

# Volumetric and Multi-View CNNs for Object Classification on 3D Data

## Supplementary Material

Charles R. Qi\* Hao Su\* Matthias Nießner Angela Dai Mengyuan Yan Leonidas J. Guibas  
Stanford University

### 1. Details on Model Training

**Training for Our Volumetric CNNs** To produce occupancy grids from meshes, the faces of a mesh are subdivided until the length of the longest edge is within a single voxel; then all voxels that intersect with a face are marked as occupied. For 3D resolution 10,30 and 60 we generate voxelizations with central regions 10, 24, 54 and padding 0, 3, 3 respectively.

This voxelization is followed by a hole filling step that fills the holes inside the models as occupied voxels.

To augment our training data with azimuth and elevation rotations, we generate 60 voxelizations for each model, with azimuth uniformly sampled from  $[0, 360]$  and elevation uniformly sampled from  $[-45, 45]$  (both in degrees).

We use a Nesterov solver with learning rate 0.005 and weight decay 0.0005 for training. It takes around 6 hours to train on a K40 using Caffe [2] for the subvolume supervision CNN and 20 hours for the anisotropic probing CNN. For multi-orientation versions of them, SubvolumeSup splits at the last conv layer and AniProbing splits at the second last conv layer. Volumetric CNNs trained on single orientation inputs are then used to initialize their multi-orientation version for fine tuning.

During testing time, 20 orientations of a CAD model occupancy grid (equally distributed azimuth and uniformly sampled elevation from  $[-45, 45]$ ) are input to MO-VCNN to make a class prediction.

**Training for Our MVCNN and Multi-resolution MVCNN** We use Blender to render 20 views of each (either ordinary or spherical) CAD model from azimuth angles in  $0, 36, 72, \dots, 324$  degrees and elevation angles in  $-30$  and  $30$  degrees. For sphere rendering, we convert voxelized CAD models into meshes by replacing each voxel with an approximate sphere with 50 faces and diameter length of the voxel size. Four fixed point light sources are used for the ray-tracing rendering.

We first finetune AlexNet [3] with rendered images for ordinary rendering and multi-resolucional sphere renderings separately. Then we use trained AlexNet to initialize the

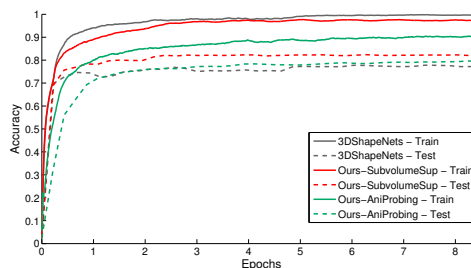


Figure 1: Learning curve of 3DShapeNets (end-to-end), and our Subvolume Supervision and Anisotropic Probing networks.

MVCNN and fine tune on multi-view inputs.

**Overfitting in Volumetric CNN Training** In Sec 4.2 of main paper we mentioned that training the volumetric CNN proposed by 3DShapeNets [5] in an end-to-end fashion is prone to overfitting. For plotting this curve, we train on ModelNet40 train set and report average class accuracy as test accuracy. In Fig 1, we see strong overfitting using 3DShapeNets architecture. Also we notice that, although overfitting still exists during training of our volumetric networks, the problem is greatly mitigated. In the end, our networks achieve both smaller train-test curve gap and higher test accuracies. Note that these numbers are from classification results with single orientation of shapes; methods like anisotropic probing can greatly improve the accuracy by adding orientation pooling.

**Other Volumetric Data Representations** Note that while we present our Volumetric CNN methods using occupancy grid representations of 3D objects, our approaches easily generalize to other volumetric data representations. In particular, we have also used Signed Distance Functions and (unsigned) Distance Functions as input (also  $30 \times 30 \times 30$  grids). Signed distance fields were generated through virtual scanning of synthetic training data, using volumetric fusion [1] (for our real-world reconstructed models, this is

the natural representation); distance fields were generated directly from the surfaces of the models. Performance was not affected significantly by the different representations, differing by around 0.5% to 1.0% for classification accuracy on ModelNet test data.

