DeepPointSet: A Point Set Generation Network for 3D Object Reconstruction from a Single Image Supplementary Material

Haoqiang Fan *
Institute for Interdisciplinary
Information Sciences
Tsinghua University
fanhqme@gmail.com

Hao Su* Leonidas Guibas
Computer Science Department

Stanford University
{haosu,guibas}@cs.stanford.edu

A. Structure of supplementary material

Our supplementary material includes the following content:

- Further validation of our method on single view 3D reconstruction;
- The VAE formulation for building a conditional shape sampler and representative results (Sec 4.4 of main paper);
- Details of training, including network parameters and post-processing method (Sec 5.2 of main paper).

B. More results on single view 3D reconstruction

B.1. More results on validation set

We plot the reconstruction results of the first 5 mini-batches (160 cases in total) of our validation set at the end of this paper (see Fig 6). Results produced by the network trained by CD and EMD are compared side-by-side. Owing to the diversity in the ShapeNet dataset, our system is able to handle a variety of object types.

B.2. Analysis of human ability for single view 3D reconstruction

We conducted human study to provide reference to our current CD and EMD values reported on the rendered dataset. We provided the human subject with a GUI tool to create a triangular mesh from the image. The tool (see Fig 1) enables the user to edit the mesh in 3D and to align the modeled object back to the input image. In total 16 models are created from the input images of our validation set. $N = 1024$ points are sampled from each model.

*equal contribution

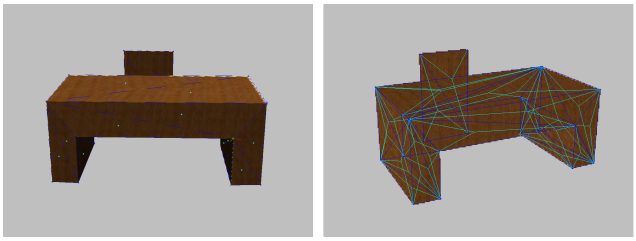

Figure 1. GUI tool used to manually model the objects. The user can change view point, edit vertex positions and connectivity in the 3D view (right). We also overlaid a wire-frame rendering of the object on the input image (left) to facilitate alignment.

As shown in Fig 2, both the EMD and the CD values of the network's reconstruction are on par with human's manual creation for most of the cases. We observed that the human subject mainly used cues of gravity direction (legs of chairs should touch the ground) and symmetry to infer the object's shape. As illustrated in input image number 4, 9 and 15, when the object is partially occluded (the table blocks the chair), ambiguous (it is unclear whether the can has a bottom) or manifests inadequate geometric cues (the guitar has non-polygonal shape and does not sit on the ground) the human subject performs poorly. The neural network trained by EMD performs reasonably well under both metrics. However, because CD emphasises only on the best matching point, the network trained by CD does not always produce predictions of uniform density and suffers high EMD value in some cases.

B.3. Analysis of failure cases

We visualize representative failure cases of our method on our rendered validation set. There are two trends, each exemplified by one input case in Fig 3. In the first kind of failure cases, the neural network is presented with a shape that it has completely no idea about. Then the networks

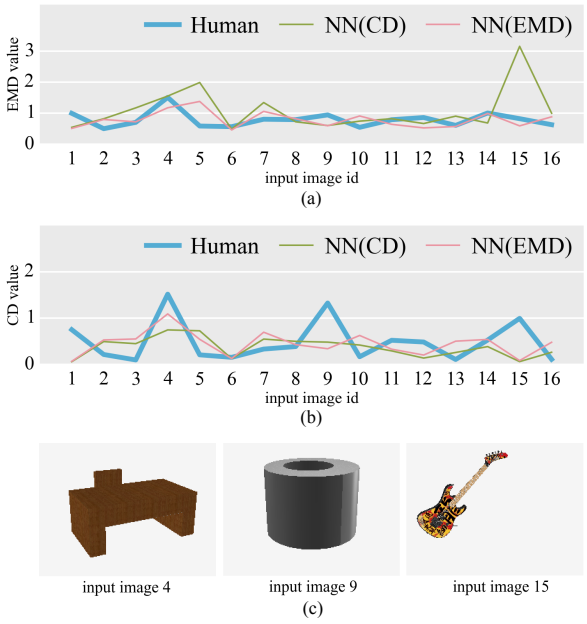

Figure 2. Comparison of reconstructions generated by the human subject, the neural network trained with CD and the neural network trained with EMD on 16 input images in the validation set. (a) Comparison of EMD value. (b) Comparison of CD value. (c) Input images numbered 4, 9 and 19 on which the human subject performs poorly.

