

PC²: 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

<https://lukemelas.github.io/projection-conditioned-point-cloud-diffusion>

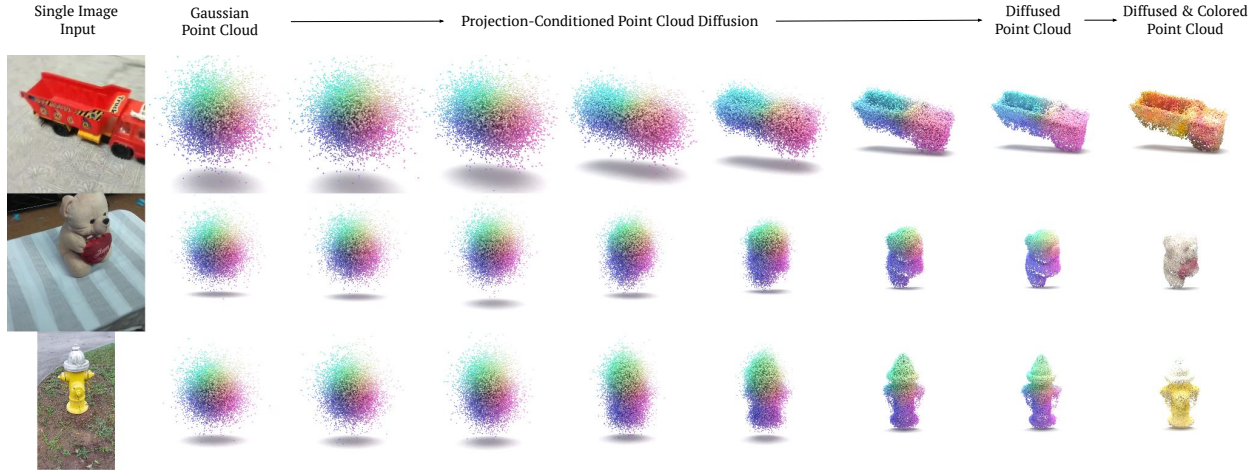


Figure 1. **Projection-Conditioned Point Cloud Diffusion:** Our model performs single-image 3D point cloud reconstruction by gradually diffusing an initially random point cloud to align the with input image. Our model has been trained through simple, sparse COLMAP supervision from videos.

Abstract

Reconstructing the 3D shape of an object from a single RGB image is a long-standing and highly challenging problem in computer vision. In this paper, we propose a novel method for single-image 3D reconstruction which generates a sparse point cloud via a conditional denoising diffusion process. Our method takes as input a single RGB image along with its camera pose and gradually denoises a set of 3D points, whose positions are initially sampled randomly from a three-dimensional Gaussian distribution, into the shape of an object. The key to our method is a geometrically-consistent conditioning process which we call *projection conditioning*: at each step in the diffusion process, we project local image features onto the partially-denoised point cloud from the given camera pose. This projection conditioning process enables us to generate high-resolution sparse ge-

ometries that are well-aligned with the input image, and can additionally be used to predict point colors after shape reconstruction. Moreover, due to the probabilistic nature of the diffusion process, our method is naturally capable of generating multiple different shapes consistent with a single input image. In contrast to prior work, our approach not only performs well on synthetic benchmarks, but also gives large qualitative improvements on complex real-world data.

1. Introduction

Reconstructing the 3D structure of an object from a single 2D view is a long-standing computer vision problem. Given more than one view, it is possible to reconstruct an object’s shape using the tools of multiple-view geometry, but

in the case of a single view the problem is highly ill-posed and requires prior understanding of the possible shapes and appearances of the objects.

Despite the difficulty of this task, humans are adept at using a range of monocular cues and prior knowledge to infer the 3D structure of common objects from single views. For example, when a person looks at a photograph of an object, they can easily imagine what the shape of the backside of the object may plausibly be. The ease with which humans perform this task, and its relative difficulty for computer vision systems, has attracted the attention of the research community for nearly half a century [4, 5, 16, 24]. Moreover, given the prevalence of single-view RGB image data in the real world, the single-view reconstruction problem has practical applications in numerous areas, such as augmented and virtual reality.