## 2. Real-world Reconstruction Test Data

In order to evaluate our method on real scanning data, we obtain a dataset of 3D models, which we reconstruct using data from a commodity RGB-D sensor (ASUS Xtion Pro). To this end, we pick a variety of real-world objects for which we record a short RGB-D frame sequence (several hundred frames) for each instance. For each object, we use the publicly-available Voxel Hashing framework in order to obtain a dense 3D reconstruction. In a semi-automatic post-processing step, we segment out the object of interest's geometry by removing the scene background. In addition, we align the obtained model with the world up direction. Overall, we obtained scans of 243 objects, comprising of a total of over XYZ thousand RGB-D input frames.

## 3. More Retrieval Results

For model retrieval, we extract CNN features (either from 3D CNNs or MVCNNs) from query models and find nearest neighbor results based on L2 distance. Similar to MVCNN (Su et al.) [4], we use a low-rank Mahalanobis metric to optimize retrieval performance. Figure 2 and Figure 3 show more examples of retrieval from real model queries.

## References

- [1] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.
- [2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [4] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV, to appear*, 2015.
- [5] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. 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.

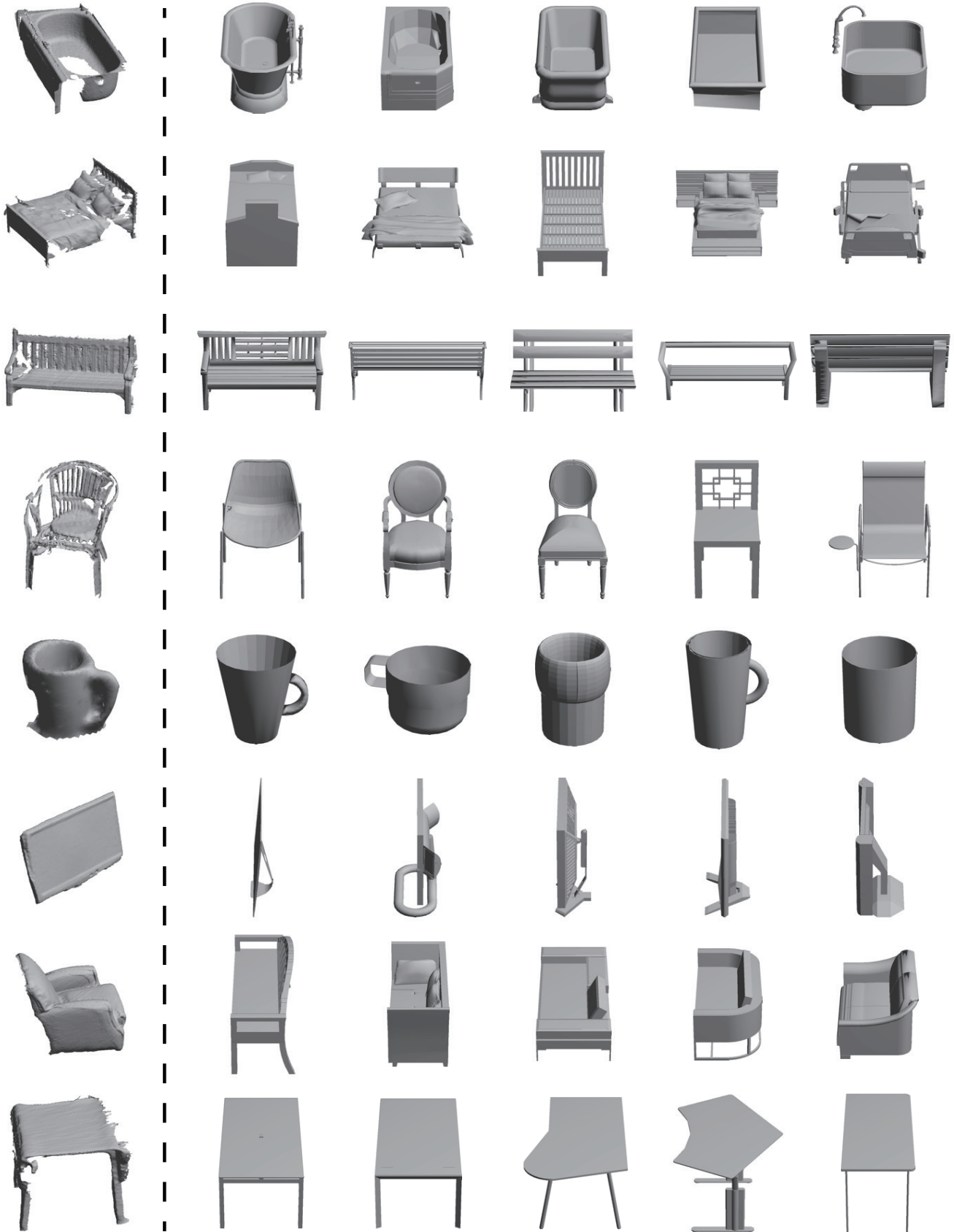


Figure 2: More retrieval results. Left column: queries, real reconstructed meshes. Right five columns: retrieved models from ModelNet40 Test800.

