Supplementary Material for "Towards Efficient Graph Convolutional Networks for Point Cloud Handling"

Yawei Li[1*], He Chen[2*], Zhaopeng Cui[3], Radu Timofte[1], Marc Pollefeys[1,4],
Gregory Chirikjian[2,5], Luc Van Gool[1,6]
[1]ETH Zürich, [2]Johns Hopkins University, [3]State Key Lab of CAD&CG, Zhejiang University,
[4]Microsoft, [5]National University of Singapore, [6]KU Leuven, Belgium
yawei.li@vision.ee.ethz.ch, hchen136@jhu.edu, zhpcui@gmail.com

In this supplementary, we first give the detailed proof of Theorem 1 in the main paper in Sec. 1. Then we describe the implementation details in Sec. 2. Sec. 3 shows the visualization of the features in the proposed network. Sec. 4 includes more experimental results.

1. Proofs

In this section, we provide the detailed proof of both the upper and lower bounds of Theorem 1.

Proof. **Upper bound.** For the simplicity of analysis, inner product and summation are selected as the edge function and the aggregation operation in **Theorem 1**. Thus, the theorem is derived under the assumption that the graph convolution has the following form

$$\mathbf{x}'_i = [\mathbf{x}'_{i1}, \cdots, \mathbf{x}'_{im}, \cdots, \mathbf{x}'_{iM}], \tag{1}$$

$$\mathbf{x}'_{im} = \sum_{k=1}^{K} < \boldsymbol{\theta}_m, \mathbf{x}_i^k >, \tag{2}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_M\}$ is the trainable parameters of the MLP with M output channels. Then the squared distance between two points $\mathbf{x}'_i$ and $\mathbf{x}'_j$ after the graph convolution is

$$\|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2 = \sum_{m=1}^{M} (\mathbf{x}'_{im} - \mathbf{x}'_{jm})^2 \tag{3}$$

$$= \sum_{m=1}^{M} (\sum_{k=1}^{K} < \boldsymbol{\theta}_m, \mathbf{x}_i^k > - \sum_{k=1}^{K} < \boldsymbol{\theta}_m, \mathbf{x}_j^k >)^2 \tag{4}$$

$$= \sum_{m=1}^{M} (\sum_{k=1}^{K} < \boldsymbol{\theta}_m, \mathbf{x}_i^k - \mathbf{x}_j^k >)^2 \tag{5}$$

$$\leq \sum_{m=1}^{M} K \sum_{k=1}^{K} < \boldsymbol{\theta}_m, \mathbf{x}_i^k - \mathbf{x}_j^k) >^2 \tag{6}$$

$$\leq K \sum_{m=1}^{M} \sum_{k=1}^{K} \|\boldsymbol{\theta}_m\|_2^2 \|\mathbf{x}_i^k - \mathbf{x}_j^k\|_2^2. \tag{7}$$

The inequality in Eqn. 6 follows that the arithmetic mean is not larger than the quadratic mean while the inequality in Eqn. 13 follows Cauchy–Schwarz inequality. Assume that the parameters $\boldsymbol{\theta}_m$ in the network are random variables that follows Gaussian distribution with 0 mean and σ^2 variance. Then the distance $\|\mathbf{x}'_i - \mathbf{x}'_k\|_2^2$ is also a random variable and the expectation is expressed as,

$$\mathbb{E}[\|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2] \leq \mathbb{E}[K \sum_{m=1}^{M} \sum_{k=1}^{K} \|\boldsymbol{\theta}_m\|_2^2 \|\mathbf{x}_i^k - \mathbf{x}_j^k\|_2^2] \tag{8}$$

$$= K \sum_{m=1}^{M} \sum_{k=1}^{K} \mathbb{E}[\|\boldsymbol{\theta}_m\|_2^2] \|\mathbf{x}_i^k - \mathbf{x}_j^k\|_2^2 \tag{9}$$

$$= \sigma^2 dKM \sum_{k=1}^{K} \|\mathbf{x}_i^k - \mathbf{x}_j^k\|_2^2, \tag{10}$$

where the term $\sum_k \|\mathbf{x}_i^k - \mathbf{x}_j^k\|_2^2$ is just the neighborhood distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. □

*Co-first author.

