Supplemental Material: Shape Completion using 3D-Encoder-Predictor CNNs and Shape Synthesis

Angela Dai[1] Charles Ruizhongtai Qi[1] Matthias Nießner[1,2]
[1]Stanford University [2]Technical University of Munich

In this supplemental material, we provide additional evaluation and results of our shape completion method "Shape Completion using 3D-Encoder-Predictor CNNs and Shape Synthesis" [3].

1. Additional Results on Synthetic Scans

Tab. 1 shows a quantitative evaluation of our network on a test set of input partial scans with varying trajectory sizes ($\geq$ 1 camera views). Our 3D-EPN with skip connections and class vector performs best, informing the best shape synthesis results.

Method	ℓ_1-Err (32^3)	ℓ_1-Err (128^3)
Ours (3D-EPN + synth)	0.382	1.94
Ours (3D-EPN-class + synth)	0.376	1.93
Ours (3D-EPN-unet + synth)	0.310	1.82
Ours (final) 3D-EPN-unet-class + synth	**0.309**	**1.80**

Table 1: Quantitative shape completion results on synthetic ground truth data for input partial scans with varying trajectory sizes. We measure the ℓ_1 error of the unknown regions against the ground truth distance field (in voxel space, up to truncation distance of 2.5 voxels).

2. Results on Real-world Range Scans

In Fig. 4, we show example shape completions on real-world range scans. The test scans are part of the RGB-D test set of the work of Qi et al. [5], and have been captured with a PrimeSense sensor. The dataset includes reconstructions and frame alignment obtained through VoxelHashing [4] as well as mesh objects which have been manually segmented from the surrounding environment. For the purpose of testing our mesh completion method, we only use the first depth frame as input (left column of Fig. 4). We use our 3D-EPN trained as described on purely synthetic data from ShapeNet [1]. As we can see, our method is able to produce faithful completion results even for highly partial input data. Although the results are compelling for both the intermediate 3D-EPN predictions, as well our final output, the completion quality looks visually slightly worse than the test results on synthetic data. We attribute this to the fact that the real-world sensor characteristics of the PrimeSense are different from the synthetically-generated training data used to train our model. We believe a better noise model, reflecting the PrimeSense range data, could alleviate this problem (at the moment we don't simulate sensor noise). Another option would be to generate training data from real-world input, captured with careful scanning and complete scanning patterns; e.g., using the dataset captured by Choi et al. [2]. However, we did not further explore this direction in the context of the paper, as our goal was to learn the completions from actual ground truth input. In addition to 3D-EPN predictions and our final results, we show the intermediate shape retrieval results. These models are similar; however, they differ significantly from the partial input with respect to global geometric structure. Our final results thus combine the advantages of both the global structure inferred by our 3D-EPN, as well as the local detail obtained through the shape synthesis optimization process.

3. Evaluation on Volumetric Representation

In Table 2, we evaluate the effect of different volumetric surface representations. There are two major characteristics of the representation which affect the 3D-EPN performance. First, a smooth function provides better performance (and super-resolution encoding) than a discrete representation; this is realized with signed and unsigned distance fields. Second, explicitly storing known-free space encodes information in addition to the voxels on the surface; this is realized with a ternary grid and the sign channel in the signed distance field. The signed distance field representation combines both advantages.

4. Single Class vs Multi-Class Training

Table 3 evaluates different training options for performance over multiple object categories. We aim to answer the question whether we benefit from training a separate

Surface Rep.	ℓ_1-Error (32^3)	ℓ_2-Error (32^3)
Binary Grid	0.653	1.160
Ternary Grid	0.567	0.871
Distance Field	0.417	0.483
Signed Distance Field	**0.379**	**0.380**

Table 2: Quantitative evaluation of the surface representation used by our 3D-EPN. In our final results, we use a signed distance field input; it encodes the ternary state of known-free space, surface voxels, and unknown space, and is a smooth function. It provides the lowest error compared to alternative volumetric representations.

network for each class separately (first column). Table 3 compares the results of training separate networks for each class with a single network trained over all classes (with and without class information). Our networks trained over all classes combined performs better than training over each individual class, as there is significantly more training data, and the network leveraging class predictions performs the best.

