Supplementary Material
—EdgeMovingNet: Edge-preserving Point Cloud Reconstruction via Joint Geometry Features

Xinran Yang[1], Donghao Ji[1], Yuanqi Li[1*], Junyuan Xie[1], Jie Guo[1], Yanwen Guo[1,2*]
[1]Nanjing University, Nanjing, China, [2]School of Software, North University of China
{xryang,donghaoji}@smail.nju.edu.cn
{yuanqili,jyxie,guojie,ywguo}@nju.edu.cn

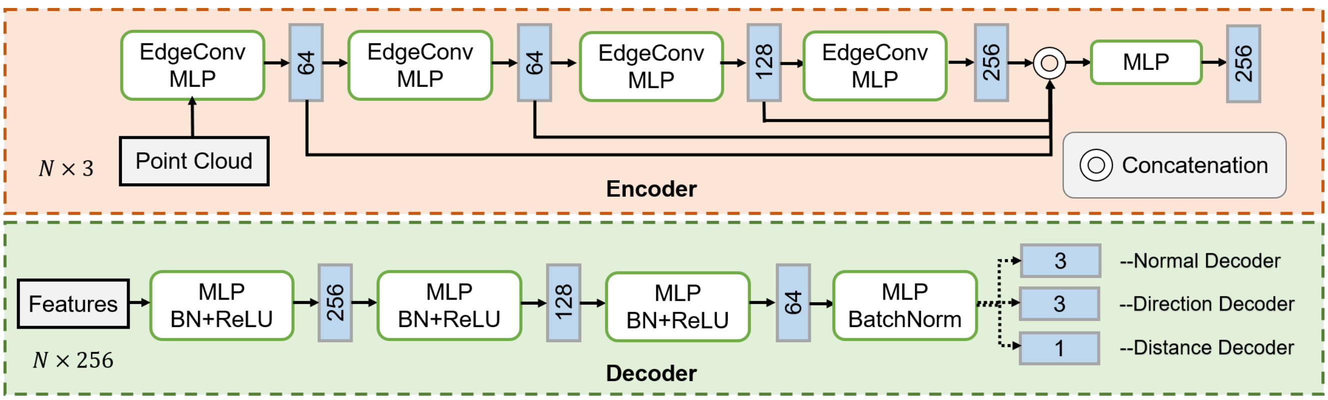

Figure 1. The architecture of encoder and decoder.

A. Network Details

EdgeMovingNet employs an encoder-decoder backbone. The architecture of encoder and decoder is shown in Fig. 1. For an input point cloud of size $N \times 3$, we stack four Edge-Conv [10] blocks as encoder to transform 3D coordinates into embeddings, which produces $N \times 256$-dimension vectors as point-wise features capturing both local and global information. Subsequently, these feature vectors are fed into three distinct decoders: normal-decoder, direction-decoder and distance-decoder. Each decoder consists of four fully connected (FC) layers. The first three layers incorporate batch normalization and ReLU activation, resulting in output dimensions of 256, 128, and 64, respectively. The final layer, equipped with batch normalization, produces outputs of different dimensions: three for normal-decoder which estimates normal n, three for direction-decoder that estimates point-to-edge direction n_e, and one for distance-decoder that estimates point-to-edge distance.

*Corresponding author.

B. Comparison Metrics

To quantitative compare our EdgeMovingNet with other methods, we introduce Chamfer distance(CD), Hausdorff distance(HD) and normal consistency(NC) as metrics for measuring the approximation error between the reconstruction results and the ground truth model. We randomly sample dense point clouds P_D and Q_D (both containing W = 100K points) from the ground truth model and the reconstructed mesh surface. The Chamfer distance D_c and Hausdorff distance D_h between P_D and Q_D are defined as follows:

$$D_c = \frac{1}{W}\sum_{i=1}^{W}\min_{\mathbf{v}\in Q_D}\|\mathbf{u}_i - \mathbf{v}\|^2 + \frac{1}{W}\sum_{j=1}^{W}\min_{\mathbf{u}\in P_D}\|\mathbf{v}_j - \mathbf{u}\|^2 \tag{1}$$

$$D_h = \max\left(\max_{\mathbf{u}\in P_D}\min_{\mathbf{v}\in Q_D}\|\mathbf{u}-\mathbf{v}\|^2, \max_{\mathbf{v}\in Q_D}\min_{\mathbf{u}\in P_D}\|\mathbf{v}-\mathbf{u}\|^2\right) \tag{2}$$

where $\mathbf{u}_i$ or $\mathbf{v}_j$ is one point in P_D or Q_D, respectively. And the normal consistency N_c is formulated as:

$$N_c = \frac{1}{2}\left(\frac{1}{W}\sum_{i=1}^{W}|n_{\mathbf{u}_i}\cdot n_{\mathbf{v}}| + \frac{1}{W}\sum_{j=1}^{W}|n_{\mathbf{v}_j}\cdot n_{\mathbf{u}}|\right) \quad (3)$$

where $n_{\mathbf{u}_i}$ and $n_{\mathbf{v}_i}$ are unit normal vectors of $\mathbf{u}_i$'s and $\mathbf{v}_i$'s sampled surface, v is the index of nearest point to $\mathbf{u}_i$ in Q_D and u is the index of nearest point to $\mathbf{v}_i$ in P_D.

