1, EVA upsampling unit achieves better upsampling results in term of fine-grained details, it confirms with our intuition that our EVA upsampling retains richer information from sharp edges and tiny structures. Instead of previous methods simply duplicating the original points, NodeShuffle [3] encodes spatial information from neighboring points and learns new points from the latent space. However, there

\*Equal contributions as co-first author

†Corresponding author

is no further geometric operations on features, making the upsampling results inferior to our method.

### C. Failure cases

As shown in Figure 4 of the main paper, our method handles fine-grained details well, such as the ox’s ear and the statue’s leg. However, there are still some very challenging cases that the upsampled point clouds fail to approximate the complex geometries accurately. The blown-up views of Figure 2 visualize details of the upsampling results, where the compared state-of-the-art method PU-GAN [2] suffers from the same problem as well.

### D. Real-scanned LiDAR results

The upsampling results of PU-EVA on LiDAR-scanned street scenes from Kitti [1] is provided in Figure 3. Visualization results show that our PU-EVA recovers the sparse and non-uniform input to obtain sharper object shapes of pedestrian and cyclist, etc.

### References

- [1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 4
- [2] R. Li, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7203–7212, 2019. 2
- [3] G. Qian, A. Abualshour, G. Li, A. Thabet, and B. Ghanem. Pu-gcn: Point cloud upsampling using graph convolutional networks. *arXiv preprint arXiv:1912.03264*, 2019. 1, 3
- [4] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2018.

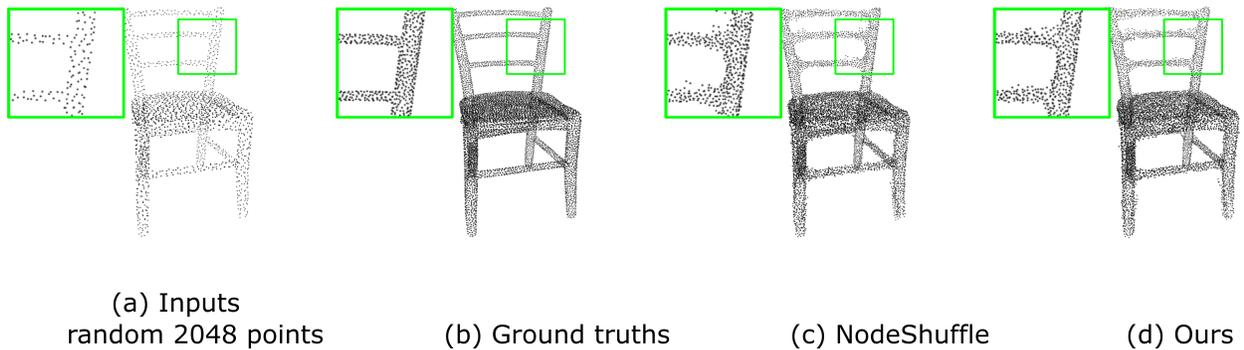


Figure 1. Qualitative comparison of upsampling unit with NodeShuffle [3]