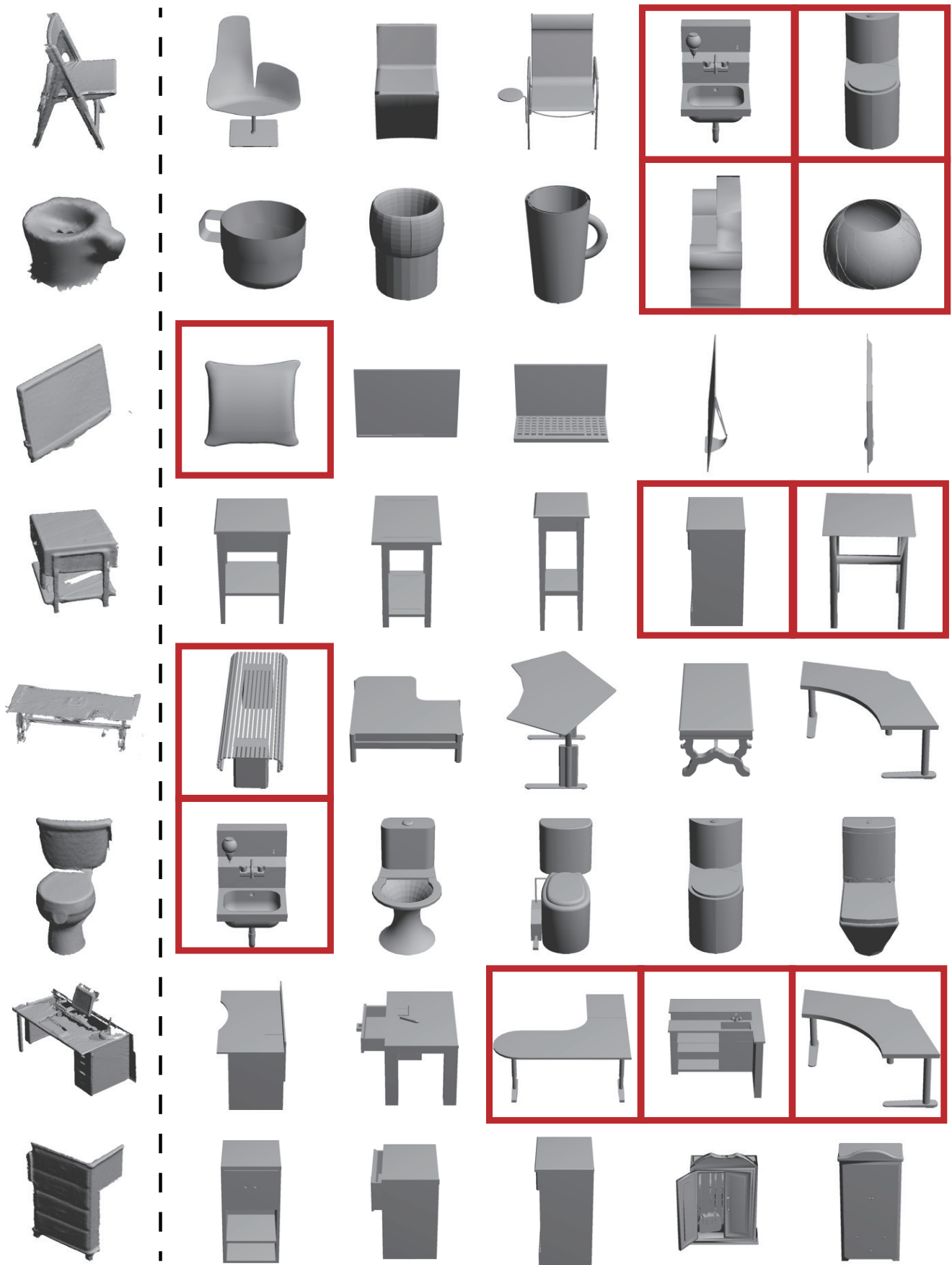


Figure 3: More retrieval results (samples with mistakes). Left column: queries, real reconstructed meshes. Right five columns: retrieved models from ModelNet40 Test800. Red bounding boxes denote results from wrong categories.











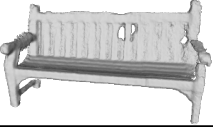
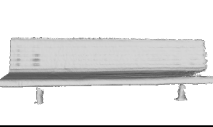
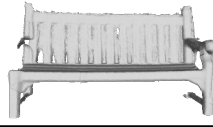

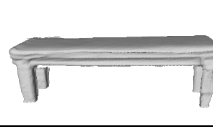


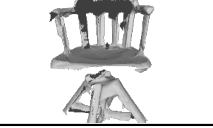



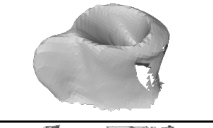
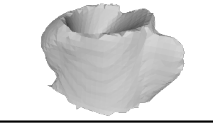
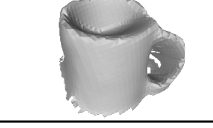
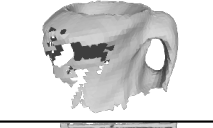
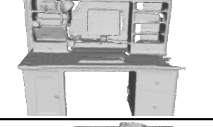
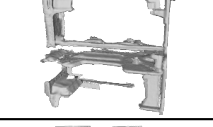
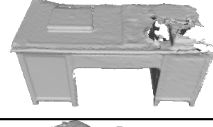