Category (# train models)	Separate EPN-unets (known class) ℓ_1-Error	EPN-unet w/o Class ℓ_1-Error	EPN-unet /w Class **Ours Final** ℓ_1-Error
Chairs (5K)	0.477	**0.409**	0.418
Tables (5K)	0.423	**0.368**	0.377
Sofas (2.6K)	0.478	0.421	**0.392**
Lamps (1.8K)	0.450	0.398	**0.388**
Planes (3.3K)	0.440	**0.418**	0.421
Cars (5K)	0.271	0.266	**0.259**
Dressers (1.3K)	0.453	0.387	**0.381**
Boats (1.6K)	0.380	0.364	**0.356**
Total (25.7K)	0.422	0.379	**0.374**

Table 3: Quantitative evaluations of 32^3 3D-EPNs; from left to right: separate networks have been trained for each class independently (at test time, the ground truth class is used to select the class network); a single network is used for all classes, but no class vector is used; our final result uses a single network trained across all classes and we input a probability class vector into the latent space of the 3D-EPN.

5. Evaluation on Different Degrees of Incompleteness

Fig. 1 shows an evaluation and comparisons against 3D ShapeNets [1] on different test datasets with varying degrees of partialness. Even for highly partial input, our method achieves relatively low completion errors. Compared to previous work, the error rate of our method is relatively stable with respect to the degree of missing data.

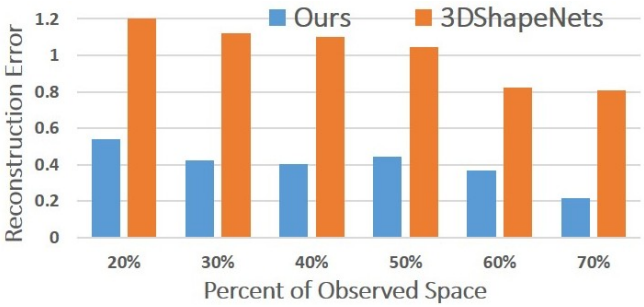

Figure 1: Quantitative evaluation of shape completion using our 3D-EPN and 3D ShapeNets [7] on different degrees of partial input. For this task, we generate several test sets with partial observed surfaces ranging from 20% to 70%. Even for very partial input, we obtain relatively low reconstruction errors, whereas 3D ShapeNets becomes more unstable.

6. Comparison against Sung et al. [6]

In Tab. 4 and Fig. 2, we compare against the method by Sung et al. [6] using the dataset published along with their method. Note that their approach operates on a point cloud representation for both in and output. In order to provide a fair comparison, we apply a distance transform of the predicted points and measure the ℓ_1 error on a 32^3 voxel grid.

Class (#models)	ℓ_1-Error (32^3)	
	Sung et. al [6]	Ours
assembly_airplanes (58)	0.56	**0.50**
assembly_chairs (64)	0.73	**0.51**
coseg_chairs (287)	0.72	**0.57**
shapenet_tables (37)	0.82	**0.45**
Total (446)	0.71	**0.54**

Table 4: Quantitative shape completion results on the dataset of Sung et. al [6]. We measure the ℓ_1 error of the unknown regions against the ground truth distance field (in voxel space, up to truncation distance of 3 voxels).

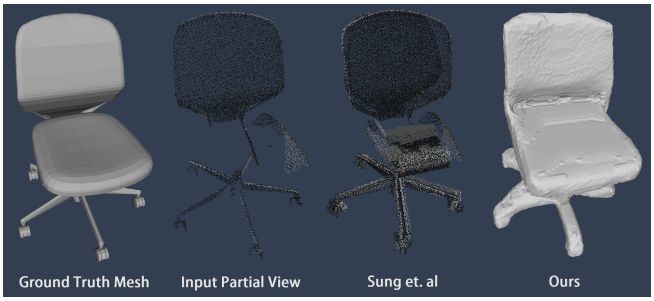

Figure 2: Qualitative comparison against Sung et. al [6]. Note that the missing chair seat and front of chair back introduce difficulties for inferring structure, whereas our method is able to more faithfully infer the global structure.

7. Shape Embeddings

Fig. 3 shows a t-SNE visualization of the latent vectors in our 3D-EPN trained for shape completion. For a set of test input partial scans, we extract their latent vectors (the 512-dimensional vector after the first fully-connected layer and before up-convolution) and then use t-SNE to reduce their dimension to 2 as (x, y) coordinates. Images of the partial scans are displayed according to these coordinates. Shapes with similar geometry tend to lie near each other, although they have varying degrees of occlusion.