Proof. **Lower bound.** In Eqn. 5, let

$$\mathbf{a}_m = \sum_{k=1}^{K} < \boldsymbol{\theta}_m, \mathbf{x}_i^k - \mathbf{x}_j^k >. \tag{11}$$

Thus, Eqn. 5 become

$$\sum_{m=1}^{M} (\sum_{k=1}^{K} < \boldsymbol{\theta}_m, \mathbf{x}_i^k - \mathbf{x}_j^k >)^2 = \sum_{m=1}^{M} \mathbf{a}_m^2. \tag{12}$$

Using Cauchy-Schwarz inequality

$$\sum_{m=1}^{M} \mathbf{a}_m \mathbf{b}_m \leq \sqrt{\sum_{m=1}^{M} \mathbf{a}_m^2} \sqrt{\sum_{m=1}^{M} \mathbf{b}_m^2} \tag{13}$$

and letting $\mathbf{b}_m^2 = 1/M$, then the inequality in Eqn 13 becomes

$$\left(\frac{1}{\sqrt{M}} \sum_{m=1}^{M} \mathbf{a}_m \right)^2 \leq \sum_{m=1}^{M} \mathbf{a}_m^2. \tag{14}$$

Thus, the lower bound of Eqn 5 becomes

$$\|\mathbf{x}_i' - \mathbf{x}_j'\|_2^2 = \sum_{m=1}^{M} (\sum_{k=1}^{K} < \boldsymbol{\theta}_m, \mathbf{x}_i^k - \mathbf{x}_j^k >)^2 \tag{15}$$

$$\geq \frac{1}{M} \left(\sum_{m=1}^{M} \sum_{k=1}^{K} < \boldsymbol{\theta}_m, \mathbf{x}_i^k - \mathbf{x}_j^k > \right)^2 \tag{16}$$

$$= \frac{1}{M} < \sum_{m=1}^{M} \boldsymbol{\theta}_m, \sum_{k=1}^{K} \mathbf{x}_i^k - \mathbf{x}_j^k >^2 . \tag{17}$$

Let $\boldsymbol{\phi} = \sum_{m=1}^{M} \boldsymbol{\theta}_m$ and $\mathbf{z} = \sum_{k=1}^{K} \mathbf{x}_i^k - \mathbf{x}_j^k$. Then

$$\|\mathbf{x}_i' - \mathbf{x}_j'\|_2^2 \geq \frac{1}{M} < \boldsymbol{\phi}, \mathbf{z} >^2 \tag{18}$$

$$= \frac{1}{M} (\sum_{l=1}^{d} \boldsymbol{\phi}_l, \mathbf{z}_l)^2 \tag{19}$$

$$= \frac{1}{M} \sum_{l=1}^{d} \sum_{n=1}^{d} \boldsymbol{\phi}_l \boldsymbol{\phi}_n \mathbf{z}_l \mathbf{z}_n. \tag{20}$$

Then taking the expectation on both sides, the inequality becomes

$$\mathbb{E}[\|\mathbf{x}_i' - \mathbf{x}_j'\|_2^2] \geq \mathbb{E}[\frac{1}{M} \sum_{l=1}^{d} \sum_{n=1}^{d} \boldsymbol{\phi}_l \boldsymbol{\phi}_n \mathbf{z}_l \mathbf{z}_n] \tag{21}$$

$$= \frac{1}{M} \sum_{l=1}^{d} \sum_{n=1}^{d} \mathbb{E}[\boldsymbol{\phi}_l \boldsymbol{\phi}_n] \mathbf{z}_l \mathbf{z}_n. \tag{22}$$