Over the past five years, a substantial body of research has emerged around reconstruction via end-to-end deep learning methods [11, 14, 18, 20, 26, 36, 46, 47, 49, 50, 53–57, 59]. These works are capable of predicting volumes from single images, yet many remain limited to low-resolution geometries (e.g. dense voxel grids), and thus have limited ability to reconstruct the precise geometry of the input image. Some recent works [9, 12, 37, 58] also utilize implicit representations and radiance fields, which are capable of rendering novel views with photographic quality but often suffer from other drawbacks, such as an inability to reconstruct a distribution of possible 3D shapes from a single input image.

In this work, we take inspiration from recent progress in the generative modeling of 2D images using denoising diffusion probabilistic models. In the domain of 2D images, we have seen that diffusion models (e.g. Latent-Diffusion [38], SDEdit [25], GLIDE [28], and DALL-E 2 [32]) can produce remarkably high-fidelity image samples either from scratch or when conditioned on textual inputs. They do so by learning complex priors over the appearances of common objects. For example, GLIDE [28] is designed to conditionally inpaint unseen regions in an image. We seek to bring these advances to the domain of 3D reconstruction to conditionally generate the shape of unseen regions of a 3D object.

In order to do so, we propose to use diffusion models for single-image 3D reconstruction. Our approach represents shapes as unstructured point clouds: we gradually denoise a randomly-sampled set of points into a target shape conditional on an input image and its corresponding viewpoint. The key to our approach is a novel way of conditioning the diffusion process to produce 3D shapes which are geometrically consistent with the input images. Specifically, at each iteration of the diffusion process, we project image features directly onto the points of the partially-denoised point cloud. When shape reconstruction is complete, we are also able to use the same projection conditioning to predict point colors.

Our method differs from prior work in three major as-

pects: (1) our point cloud-based shape representation, (2) our projection-conditioned diffusion model, and (3) our use of projection conditioning to predict point colors in addition to point shapes. Although these may seem separate from one another, they are in fact directly linked: the order-agnostic and unstructured nature of point clouds naturally lends itself to the highly flexible nature of diffusion models. Without the point cloud-based representation, our projection-conditioned diffusion approach would be limited to coarse voxel-based shape approximation.

Finally, due to its probabilistic nature, given a single input image, our model is able to generate multiple plausible 3D point clouds which are all consistent with the input. We leverage this property by introducing a novel filtering step in the sampling process, which helps to address the ill-posed nature of the single-view 3D reconstruction problem. Specifically, we generate multiple point clouds for a given input image, and filter these point clouds according to how well they match the input mask. Filtering enables us to benefit from the diversity of our model’s reconstructions in an entirely automated fashion.

Experimentally, we not only perform competitively on the synthetic ShapeNet benchmark, but we also move beyond synthetic data: we demonstrate high-quality qualitative single-view reconstruction results on multiple categories in the challenging, real-world Co3D [34] dataset.

We present this work as a first step toward using denoising diffusion models for large-scale 3D reconstruction. Given the success of scaling 2D diffusion models over the past two years, our work demonstrates that 3D diffusion models could represent a path toward large-scale 3D reconstruction of complex objects from single-view images. To encourage further research into diffusion models for 3D reconstruction, we will release code and pretrained models on acceptance.

2. Related work

Single-View 3D reconstruction Originally, research on 3D reconstruction focused primarily on reconstruction from multiple views using classical geometric techniques [6]. Most commonly, these methods map the input image into a feature representation using a 2D convolutional network, and then decode the features into a 3D representation using a 3D convolutional network or a sequence model. This decoding process generally requires the output 3D representation to have a fixed and regular structure, such as a voxel grid.

