Appendix

A. Network architecture

Our network architecture has two parts, one for embedding the conditional input r to model uncertainty and the other for encoding and decoding input images. Figure 1 shows the encoder-decoder network architecture. For the embedding part, we set random input r as a 128-dimensional vector. At the training stage and at the initialization of test stage, each dimension is sampled from a Gaussian distribution $N(0, 1)$. Then we use two fully-connected layers (with 256 and 2304 output channels respectively), a reshape layer (from 2304 to 24×32×3) and two 3x3 convolutional layers (with 32 and 128 output channels respectively). For the encoder-decoder part, we use the two-branch version proposed in [4]. This encoder-decoder network consists of two prediction branches, one is used for capturing high level structures and the other learns geometric continuity. **Code will be made available.**

First, the input image is encoded into an intermediate latent variable z_i and the random input is embedded as z_r. Both z_i and z_r have the equal shapes of 24×32×128. Then, channel-wise concatenation is performed on the two encoded latent variables. Finally, the concatenated feature is decoded into the output point cloud.

B. More Implementation Details

B.1. Datasets

ShapeNet [2] contains 57386 CAD models across 55 different categories. We randomly took 80% of the objects for training and the rest for testing. For multi-view images rendering, we used the off-the-shelf renderer[1] provided by [7]. For the groundtruth point clouds, we used the data[2] provided by [1]. Each point cloud consists of 2048 points uniformly sampled from the mesh on the dataset. We used "chair" for single-category experiments and the 13 popular categories following 3D-R2N2 [3] for multi-category experiments.

Stanford Online Products [6] is an online repository initially released to accelerate the field of metric learning. It contains automatically downloaded data from https://www.ebay.com. We used "chair" and "sofa" in our experiments for multi-view reconstruction on real world images.

B.2. Baseline approaches

We reproduced several benchmark results of these methods on the datasets with their released code. In this section, we will show some details on these experiments.

3D-R2N2 [3] For the 32×32×32 voxelized groundtruth for 3D-R2N2 [3], we directly used the provided voxels from their repositories. Following the paper, we applied two-stage training for 20k and 40k iterations on the training data. For Chamfer Distance computation, we uniformly sampled point clouds on the predicted voxels using their off-the-shelf functions.

PTN [8] Similar to 3D-R2N2 [3], we uniformly sampled point clouds on the predicted voxels to enable comparsion with the groundtruth point clouds.

PSGN [4] For fair comparison, we used the two-branch version of the network architecture described in [4] with an output of 2048 points. We trained the fully-supervised deterministic model for 100k iterations with an Adam initial learning rate 1e-4.

Lin *et al.* [5] We followed the two-stage training strategy in their paper. For the depth map rendered from the fixed 8 poses, we used their off-the-shelf released data described at https://github.com/chenhsuanlin/3D-point-cloud-generation. As they only released data for single-category experiments, we did not reproduce their 13-category results. Note that our input images and the groundtruth shapes are different from theirs (our groundtruth consist of 2048 points for each shape, which differs from their 10k dense point clouds). This made us unable to directly compare our 13-category performance with that reported in their main paper.

Table 1: Results of our model on different number of input views. 'n' denotes the number of views. 'cat1' and 'cat13' denote single-category and multi-category experiments respectively. CD (FPS-CD) is reported.

n	cat13	cat1
1	5.76(5.76)	5.37(5.37)
2	4.80(4.93)	4.57(4.69)
3	4.32(4.59)	4.08(4.33)
4	4.04(4.44)	3.79(4.18)
5	3.92(4.42)	3.69(4.18)
6	3.77(4.37)	3.54(4.13)
7	3.67(4.36)	3.44(4.12)
8	3.58(4.34)	3.37(4.08)

B.3. Highly diverse generative model design

We present details on the highly diverse model used in the final paragraph of Section 4.4 in our main paper. To