C. Backbone Ablation

We also conduct ablation studies on different backbone choice of EdgeMovingNet. We test three classic point cloud feature learning backbones—PointNet++ [8], DGCNN [10] and PointTransformer [13] as our network encoder. All of them are able to predict accurate edge points and generate similar edge-preserving results based on our pipeline. The detailed evaluation on metrics is presented in Tab. 1. We selected DGCNN as the final backbone for most of our experiments because it is lightweight and powerful enough to produce good results for our reconstruction task. In the future, with the advancement of more effective point cloud feature encoders, our EdgeMovingNet can benefit more from them.

Table 1. Backbone ablation study of EdgeMovingNet. CD and HD are multiplied by 10^2

Backbone	CD↓	HD↓	NC↑	Params.	FLOPs
PointNet++[8]	0.744	1.488	0.978	1.0M	7.2G
DGCNN[10]	0.703	1.368	**0.991**	1.3M	4.8G
PointTransformer[13]	**0.702**	**1.352**	0.990	7.8M	5.6G

D. Hyperparameter Ablation

We offer the more detailed ablation on the choice of hyperparameters within our pipeline, including the distance threshold of edgemask δ, radius of edge neighbors r, the angle threshold θ and the optimization refinement parameter μ. This analysis is also performed on our dataset from ABC[6]. The evaluation results for these ablations are summarized in Tab. 2, where CD and HD values are scaled by a factor of 10^2.

Table 2. Ablation on hyperparameters.

Edgemask δ	CD ↓	HD ↓	Radius r	CD ↓	HD ↓
0.02	0.94	1.57	0.05	0.84	1.52
0.05	**0.71**	**1.37**	0.1	**0.70**	**1.38**
0.01	0.82	1.44	0.2	0.81	1.50
Angle θ	CD ↓	HD ↓	Parameter μ	CD ↓	HD ↓
5°	0.77	1.42	0.005	0.75	1.41
10°	**0.70**	1.37	0.01	**0.71**	**1.38**
15°	0.72	**1.34**	0.02	0.72	1.39

It can be found that these hyperparameters have slight impacts on the final reconstruction metrics, with the edge mask threshold δ and the neighbor radius r being more influential. Accordingly, we choose δ=0.05, r=0.1, θ=10° and μ=0.01 as our final hyperparameter settings, as detailed in our paper. These parameters are applied across all our experiments and have demonstrated strong robustness across various datasets.

E. Running Time

It takes 3 days to train EdgeMovingNet on single NVIDIA-2080Ti GPU using our dataset. Once trained, EdgeMovingNet's inference time for a single object is approximately 1.4s, and the refinement process takes about 3s on AMD Ryzen 5 5600X CPU in Python implementation. This is more convenient and efficient compared to methods that require training or learning priors for each object, such as PCP [7]. Our method is capable of serving as a tool for high quality point cloud reconstruction within limited resources while preserving edges.

F. More Results

We offer more reconstruction results of common objects in ModelNet [11] and ShapeNet [3] dataset. We make visual comparisons against Poisson [5], Voronoi [1], POCO [2], PCP [7], RFEPS [12], GeoUDF [9] and NKSR [4]. The results are illustrated in Fig. 2. Poisson [5] and RFEPS [12] require point normal as input, while other methods make reconstruction with only points.

References

[1] Dobrina Boltcheva and Bruno Lévy. Surface reconstruction by computing restricted voronoi cells in parallel. *Computer-Aided Design*, 90:123–134, 2017. 2, 4

[2] Alexandre Boulch and Renaud Marlet. Poco: Point convolution for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6302–6314, 2022. 2, 4

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 4

[4] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4369–4379, 2023. 2, 4

[5] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 2, 4

[6] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019. 2

[7] Baorui Ma, Yu-Shen Liu, Matthias Zwicker, and Zhizhong Han. Surface reconstruction from point clouds by learning predictive context priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6326–6337, 2022. 2, 4

[8] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[9] Siyu Ren, Junhui Hou, Xiaodong Chen, Ying He, and Wenping Wang. Geoudf: Surface reconstruction from 3d point clouds via geometry-guided distance representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14214–14224, 2023. 2, 4

