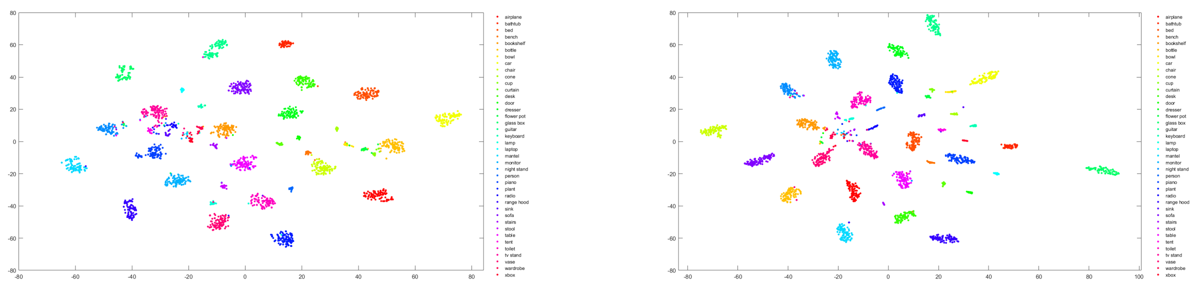
View-GCN: View-based Graph Convolutional Network for 3D Shape Analysis
Supplementary Material

Xin Wei, Ruixuan Yu and Jian Sun (✉)
Xi'an Jiaotong University, Xi'an, 710049, China
{wxmath,yuruixuan123}@stu.xjtu.edu.cn, jiansun@xjtu.edu.cn

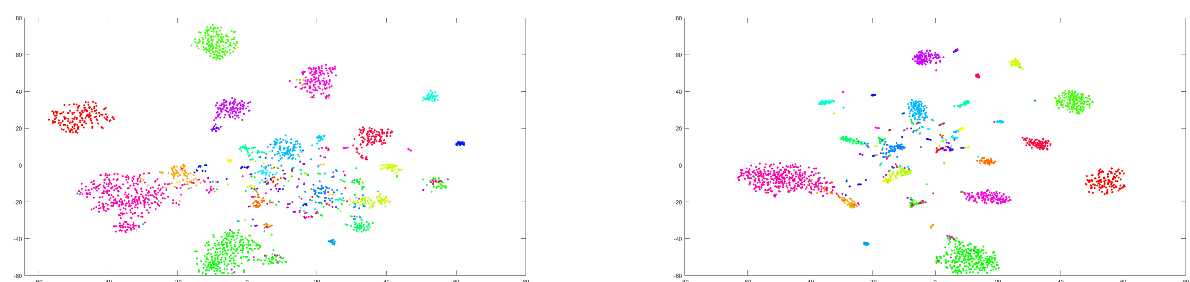
In this supplementary material, we provide more visualization results for our learned shape features in Figures 1 & 2, the shape classification results in Figure 3, and the selected views on hierarchical view-graphs in Figure 4.



(a) MVCNN-new (95.0%)

(b) view-GCN (97.6%)

Figure 1. Visualization of shape features learned by MVCNN-new (a) and view-GCN (b) using t-SNE on ModelNet-40 dataset. The learned features by MVCNN-new and view-GCN achieve 95.0% and 97.6% per instance accuracies respectively on the dataset. MVCNN-new is our baseline algorithm that directly max-pools the view features.



(a) MVCNN-new (88.9%)

(b) view-GCN (90.9%)

Figure 2. The t-SNE visualization of shape features learned by MVCNN-new (a) and view-GCN (b) on ShapeNet Core55 dataset. The learned features by MVCNN-new and view-GCN achieve 88.9% and 90.9% per instance accuracies respectively on the dataset. MVCNN-new is our baseline algorithm that directly max-pools the view features.