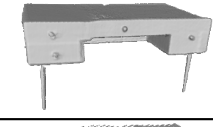



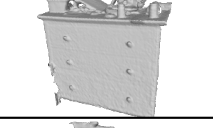

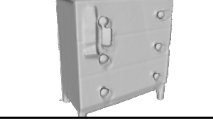
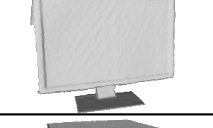




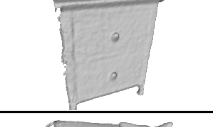



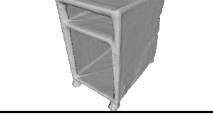





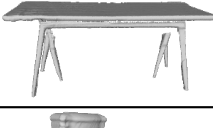



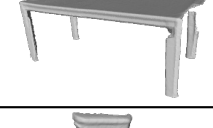



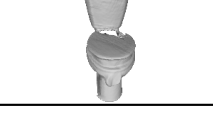

	Class	# of Models					
Bathtub		7					
Bed		27					
Bench		19					
Chair		17					
Cup		18					
Desk		16					
Dresser		12					
Monitor		45					
Night-stand		21					
Sofa		26					
Table		18					
Toilet		17					

Figure 4: Our real-world reconstruction test dataset, comprising 12 categories and 243 models. Each row lists a category along with the number of objects and several example reconstructed models in that category.