References

[1] 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, 2

[2] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun. A large dataset of object scans. *arXiv preprint arXiv:1602.02481*, 2016. 1

[3] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages –, 2017. 1

[4] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013. 1

[5] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016. 1

[6] M. Sung, V. G. Kim, R. Angst, and L. Guibas. Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, 34(6):175, 2015. 2

[7] 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. 2

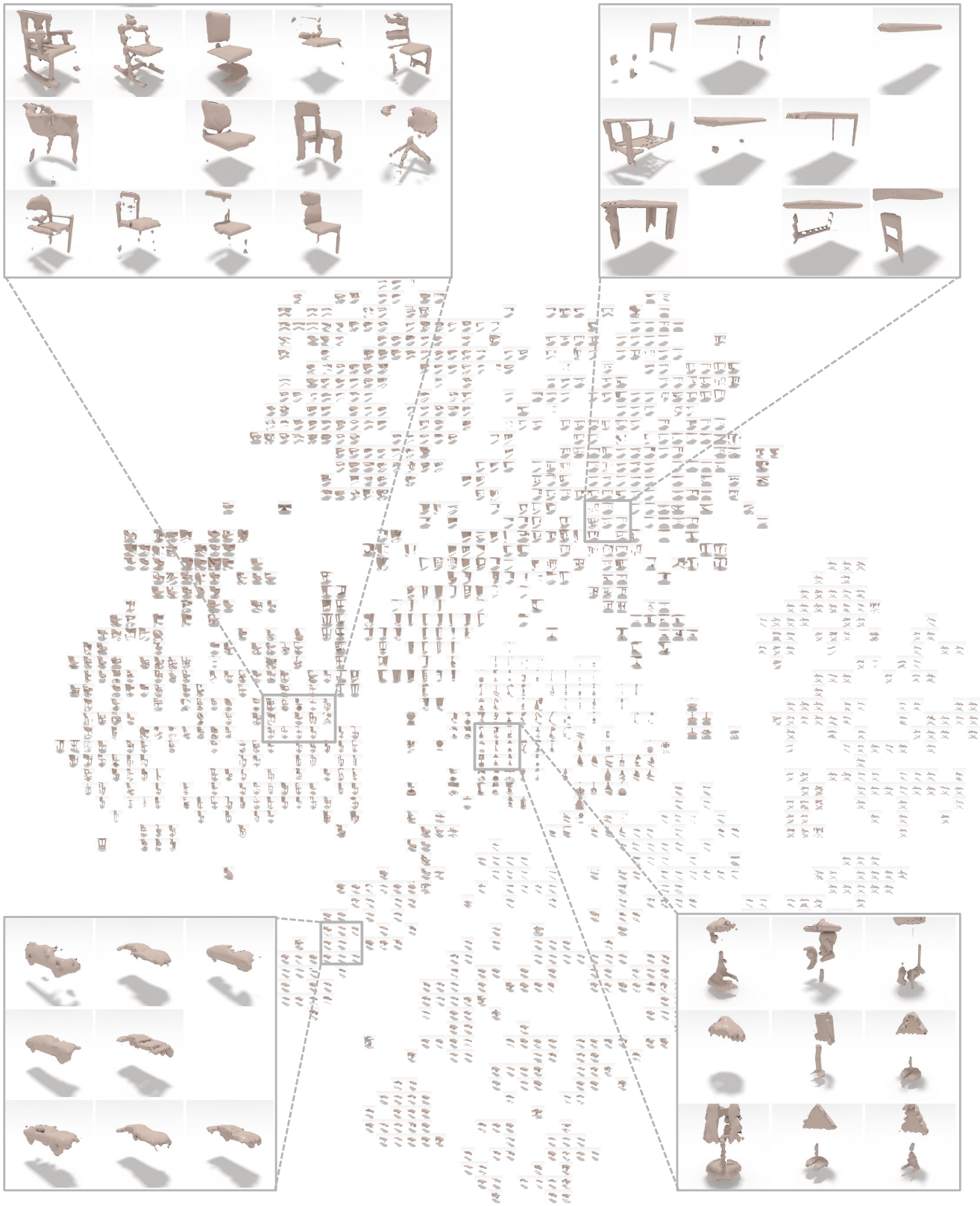

Figure 3: t-SNE visualization of the latent vectors in our 3D-EPN trained for shape completion. The rendered images show input partial scans. Four zoom-ins are shown for regions of chairs (top left), tables (top right), cars (bottom left) and lamps (bottom right).

