# Supplementary Material for PC$^2$: Projection-Conditioned Point Cloud Diffusion for Single-Image 3D Reconstruction

Luke Melas-Kyriazi Christian Rupprecht Andrea Vedaldi

Visual Geometry Group, Department of Engineering Science, University of Oxford
{lukemk,chrisr,vedaldi}@robots.ox.ac.uk

## A. Implementation Details

Here, we provide additional implementation details.

First, we discuss the Point-Voxel [1] model which is used to process the partially-denoised point cloud at each step of the diffusion process. The Point-Voxel [1] model processes a point cloud using two branches simultaneously: a point-based branch and voxel-based branch. The point-based branch is a simple multi-layer perceptron which is applied to each point independently, as in PointNet [2, 3] (without the global pooling in the final layer of PointNet). The voxel-based branch first discretizes the points into a coarse voxel grid of size $128^3$, which is fed into a 3D U-Net. As in [1], the 3D U-Net consists of four downsampling ("Set Abstraction") layers followed by four upsampling ("Feature Propogation") layers. Due to this fine-to-coarse-to-fine structure, the network is able to capture both global and local shape information. Additionally, to make the model aware of the current timestep of the diffusion process, we concatenate an embedding of the current timestep to the point features at the input to each layer.

Second, we discuss the implementation of the projection feature. We perform the projection by rasterizing the point cloud from the given camera view. We utilize the `PointRasterizer` class of PyTorch3D using a point radius of $0.0075$ and 1 point per pixel. For each point in the point cloud, if the point is rasterized onto a pixel in the input image, we concatenate the image features corresponding to the pixel onto that point's existing feature vector (which is simply a sinusoidal positional embedding of its current position) for input to the model. Additionally, we concatenate the value of the (binary) object mask at the given pixel and a two-dimensional vector pointing from the pixel to the closest pixel in the mask (i.e. a two-dimensional distance function corresponding to the mask region; this is the zero vector inside the mask and a nonzero vector outside the mask). If a pixel is not rasterized to a point (for example, because it is occluded by another point), we concatenate a vector of zeros in place of all the quantities above.

## B. Evaluation Metric.

As described in the main paper, we use the F-score metric proposed by Tatarchenko [4]. For two point clouds $X$ and $\hat{X}$, it is defined as

$$\text{F-Score}_d(X, \hat{X}) = \frac{2P_d(X, \hat{X})R_d(X, \hat{X})}{P_d(X, \hat{X}) + R_d(X, \hat{X})} \tag{1}$$

where $P_d$ and $R_d$ denote precision and recall, respectively, and $d$ is a fixed threshold distance $d$. Precision and recall are defined as

$$P(d) = \frac{1}{n_{\hat{X}}} \sum_{\hat{p} \in \hat{X}} \left[ \min_{p \in X} ||p - \hat{p}|| < d \right] \tag{2}$$

$$R(d) = \frac{1}{n_X} \sum_{p \in X} \left[ \min_{\hat{p} \in \hat{X}} ||p - \hat{p}|| < d \right] \tag{3}$$

and we use $d = 0.01$ following prior work.

## C. Additional Qualitative Examples

We provide additional qualitative examples of our method in Figs. 1, 2 and 4 to 6. Figures 1 to 3 show examples of reconstructions on additional categories of Co3D, including hydrants, teddybears, glasses, remotes, motorcycles, hairdryers, plants, and donuts. Fig. 4 contains a selection of the best reconstructions produced by our model for each category of ShapeNet, as ranked by F-score. Fig. 6 contains random examples of reconstructions produced by our model on ShapeNet. Finally, Fig. 5 shows a selection of the worst examples produced by our model for each category on ShapeNet, as ranked by F-score.

## D. Additional Ablations

We include additional ablations omitted from the main paper due to space constraints. These ablations were performed on a subset of ShapeNet dataset consisting of only the `sofa` category.

**Mask Distance Function.** We removed the 2D mask distance function described in Appendix A. This change had a small effect, reducing the $F$-score by $0.019$ points, a relative decrease of $9\%$. Qualitatively, the generated point clouds were similar to those produced using the mask distance function.

**Projection Method.** We replaced the rasterization-based projection described in Section 3.4 with a naive projection that projects all points (including occluded points) onto the image. This change was detrimental, reducing the $F$-score by $0.081$ points, a relative decrease of $40\%$. Qualitatively, these point clouds were significantly worse than those with the rasterization-based projection. These results suggest that the rasterization-based projection is a key component of the method.