Note that the elements of $\boldsymbol{\theta}_m$ follow independent Gaussian distribution with 0 mean and σ^2 variance and $\boldsymbol{\phi} = \sum_{m=1}^{M} \boldsymbol{\theta}_m$. Thus, the elements of $\boldsymbol{\phi}$ follows independent Gaussian distribution with 0 mean and $M\sigma^2$ variance. Thus,

$$\mathbb{E}[\boldsymbol{\phi}_l \boldsymbol{\phi}_n] = \begin{cases} 0 & l \neq n \\ M\sigma^2 & l = n \end{cases}. \tag{23}$$

Thus, the lower bound becomes

$$\mathbb{E}[\|\mathbf{x}_i' - \mathbf{x}_j'\|_2^2] \geq \frac{1}{M} \sum_{l=1}^{d} M\sigma^2 \mathbf{z}_l^2 \tag{24}$$

$$= \sigma^2 \|\mathbf{z}\|_2^2 \tag{25}$$

$$= \sigma^2 \| \sum_{k=1}^{K} \mathbf{x}_i^k - \mathbf{x}_j^k \|_2^2 \tag{26}$$

$$= \sigma^2 K^2 \| \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_i^k - \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_j^k \|_2^2. \tag{27}$$

Thus, the distance between two points after graph convolution is lower bounded by the neighborhood centroid distance of the corresponding points before graph convolution up to a scaling factor.

□

2. Implementation Details

2.1. Classification of Point Cloud

For classification task, we used ModelNet40 dataset. ModelNet40 dataset consists of 12,311 meshed CAD models of 40 categories. We follow the experimental setting of PointNet [4, 5] and DGCNN [6]. In order to evaluate our performance, we uniformly sample points of different numbers from the mesh faces to formulate point clouds. The number of parameters of all the networks is 1.8k. Inference runtime is measured on a single Titan Xp GPU and the batch size is reduced to 16 for running with 2048 points.

2.2. Segmentation of Point Cloud

For part segmentation task, we used ShapeNetPart dataset. In ShapeNetPart, there are 16 object categories and 16,881 3D shapes, annotated with 50 parts. 2048 points are sampled from each shape. For semantic segmentation task, we used Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS). S3DIS consists of indoor scenes of 272 rooms in six indoor areas, annotated with 13 semantic categories. Our experiments follow the standard training, validation, and test split of DGCNN. Part segmentation experiments are run on two Titan Xp GPUs and the batch size is 32.

2.3. Surface Reconstruction of Point Cloud

The visualization of meshes is done on the platform Open3D [8].

3. Feature Visualization

In order to validate that the neighborhood geometric features are preserved after all the operations and acceleration strategies, experiments are designed by extracting and visualizing the feature map as a distance colormap rendered on the 3D point cloud. The evolution of feature space *w.r.t.* the number of epochs is shown in Fig. 1. The local structures are preserved and converge to smaller and smaller regions as the network propagates. Gradually, the yellow points (relatively nearer points in the latent space) all lie in the green dot region(KNN), this proves the effectiveness of the local feature extraction is kept by the accelerated network. By observing the figure, we can come to the conclusion that such convergence trends to grow as the iterations move on. At Epoch 250 when the loss of the classification neural network converges, the yellowish neighbor features also converge into a very small region, and this region is smaller than the KNN represented by the green points. Fig. 2 shows more results of feature space for point cloud classification on ModelNet40. Fig. 3 shows more results of feature space for part segmentation task. The convention is the same as Fig. 8 of the main paper.

4. Additional Experimental Results

In this section, we show additional experimental results. Fig. 4 shows the computational resource comparison for the task of semantic segmentation. The accelerated network is more efficient than the original network in terms of inference time, GPU memory consumption, and computational complexity. A study of how a wide range of K and P values (defined in Sec. 4.2) affects performance is carried out in Fig. 6 for point cloud classification. Fig. 7 presents the ablation study of performance for semantic segmentation *w.r.t.* a wide range of parameters. The final parameters K and P are selected according to the ablation study. Fig. 5 shows more qualitative results for surface reconstruction. As shown in the figure, the accelerated network leads to reconstructed surfaces similar to that from the original network.

