MortonNet: Self-Supervised Learning of Local Features in 3D Point Clouds (Supplementary Material)

Ali Thabet* Humam Alwassel* Bernard Ghanem
King Abdullah University of Science and Technology (KAUST), Saudi Arabia
{ali.thabet,humam.alwassel,bernard.ghanem}@kaust.edu.sa

1. Z-order Curves

To create a Z-ordered Space Filling Curve, we first compute a Morton-order value for each coordinate in a point cloud. We use Morton-order values to sort the points in the point cloud. To compute a Morton-order value for an integer set of coordinates, we first convert each value to binary and then interleave the bits of each coordinate. Following is a simple example [1]:

$$(x, y, z) = (5, 9, 1) = (0101, 1000, 0001) \tag{1}$$
$$\text{Morton-order} = 0100010001111 \tag{2}$$

There are numerous efficient ways of computing Morton-orders, some of them with constant time complexity [1].

2. PointNet + MortonNet Architecture

In experiments 1 and 2, we use Morton features fused into PointNet [3], in order to perform semantic segmentation of S3DIS and vKITTI. Figure 1 shows the final architecture of PointNet fused with Morton features.

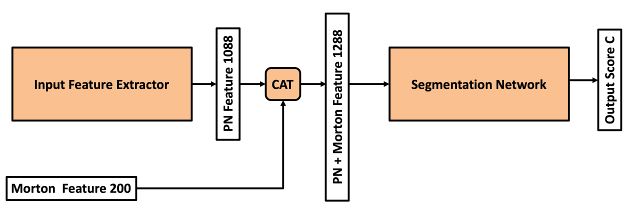

Figure 1. **PointNet + MortonNet**: We fuse Morton features into PointNet [3] by concatenating them with with the output of the feature extractor, just before the point classification.

3. RSNet + MortonNet Architecture

In experiments 1 we also use Morton features fused into RSNet [2], in order to perform semantic segmentation of S3DIS. Figure 2 shows the final architecture of RSNet fused with Morton features.

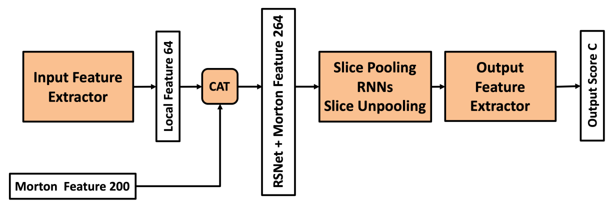

Figure 2. **RSNet + MortonNet**: RSNet [2] extract point features, and feeds these features into a series of RNNs, where the inputs are organized into structured slices. We concatenate Morton features to the output of the feature extractor and feed the resulting features to the slicing and recurrent layers.

4. MortonNet for ShapeNet Part Segmentation

Figure 3 shows the simple classifier architecture we employ to transfer our Morton features to the part segmentation task on ShapeNet. Our classifier operates directly on Morton features. The network consists of 4 convolution layers, the first 3 followed by batch normalization and ReLU. The final convolution layer outputs a score for each class.

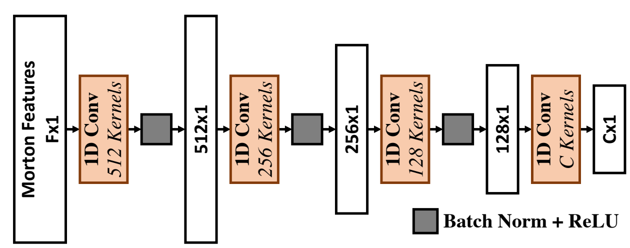

Figure 3. **MortonNet for Part Segmentation**: The architecture of the classifier we use on top of our Morton features to preform point-based 3D tasks such as part segmentation. F = Morton feature size; C = Number of parts/classes.

5. Part Segmentation Qualitative Results on ShapeNet [4].

Figure 4 shows qualitative results for our method, Morton features + simple classifier. We show results for all the object categories of the dataset.

*indicates equal contribution.