One of the pioneering methods in this line of work was 3D-R2N2 [1], which encodes an input 2D image into a low dimensional embedding using a standard 2D convolutional network, process this embedding with a 3D-LSTM, and then decode an output voxel grid with a 3D convolutional network. [1] also introduced a standard set of views and renders for evaluating single-view reconstruction methods based on the ShapeNet dataset, which we use for evaluation

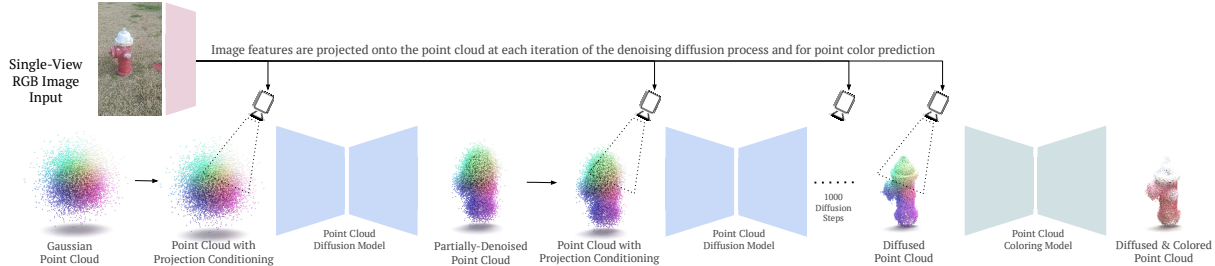


Figure 2. **PC²** reconstructs a colored point cloud from a single input image along with its camera pose. The method contains two sub-parts, both of which utilize our model projection conditioning method. First, we gradually denoise a set of points into the shape of an object. At each step in the diffusion process, we project image features onto the partially-denoised point cloud from the given camera pose, augmenting each point with a set of neural features. This step makes the diffusion process conditional on the image in a geometrically-consistent manner, enabling high-quality shape reconstruction. Second, we predict the color of each point using a model based on the same projection procedure.

in this work (Sec. 4). LSM [13] processes images into feature maps with a 2D network, unprojects these features into a 3D voxel grid, and processes the grid with a 3D convolutional network. Pix2Vox [54] and its successor Pix2Vox++ [55] use a simple encoder-decoder architecture consisting of a 2D convolutional encoder and a 3D convolutional decoder augmented with a multi-scale fusion module. Most recently, LegoFormer [56] adopts a transformer-based approach. They encode an image into a feature vector and decode a 32^3 dense voxel grid as a sequence of $512 \times 4 \times 4$ blocks. For this decoding they employ a non-autoregressive transformer-based decoder with learned queries.

Additionally, over the past two years a new body of research has emerged around differentiable rendering. Popularized by NeRF [27], these methods learn implicit representations of either radiance [27] or signed distance [51]. Although the vast majority of this work operates on the level of single scenes with abundant multi-view data, a few recent works have tackled the single-view setting [9, 12, 17, 37, 57]. Most relevant for this paper are Nerf-WCE [9] and PixelNeRF [57], which concurrently developed methods for single/few-view reconstruction, conditioning a NeRF on image features from reference views. Both work well in the few-view settings on datasets such as Co3D [35], but strong results in the single-view setting remain elusive. We compare qualitatively against NeRF-WCE in Sec. 4.

In this paper, we take an entirely different approach from those above. We utilize the flexible machinery of denoising diffusion probabilistic models for point cloud-based reconstruction. Due to their probabilistic nature, diffusion models enable us to capture the ambiguity of unseen regions while also generating high-resolution point cloud-based shapes. Below, we give an overview of the related literature on diffusion models.