5. Extension to other tasks.

The proposed method in this paper could be applied to other tasks such as semi-supervised learning. Semi-supervised learning can be regarded as a research direction parallel to efficient computation. It aims at using fewer samples or labels for learning on point clouds [2, 1, 3, 7]. And the core problem is the design of the constraints and losses, and label propagation *etc.*Most of the semi-supervised methods are model-agnostic. PointNet and DGCNN are the commonly used backbone networks. Thus, it is straightforward to replace backbones with ours while keeping the losses and training protocols.

References

[1] Mingmei Cheng, Le Hui, Jin Xie, and Jian Yang. Sspc-net: Semi-supervised semantic 3d point cloud segmentation network. *arXiv preprint arXiv:2104.07861*, 2021. 3

[2] Jilin Mei, Biao Gao, Donghao Xu, Wen Yao, Xijun Zhao, and Huijing Zhao. Semantic segmentation of 3d lidar data in dynamic scene using semi-supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2496–2509, 2019. 3

[3] Omid Poursaeed, Tianxing Jiang, Han Qiao, Nayun Xu, and Vladimir G Kim. Self-supervised learning of point clouds via orientation estimation. In *International Conference on 3D Vision*, pages 1018–1028. IEEE, 2020. 3

[4] 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. 2

[5] 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*, pages 5099–5108, 2017. 2

[6] 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. 2

[7] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *Proc. CVPR*, pages 13706–13715, 2020. 3

[8] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 2

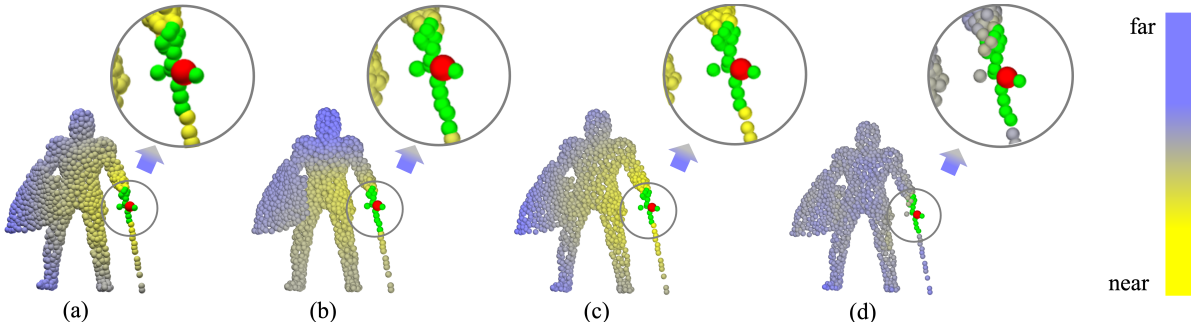


Figure 1: Qualitative result on Modelnet40. (a), (b) Input space and last-layer feature space rendered as colormap between the red point and the rest points at epoch 0. The green points are KNN of the red point. (c), (d) follow the same layout with (a), (b) at epoch 250.