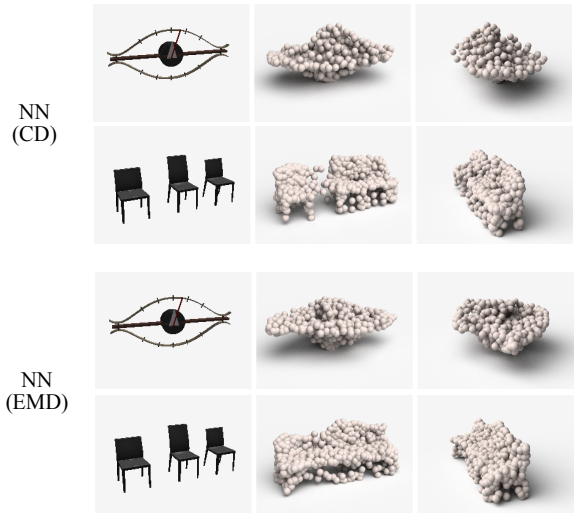

Figure 3. Examples of failure cases of our method on the validation set. Top: results of the neural network trained by CD. Bottom: results of the neural network trained by EMD. Both networks give unsatisfactory results.

tried to explain the input by something similar (a plane without wings?) but fundamentally wrong. In the second kind of failure cases, the neural network sees a composition of multiple objects. Because we have not implemented any detection or attention mechanism, the networks produce distorted output.

Figure 4. Result obtained by VAE training. Top: half-side view; middle: side view; bottom: back view.

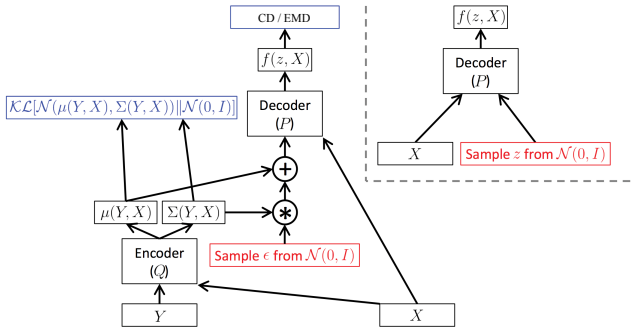

Figure 5. Network for conditional variational autoencoder shape sampler $P(S|X)$. Left: a training-time conditional variational autoencoder implemented as a feedforward neural network. Here, Y is the volumetric form of the groundtruth shape S, whereas $f(z, X)$ is the point cloud form of the predicted shape for S. Right: the same model at test time. (Modified from Doersch et al. [1])

C. The VAE formulation and representative results

As mentioned in Sec 4.4 of main paper, an alternative way to achieve the conditional shape sampler is by a conditional variational autoencoder. For more details about variational autoencoders, please refer to [1]. Fig 5 shows the system architecture for training and testing a conditional variational autoencoder $P(S|X)$ in our case. Here, X is the input image and S is the *point cloud* representation of the groundtruth 3D shape. At training time, each input image X will be augmented by a random variable that is conditioned on Y, which takes the *volumetric* representation of the groundtruth shape S. A 3D convolutional network is used as the encoder Q (see [2] for a good reference of 3D conv networks). Therefore, a local proximity in the embedding space contains the variations of possible groundtruth 3D shapes.

In Fig 4, we visualize the results of VAE. Compared to the result of Mo2 (see the main paper), the prediction of VAE looks plumper; however, it also captures the local directions of ambiguity in the shape.

D. Implementation details

D.1. Network parameter and training

Our network works on input images of 192x256. The deconv branch produces 768 points, which correspond to a 32x24 three-channel image. The fully connected branch produces 256 points. The convolutional layer has 16 feature maps in the highest resolution, and the number of channels are doubled after each decrease in resolution. We use strided convolution instead of max-pooling to increase speed. The training program is implemented in TensorFlow. 300000 gradient steps are taken, each computed from a minibatch of 32. Adam is used as the optimizer. We observed that the training procedure is smooth even without batch normalization. All activation functions are relu.