Diffusion Models Denoising diffusion probabilistic models are a class of generative models based on iteratively reversing a Markovian noising process. Early work on

Category	Global Cond.	PC ²
airplane	0.197	0.473
table	0.156	0.290

Table 1. **Motivating Experiment.** Here, we show a small quantitative comparison on two categories from ShapeNet [1] which motivates our projection conditioning method. The table shows F-scores (higher is better). For the global conditioning approach, the input image is first processed into a global feature vector using a convolutional network [41], and this vector is used to condition a denoising diffusion model. Relative to the global conditioning, our projection conditioning is much more effectively able to reconstruct object geometries.

diffusion models for image generation, inspired by thermodynamics, framed the problem as one of optimizing a variational lower bound of a latent variable model [10]. Later work showed that this modeling paradigm is equivalent to score-based generative modeling [43, 44], and can also be seen as a discretization of a continuous process based on stochastic differential equations [45]. Recent work has explored faster/deterministic sampling for diffusion models [19, 42, 52], class-conditional models [2, 44], text-conditional models [28], and modeling in latent space [38].

Diffusion Models for Point Clouds Over this past year, three applications of diffusion models for unconditional point cloud generation have emerged. [22] and [60] proposed similar generation setups, differing in their use of a PointNet [30] and a Point-Voxel-CNN [60], respectively. Most recently [23] uses a diffusion model followed by a refinement model for the purpose of completing partial point clouds. However, these three works only tackle the problem of unconditional shape generation or completion and do not address the question of how to reconstruct real-world images. Additionally, they train only on synthetic datasets, whereas we show results for complex real-world data.