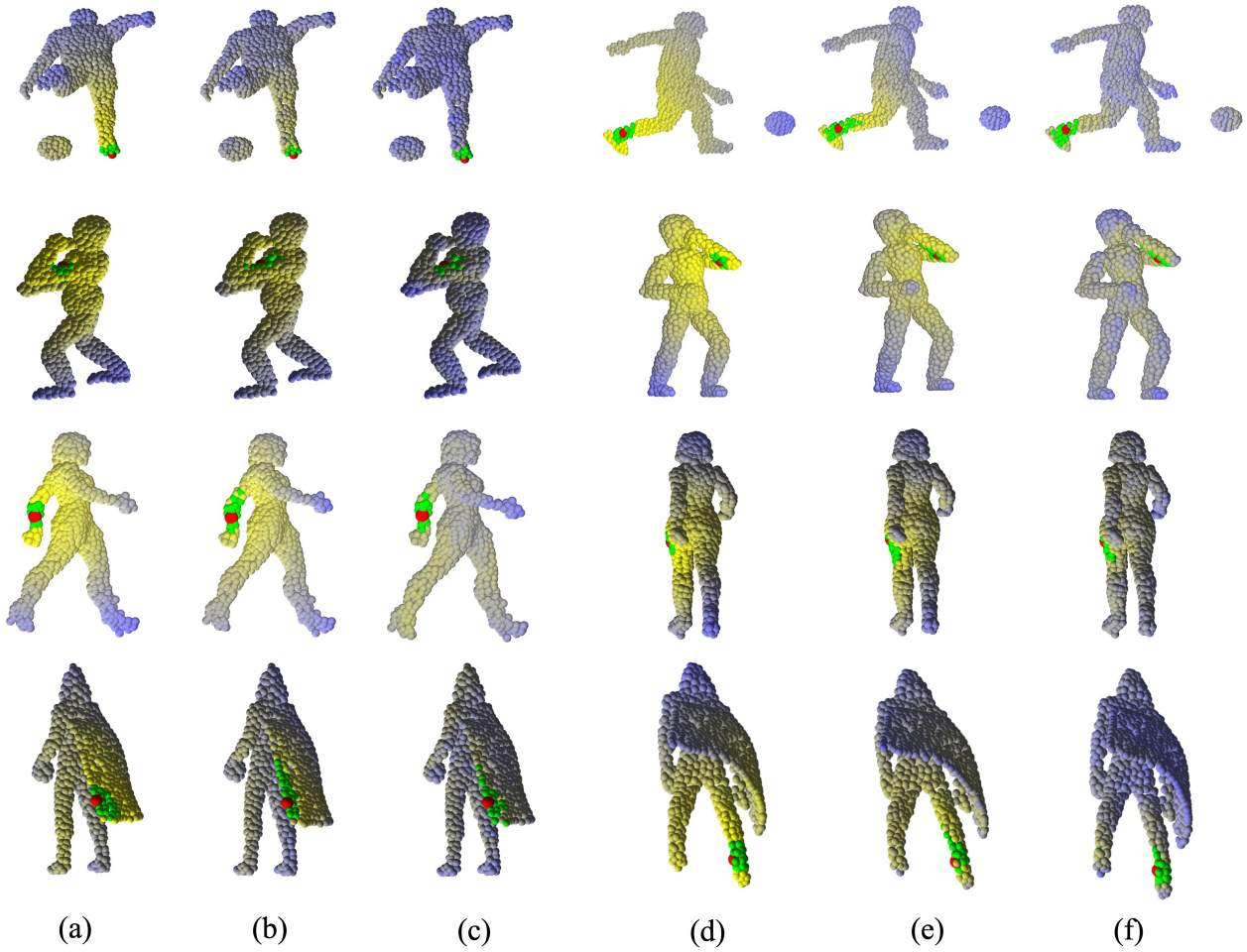
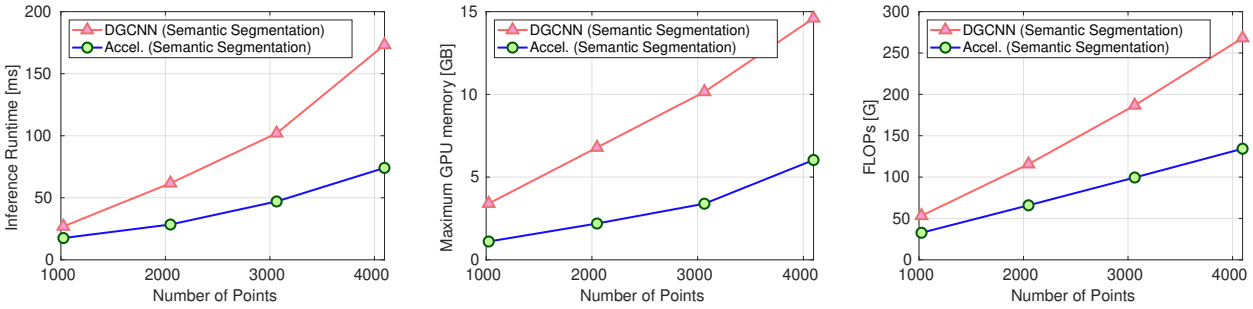