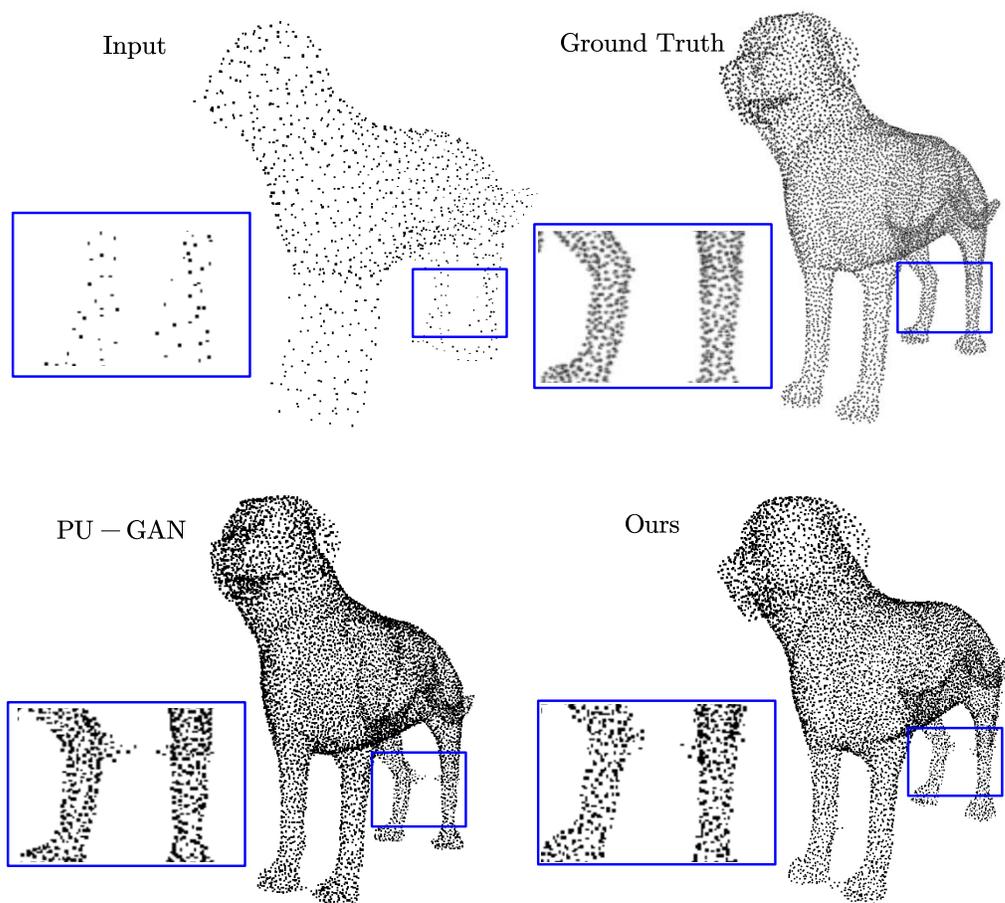


Figure 2. A typical failure case for PU-EVA.

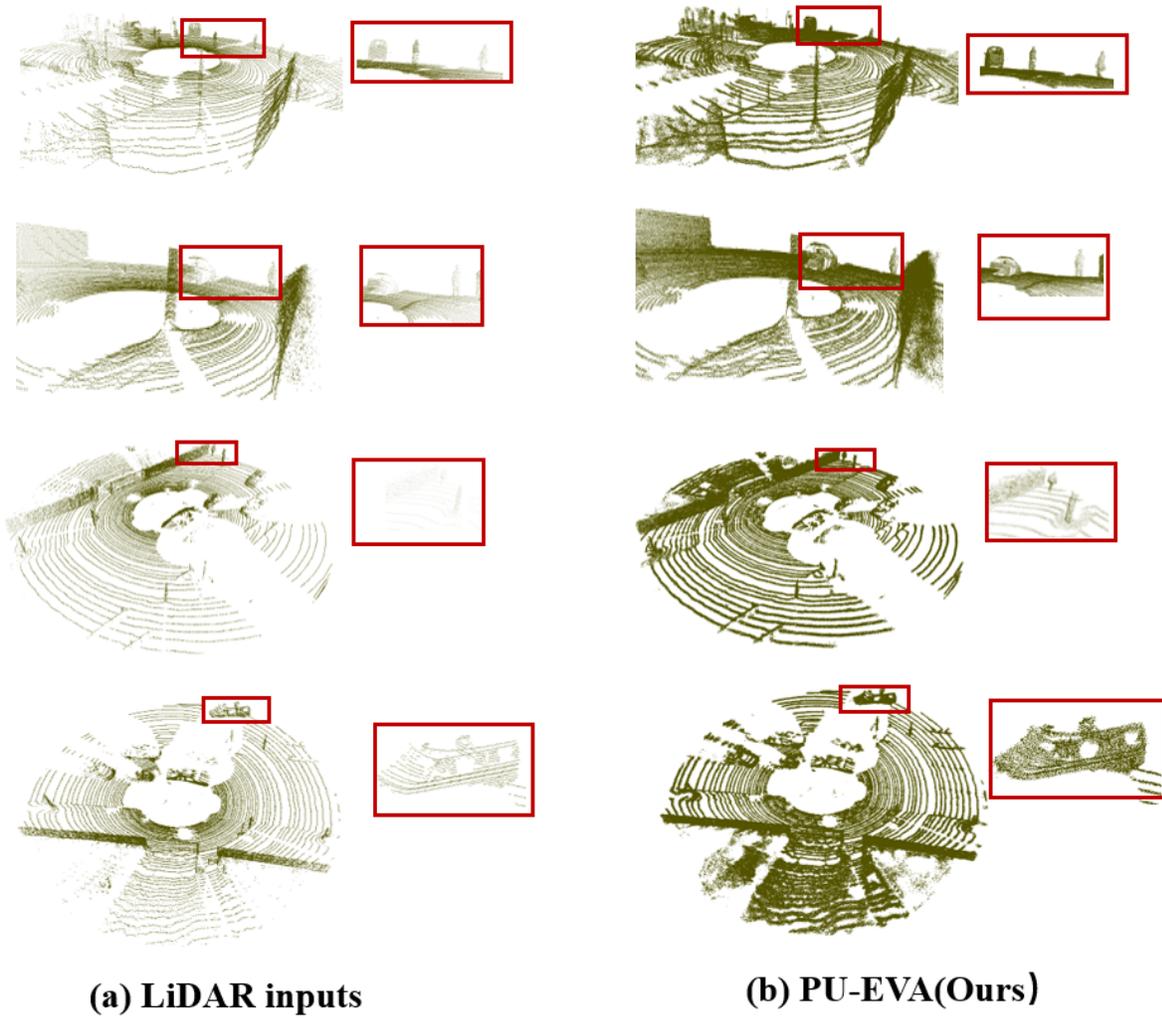


Figure 3. Real-world testing by applying PU-EVA to LiDAR-scanned street scenes from Kitti [1].