## E. Analysis of Failure Cases

Failure cases of our model are shown in Fig. 5. Note that these are from the ShapeNet-R2N2 dataset, which combines 13 ShapeNet classes but *does not* permit the use of category labels. In other words, the model is image-conditional, but not class-conditional.

Examining these failure cases, we observe that our model sometimes performs poorly on images with ambiguous categories. For example, in the $8^{th}$ column of the $2^{nd}$ row of the figure, it appears that the model generates a chair rather than a box. Similarly, in the $12^{th}$ row of the $5^{th}$ row of the figure, the object seems to have generated a box rather than a couch. These errors are most likely due to the fact that these categories all have instances which resemble rectangular prisms from certain views.

It is also notable that on many of the challenging examples on which our model struggles (*e.g.*, the examples for the `watercraft` category located in the last row of the figure), other models also struggle to a similar degree.

Figure 1. **Additional qualitative examples.** Examples from the hydrants category of Co3D. The first column in each row shows the input image. The second and third columns show intermediate steps in the diffusion process. The fourth column shows the final reconstructed point cloud with color. The remaining five rows show the final predicted point cloud from novel views.

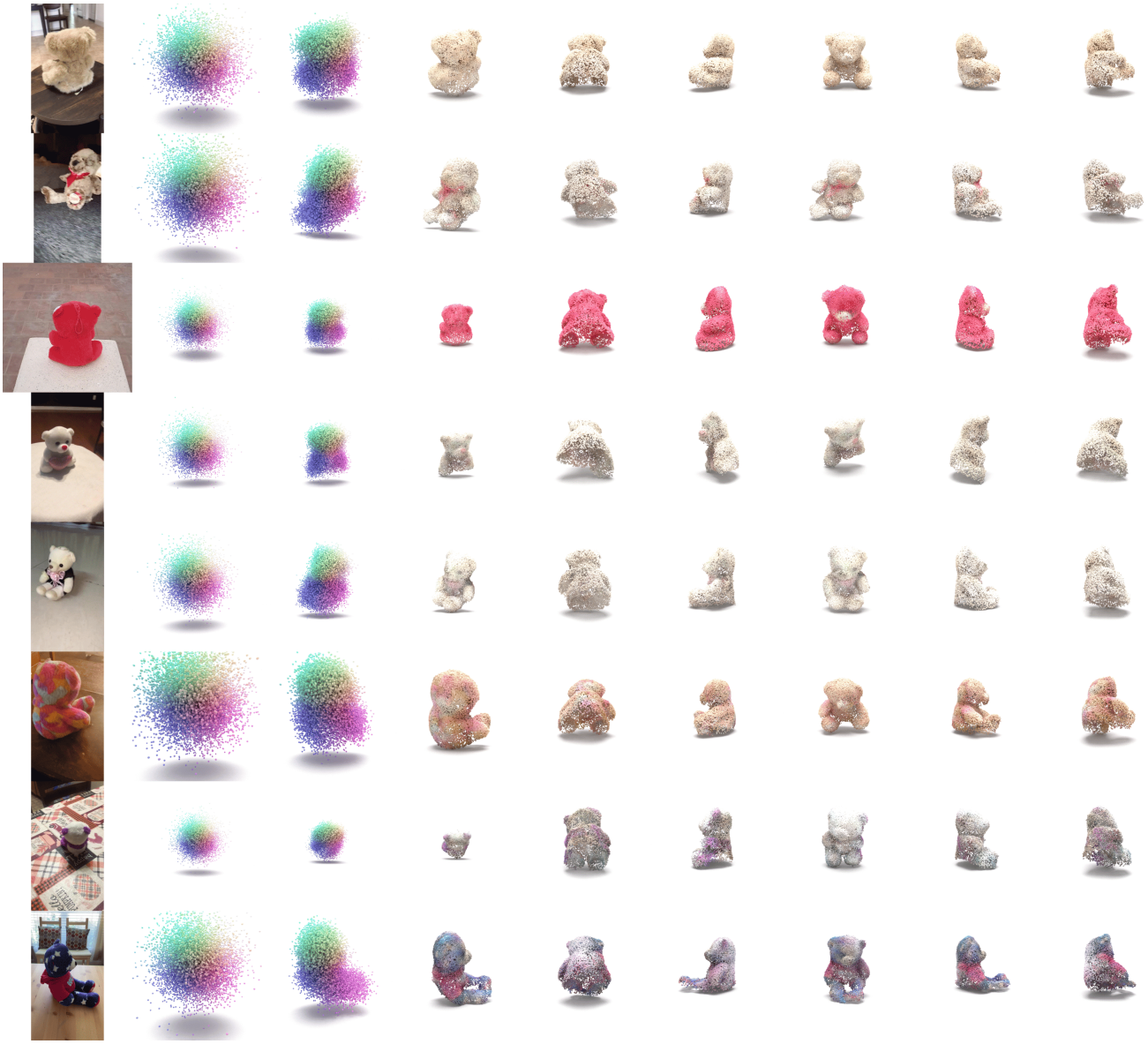

Figure 2. **Additional qualitative examples.** Examples from the teddy bear category of Co3D. The first column in each row shows the input image. The second and third columns show intermediate steps in the diffusion process. The fourth column shows the final reconstructed point cloud with color. The remaining five rows show the final predicted point cloud from novel views.

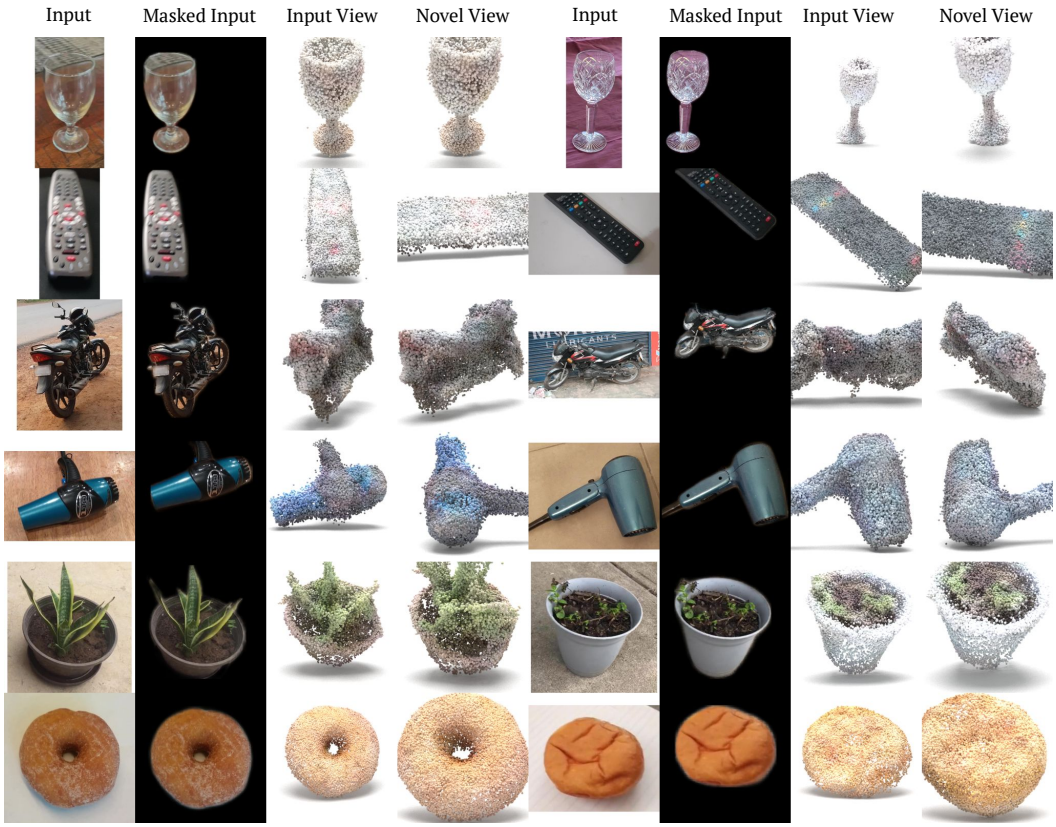


Figure 3. **Additional qualitative examples.** Examples from six additional categories: glasses, remotes, motorcycles, hairdryers, plants, and donuts.


Figure 4. **Successful examples** produced by our method along with prior work. The leftmost image in each set of images is the input image. Note that there are no images in the last row of the right half of the figure because we show examples for all 13 ShapeNet-R2N2 classes (seven on the left and six on the right).


Figure 5. **Failure cases** of our method along with prior work. The leftmost image in each set of images is the input image. Note that there are no images in the last row of the right half of the figure because we show examples for all 13 ShapeNet-R2N2 classes (seven on the left and six on the right).