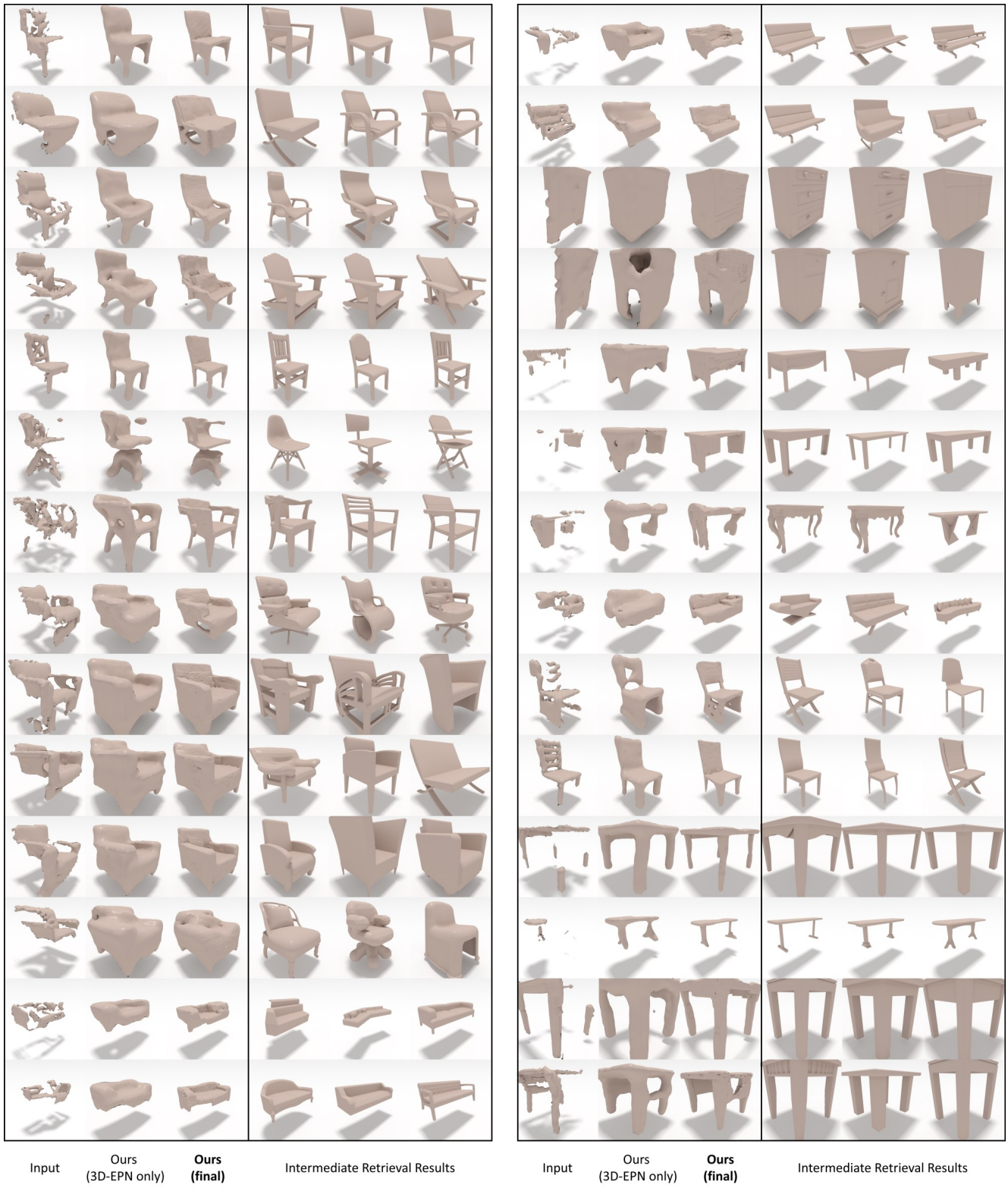

Figure 4: Example shape completions from our method on real-world range scans from commodity sensors (here, a PrimeSense is used). We visualize partial input, 3D-EPN predictions, and our final results. In addition, we show the retrieved shapes as intermediate results on the right. Note that although the retrieved models look clean, they are inherently different from the input with respect to global structure.