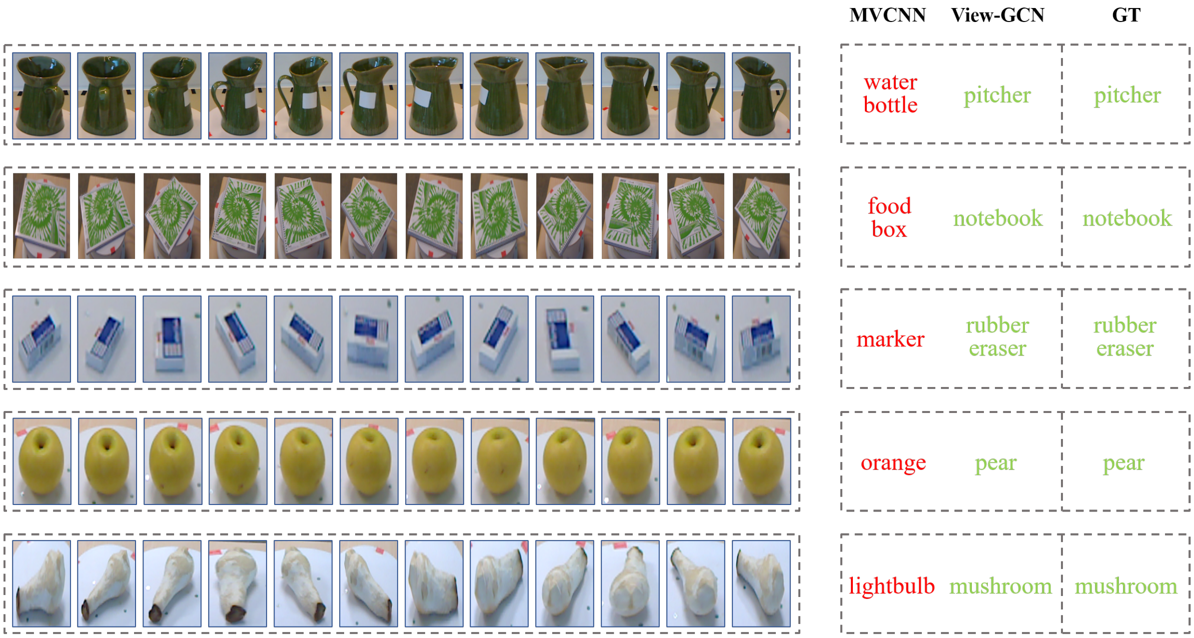
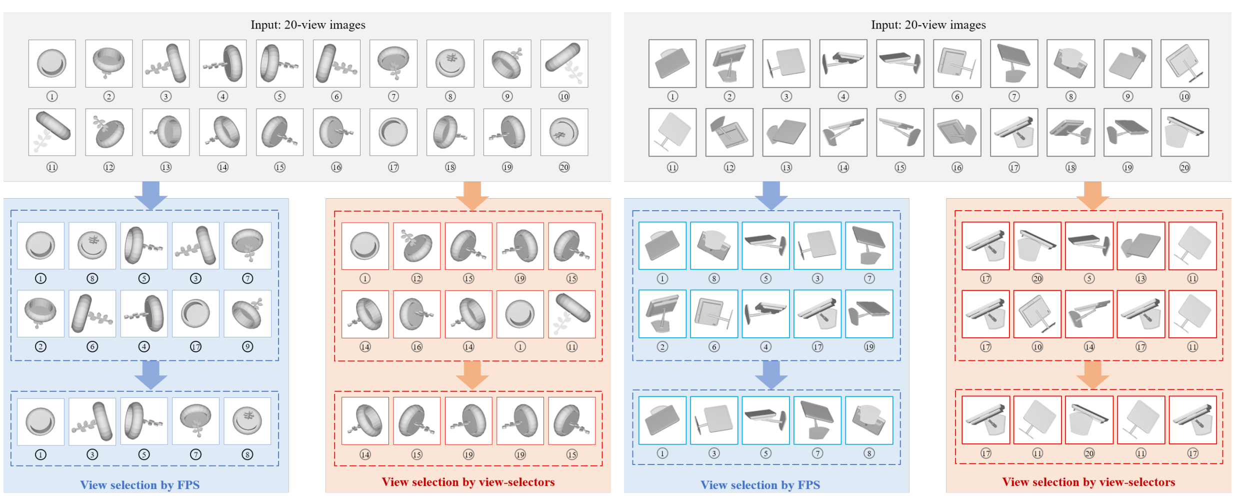


Figure 3. Examples of classification results by MVCNN and our view-GCN on RGBD dataset. In the rows of left sub-figure, we present the 12-view images of the shapes captured by real cameras. In the right sub-figure, we present the predictions by MVCNN and view-GCN, together with ground-truth labels. On the RGBD dataset, view-GCN achieves 91.9% per instance accuracy compared with 86.1% achieved by MVCNN.



(a) (b)

Figure 4. Illustration of selected views for constructing hierarchical view-graphs by FPS and our view-selector. (a) View selection by FPS (blue) and our view-selector (orange) for an object of flower pot. (b) View selection by FPS (blue) and our view selector (orange) for an object of monitor. For different shapes, our view selector can sample the views adaptively, while FPS takes fixed views. The input objects are with 20 views, and coarsened twice to 10 and 5 views to build coarsened view-graphs. Our sampled views using view-selectors produce better results as discussed in experiments.