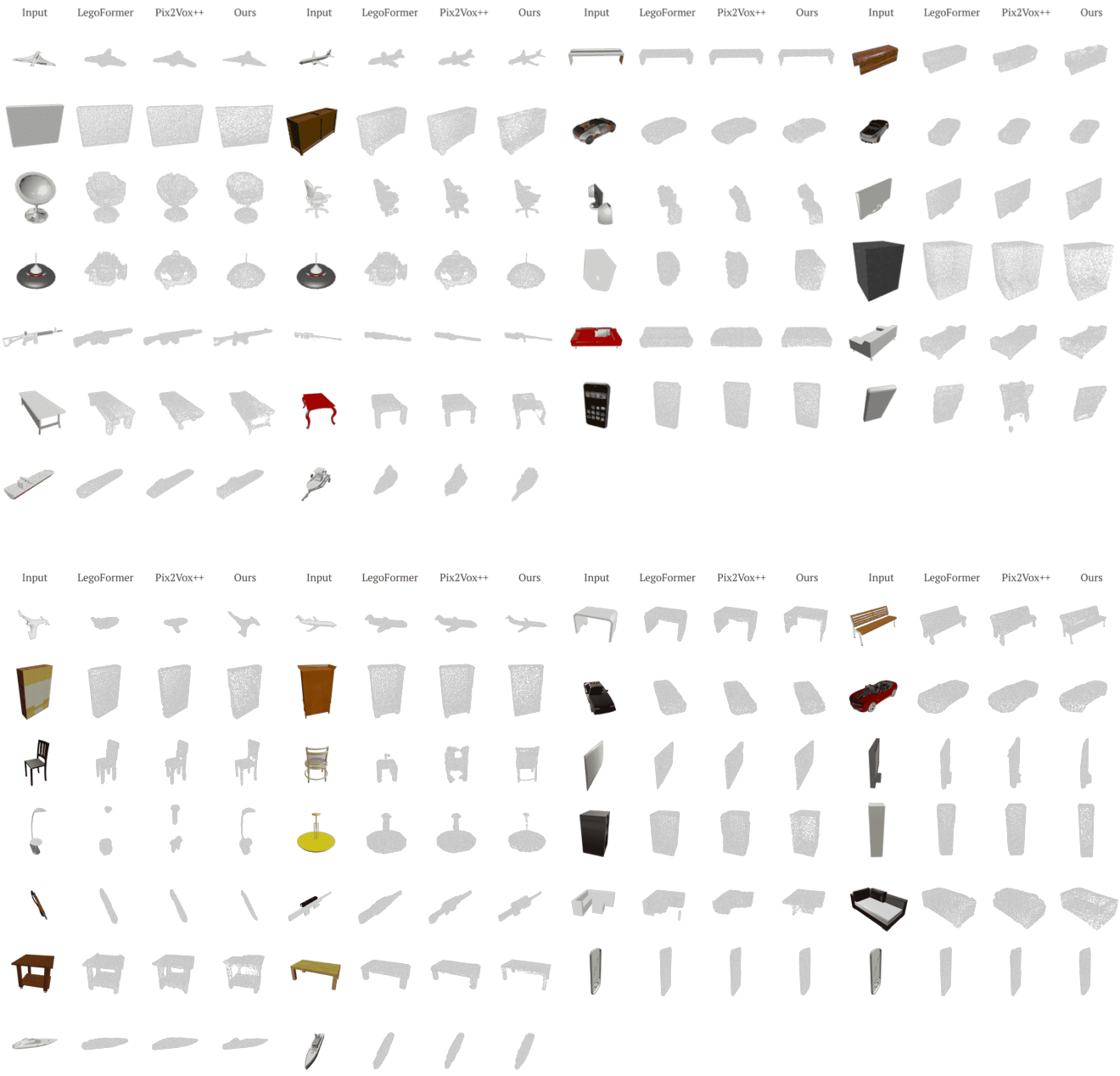


Figure 6. **Random examples** of our method along with prior work. The leftmost image in each set of images is the input image. Note that there are no images in the last row of the right half of the figure because we show examples for all 13 ShapeNet-R2N2 classes (seven on the left and six on the right).

## References


[1] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 1

[2] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, pages 652–660, 2017. 1

[3] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Nips*, pages 5099–5108, 2017. 1

[4] Maxim Tatarchenko, Stephan R. Richter, Rene Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proc. CVPR*, June 2019. 1