[1] https://github.com/shubhtuls/mvcSnP/tree/master/preprocess/synthetic/rendering

[2] https://github.com/optas/latent_3d_points

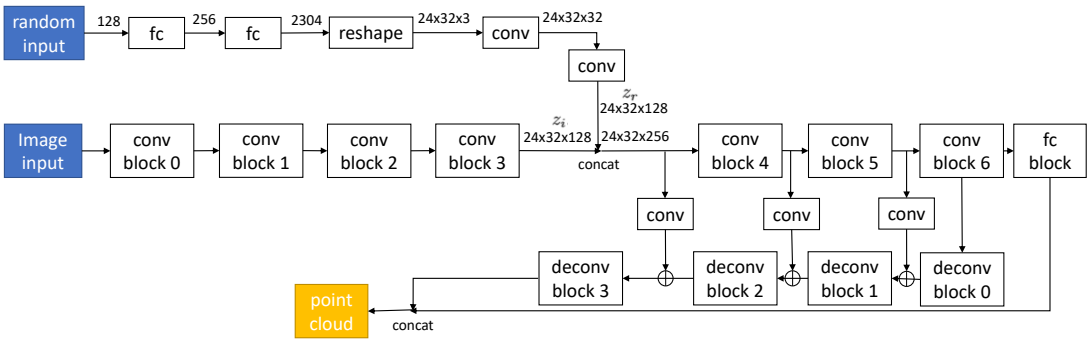

Figure 1: Network architecture of the conditional generative model.

better demonstrate the positive correlation between the consistency loss and the 3D reconstruction error, we trained a highly diverse conditional generative model on multi-view images. Specifically, we applied diversity constraint on the whole concatenated point clouds at the second stage of the training. We used $\alpha = 15.0$ and $\beta = 0.5$ in this experiment. Similar to the main experiment, we trained the model for 40,000 iterations using Adam with an initial learning rate 1e-4.

From the Table 6 in the main paper we can infer the positive correlation between the consistency loss and CD at inference stage. Moreover, it is shown that applying the diversity constraint to the single-view predicted point clouds rather than the concatenated results gives much higher performance (3.37 vs. 4.66 for CD).

C. More Ablation Studies

C.1. Ablation on number of input images

We conducted experiments with different number of input views. We randomly sample n views and run inference on both single and multiple categories. Results are shown in Table 1. When we input only one view , the consistency loss is unable to work, so the performance of the conditional model is relatively poor. With more views observed, the performance becomes consistently better.

C.2. Runtime analysis

We did not use any type of connectivity on the view-based sampling layer. On 2048 (10k) points, our layer, which takes 6.6ms (10.2ms) on average, is an $O(n)$ approximation with 10.4% (3.6%) hidden parts included. The accurate mesh-based sampling is at least $O(nlgn)$ with a large constant. Generating a triangle mesh from 2048 (10k) points already takes 209.8ms (1.12s), which becomes the speed bottleneck at both training and inference.

Note that in some cases the current system suffers from the problem of empty faces. It could be due to that the current view-based sampling is an approximation form which might wrongly samples the points from the back part. This will get the wrongly sampled points to be closer to the front, resulting in unbalanced density. Developing efficient usage of connectivity to better approximate the view-based sampling process might help resolve this issue.

D. More Qualitative Results

In this section, we show more samples of qualitative results on single-view conditional predictions and multi-view reconstruction.