D.2. Post processing

We use a local method to post process the point cloud into a volumetric representation. First, the point cloud is registered into the 32x32x32 grid with bilinear interpolation. This can be think of as interpreting the points as 1x1x1 cubes and averaging the intersection volume with each grid cell (the occupancy representation). Then each voxel exams a local neighborhood to determine the final value. We implement this as a trained 3D convolutinoal neural network with 6 layers of 3x3x3 convolutions. This post-processing network is trained by IoU on the same training partition as the point cloud generation network. In order to compensate for difference in point density among objects of different volumes, we trained another network to predict the object's volume. The predicted volume is concatenated with the registered occupancy as the 3D conv network's input. Using the point cloud generation network trained by either EMD or CD to is enough to outperform 3D-R2N2's result. The maximum performance as reported in the main paper is obtained by feeding both network's prediction into the post processing network. We also notice that the volume prediction network is not necessary to outperform 3D-R2N2. However, it consistently gives performance gain, so we kept this component in our experiments.

References

[1] C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[2] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2015.

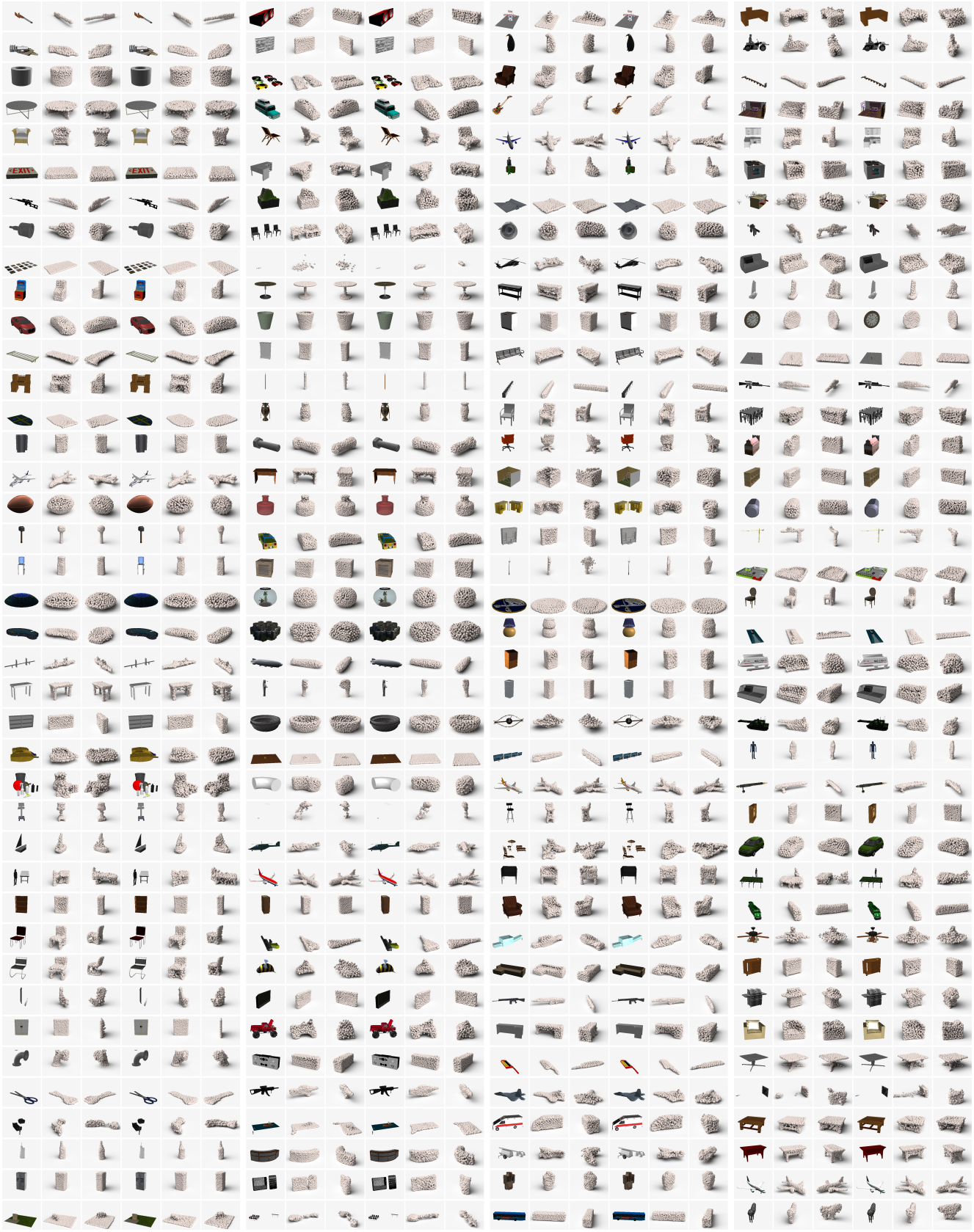

Figure 6. First 5 mini-batches of our validation set. Result obtained by CD is on the left, EMD on the right.