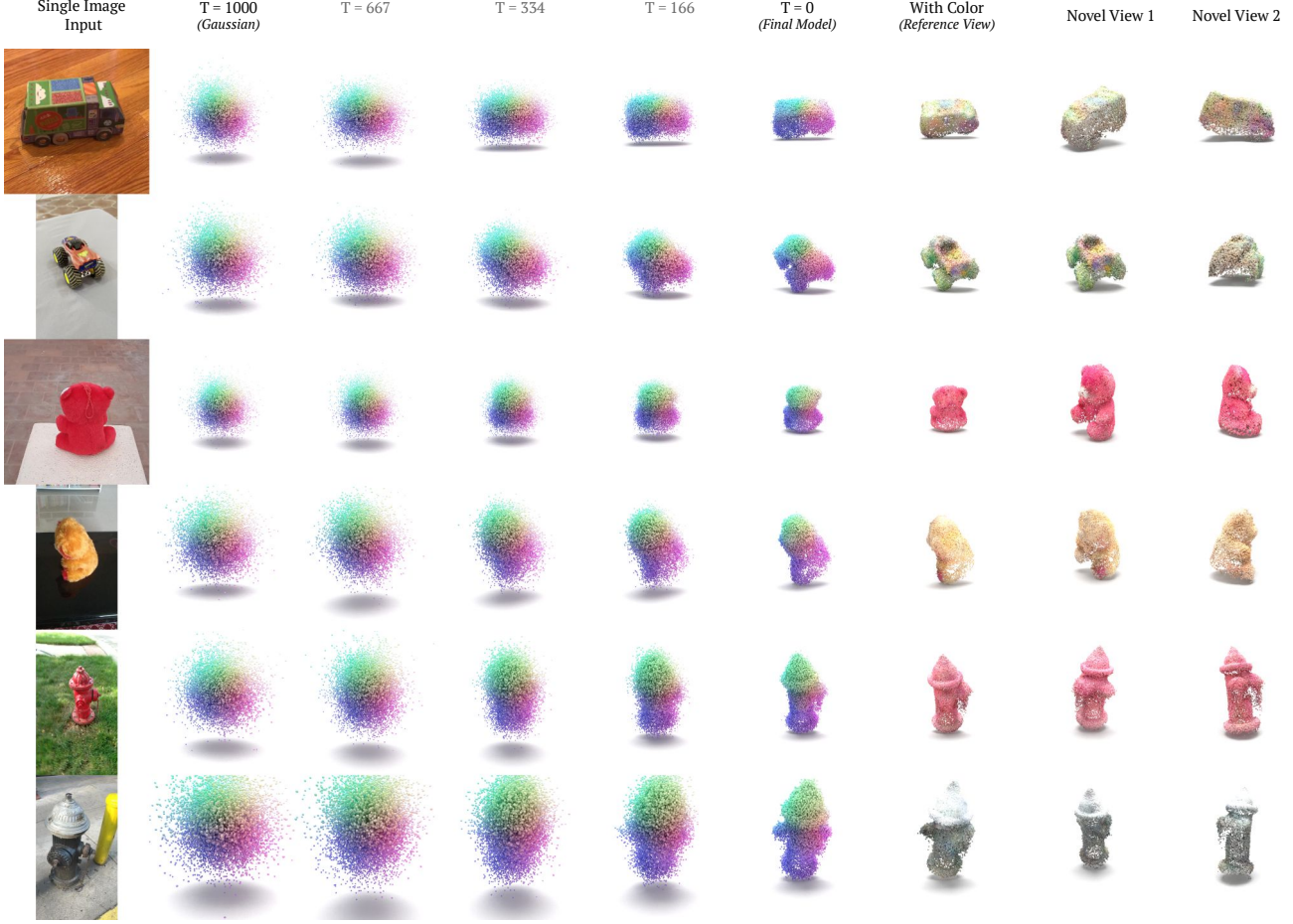


Figure 3. Examples of PC^2 on real-world images from Co3D [34], along with intermediate steps in the diffusion process. The first column shows input images, while the subsequent five columns show the evolution of the point cloud from a randomly sampled Gaussian to a final shape over the course of the diffusion process. The following column shows the result of our coloring model applied to our reconstructed shape. The final two columns show renders of our colored point cloud from two randomly sampled novel views, demonstrating that it is a complete 3D shape. Our method is capable of reconstructing objects from real-world images with challenging viewpoints.

3. Method

In the following sections we describe our method. We begin with an overview of denoising diffusion models, which forms the foundation of our method. Next, we introduce our novel conditioning scheme PC^2 . Finally, we describe our filtering method $\text{PC}^2\text{-FM}$, which utilizes the probabilistic nature of diffusion to address the ill-posed nature of the single-view 3D reconstruction problem.

3.1. Diffusion Models

Diffusion denoising probabilistic models are general-purpose generative models inspired by stochastic differential equations and non-equilibrium thermodynamics. Diffusion denoising models are based on an iterative noising process, which adds noise to a sample $X_0 \sim q(X_0)$ from a target data distribution $q(X_0)$ over a series of steps. The stepsize

of each step of this process is given by a variance schedule $\{\beta_t\}_{t=0}^T$:

$$q(X_t|X_{t-1}) = \mathcal{N}(X_t; \sqrt{1 - \beta_t}X_{t-1}, \beta_t\mathbf{I})$$

Each $q(X_t|X_{t-1})$ is a Gaussian distribution, implemented by using the well-known reparameterization trick:

$$q(X_t|X_0) = \sqrt{\bar{\alpha}_t}X_0 + \epsilon\sqrt{1 - \bar{\alpha}_t}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=0}^t \alpha_s$, and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

In order to form a generative model, we consider the *reverse* diffusion process, which begins with a sample from the noise distribution $q(X_T)$ and then denoises over a series of steps to obtain a sample from the target distribution $q(X_0)$. Confusingly, this reverse process is sometimes also referred to as the “diffusion process.” In order to reverse the diffusion process, we learn the distribution $q(X_{t-1}|X_t)$ using a neural network $s_\theta(X_{t-1}|X_t) \approx q(X_{t-1}|X_t)$. We can

then (approximately) sample from $q(X_0)$ by starting with a sample $X_T \sim q(X_T)$ and then iteratively sampling from $q(X_{t-1}|X_t)$. When the sampling step size is sufficiently small, $q(X_{t-1}|X_t)$ is well approximated by an isotropic Gaussian with a fixed small covariance. Thus, one only has to predict the mean of $q(X_{t-1}|X_t)$, which in practice is parameterized as predicting the noise ϵ with a neural network $s_\theta(X_{t-1}|X_t)$.