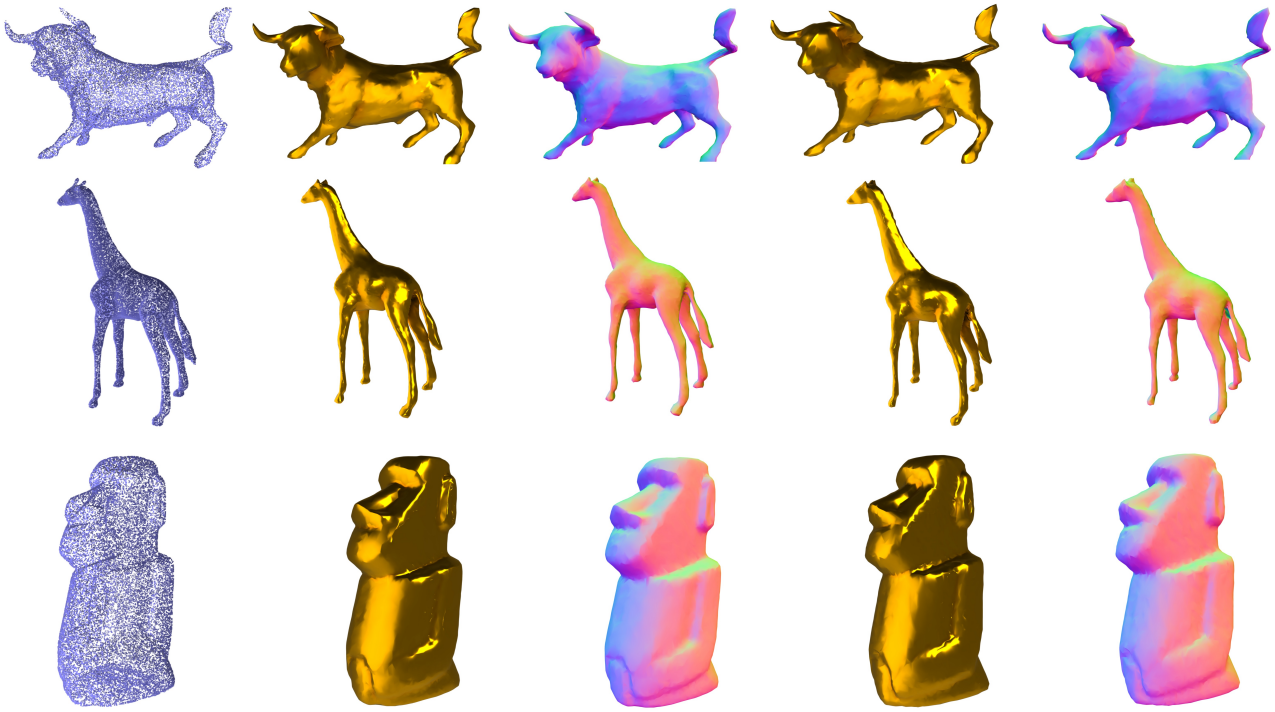
Figure 2: Renderings of input space and feature space as colormap between the red point and the rest of the points on ModelNet40 dataset. The green points represent KNN of the red point. (a) represents the input space. (b) represents the feature space extracted from the second layer of the network. (c) represents the feature space extracted from the last layer of the network. (d), (e),and (f) respectively follows the same layout with (a), (b), and (c).

Figure 3: Visualization of the point distance across the accelerated network for part segmentation task. The distance of points to the red point in the figures is computed. Lighter color means closer distance. (a) The input shape. (b) Distance between points in the raw data. (c)-(e) Distance between point in the feature space from Layer 1, Layer 2, Layer 3 of the accelerated network. (f) Segmentation result. The accelerated network could still capture long-range dependency between points.