[10] 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)*, 38(5):1–12, 2019. 1, 2

[11] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2, 4

[12] Rui Xu, Zixiong Wang, Zhiyang Dou, Chen Zong, Shiqing Xin, Mingyan Jiang, Tao Ju, and Changhe Tu. Rfeps: Reconstructing feature-line equipped polygonal surface. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022. 2, 4

[13] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2

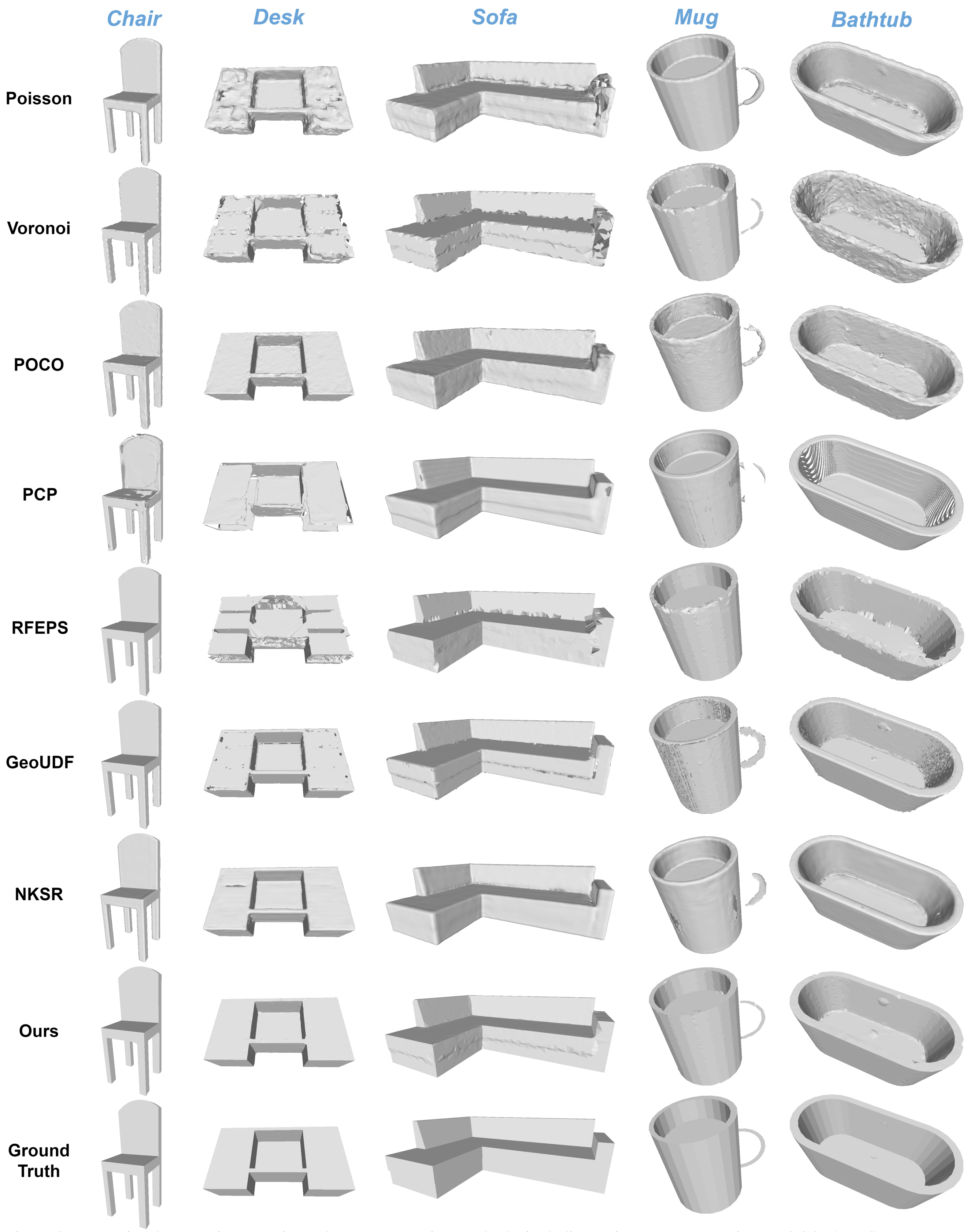

Figure 2. More visual comparisons against other reconstruction methods, including Poisson [5], Voronoi [1], POCO [2], PCP [7], RFEPS [12], GeoUDF [9] and NKSR [4]. Chair and bathtub are from ShapeNet [3] dataset. Desk, sofa and mug are from ModelNet [11] dataset.