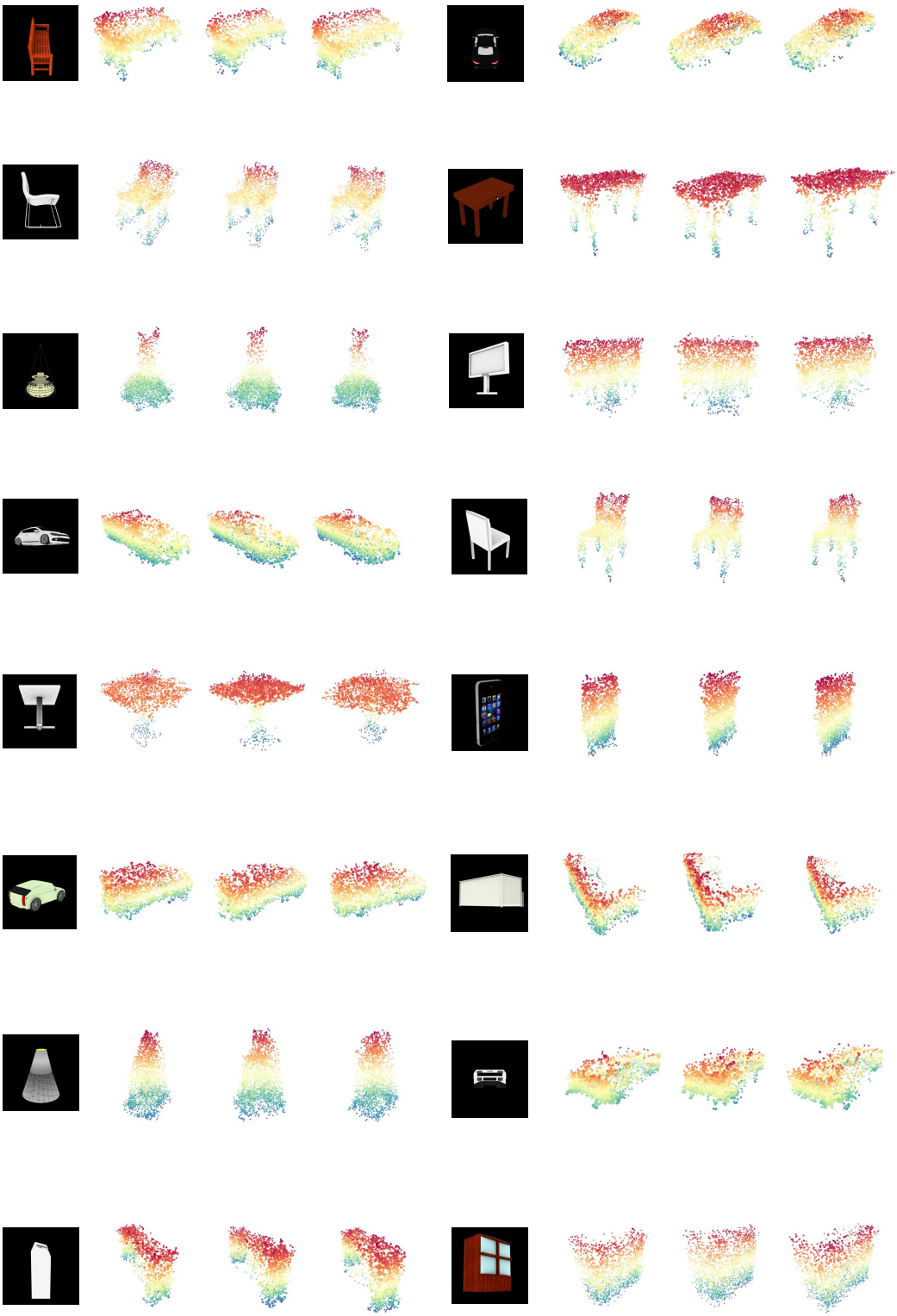

Figure 2: Visualization on multiple predictions on single-view images.