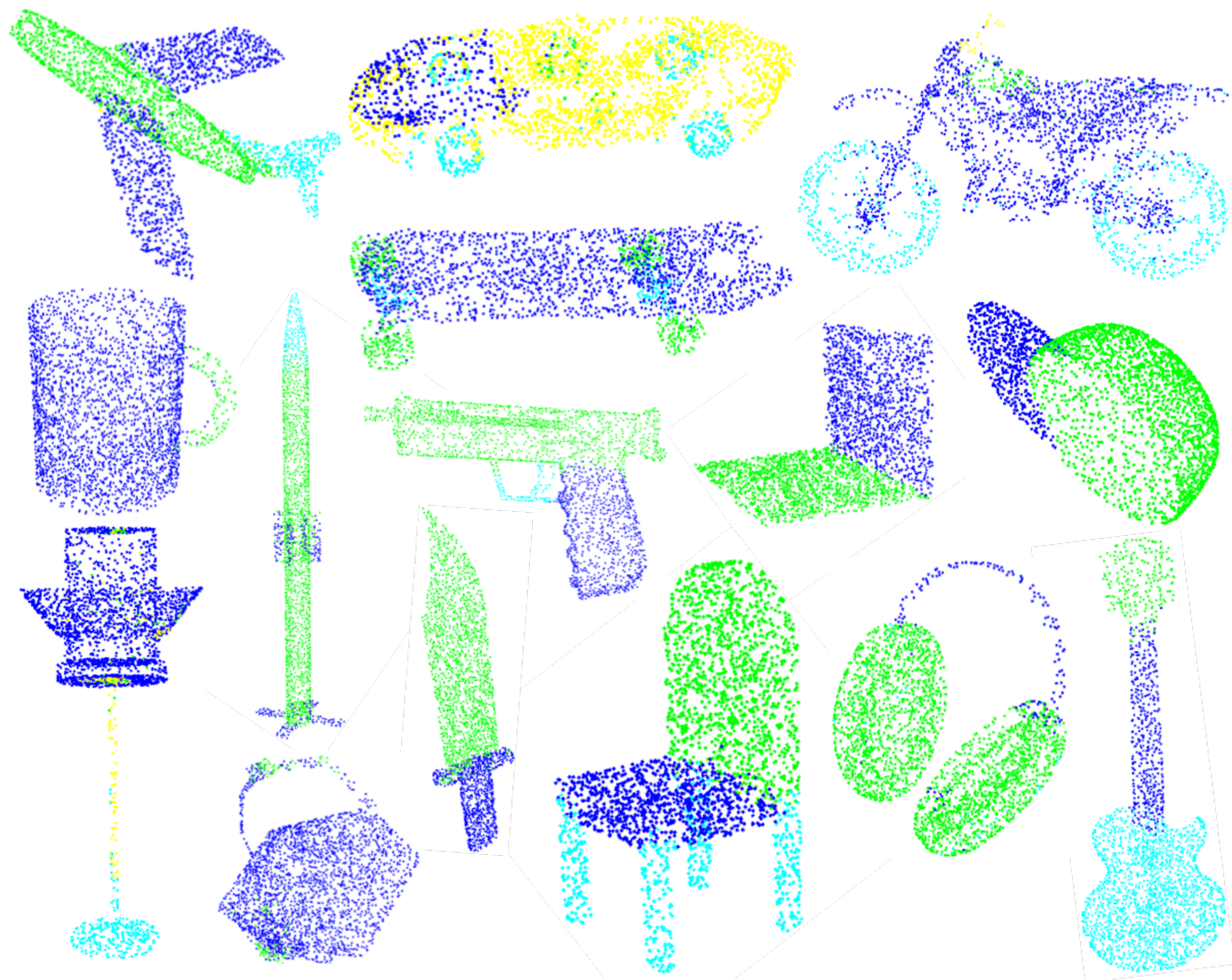

Figure 4. **ShapeNet Qualitative Results**: Part segmentation results of MortonNet on each shape category in ShapeNet [4].

References

[1] J. Baert. Libmorton: C++ morton encoding/decoding library. `https://github.com/Forceflow/libmorton`, 2018.

[2] Q. Huang, W. Wang, and U. Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2635, 2018.

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

[4] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016.