(a) Runtime (b) GPU memory (c) FLOPs

Figure 4: Comparison between DGCNN and the accelerated version on point cloud semantic segmentation. (a) Runtime, (b) GPU memory consumption, and (c) FLOPs are reported for comparison.The proposed method can achieve significant reduction of computation resources.



(a) (b) (c) (d) (e)

Figure 5: Qualitative results of surface reconstruction. (a) Input point cloud. (b), (c) Surface and normal map reconstructed by Point2Mesh. (d), (e) Surface and normal map reconstructed by our method.

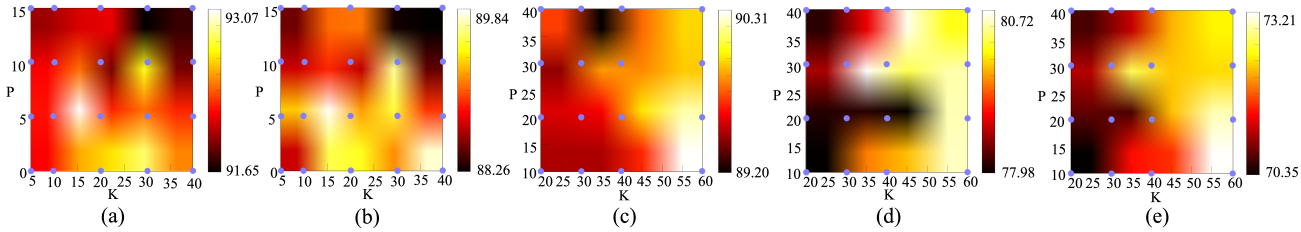
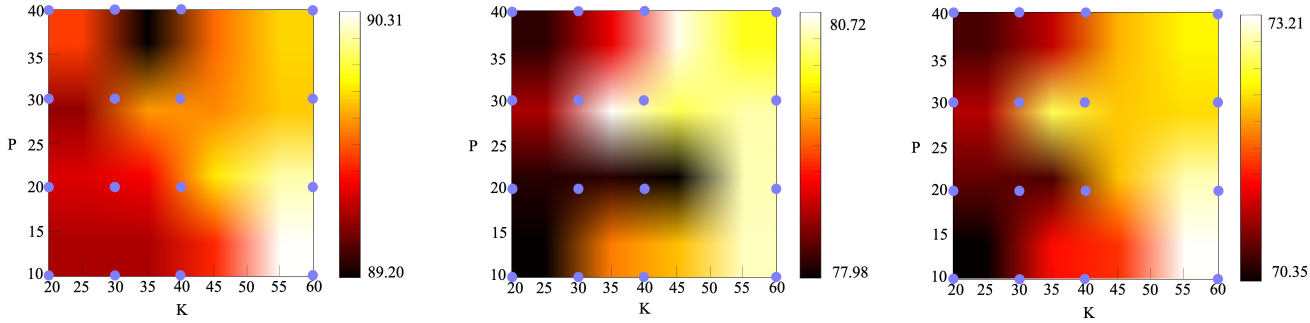


Figure 6: Ablated experimental result of different hyper-parameter choices.(a) Ablation study of overall acc. *w.r.t* parameters K and P for classification task. Values calculated are the points on the grid, and the hotmap is derived by bilinear interpolation. (b) follows the same layout as (a) for balanced acc. of classification task. (c), (d), (e) resp. follow the same layout as (a) for overall acc., balanced acc., and mean IoU of segmentation task.



(a) Heatmap of overall acc. *w.r.t.* parameters K and P.

(b) Heatmap of balanced acc. *w.r.t.* parameters K and P.

(c) Heatmap of mIoU *w.r.t.* parameters K and P.

Figure 7: (a) Ablation study of overall acc. *w.r.t* parameters K and P. Values calculated are the points on the grid, and the hotmap is derived by bilinear interpolation. (b) and (c) follows the same layout with (e) for balanced accuracy and mean IoU.