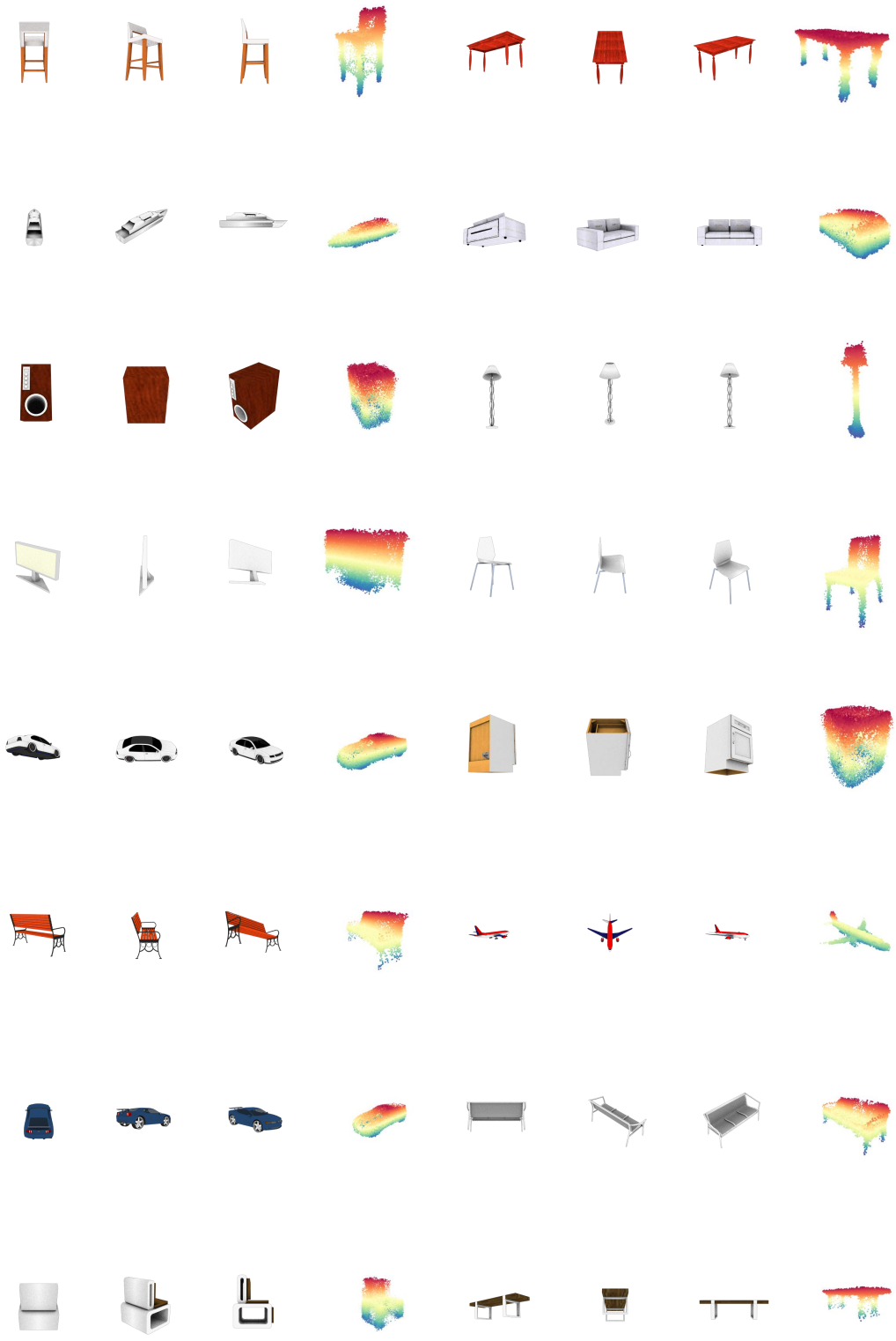

Figure 3: Visualization on multi-view reconstruction with our proposed method.

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017. 1

[2] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 1

[3] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 1

[4] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, pages 605–613, 2017. 1

[5] C.-H. Lin, C. Kong, and S. Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI*, 2018. 1

[6] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012, 2016. 1

[7] S. Tulsiani, A. A. Efros, and J. Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *CVPR*, 2018. 1

[8] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *NIPS*, pages 1696–1704. 2016. 1