3.2. Point Cloud Diffusion Models

We consider 3D point cloud with N points as a $3N$ dimensional object and learn a diffusion model $s_\theta : \mathbb{R}^{3N} \rightarrow \mathbb{R}^{3N}$. This network denoises the positions of a set of points from a spherical Gaussian ball into a recognizable object. At a very high level, at each step, we predict the offset of each point from its current location, and we iterate this process to arrive at a sample from our target distribution $q(X_0)$.

Concretely, the network is trained to predict the noise $\epsilon \in \mathbb{R}^{3N}$ added in the most recent time step using an L_2 loss between the true and predicted noise values:

$$\mathcal{L} = E_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} [\|\epsilon - s_\theta(X_t, t)\|_2^2]$$

At inference time, a random point cloud $X_T \sim N(\mathbf{0}, \mathbf{I}_{3N})$ is sampled from a $3N$ -dimensional Gaussian and the reverse diffusion process is run to produce a sample X_0 . At each step, we recover the mean $\mu_{\theta, t}$ of our approximation of $q(X_{t-1}|X_t)$ from the prediction $s_\theta(X_t, t)$, and we use this mean to sample from $q(X_{t-1}|X_t)$.

3.3. Conditional Point Cloud Diffusion Models

We formulate 3D reconstruction as conditional generation: the target distribution is the conditional distribution $q(X_0|I, V)$ where I is an input image and V is the corresponding camera view. The key question of this paper is how exactly one should condition the model on the reference image and camera view. The most immediate way of conditioning would be to provide a global learned embedding of the input image as an auxiliary input to the denoising function s_θ . This setup resembles prior work which used encoder-decoder architectures conditioned on image embeddings to generate 3D shapes [1, 54, 55].

However, this approach is lacking in that it only promotes a weak form of geometric consistency between the input image and the reconstructed shape. Empirically, preliminary experiments showed that it often generated plausible shapes, but that these shapes did not always match the input image from the given view (see Tab. 1).

This finding matches up precisely with the findings of [48], who analyzed voxel-based methods which decode from a global feature vector in a similar manner: they found that these networks primarily performed a sort of coarse classification of the input image into a small number of shape

categories and then generated a generic object from that category. Although coarse classification may produce adequate results in a highly curated synthetic setting such as ShapeNet, it will inevitably fail to capture the complexity of objects in real-world scenes. These insights led us to develop **PC²**, a geometrically consistent method for locally-conditional point cloud diffusion.

3.4. PC²: Projection-Conditional Diffusion Models

To address the geometric consistency issues observed above, our method **PC²** instead *projects* the image onto the partially-denoised point at each step in the diffusion process. We first process the image into a dense feature volume using a standard 2D image model (e.g. a CNN or ViT [3]). Next, before each diffusion step, we project these features onto the point cloud, attaching a neural feature to each point. Crucially, each point is augmented with a *different* neural feature, as determined by its location in space relative to the input camera and the input image.

Formally, let $I \in \mathbb{R}^{H \times W \times C}$ be the feature volume produced by our 2D image model applied to our input image, where C is the number of feature channels. s_θ is now a function $\mathbb{R}^{(3+C)N} \rightarrow \mathbb{R}^{3N}$ which predicts the noise ϵ from the augmented point cloud $X_t^+ = [X_t, X_t^{proj}]$. The projected features X_t^{proj} are given by $X_t^{proj} = \mathcal{P}_{V_I}(I, X_t)$ where \mathcal{P}_{V_I} is the projection function from camera view V_I , I is the input image, and X_t is the partially-noised point cloud.

One straight-forward approach to design $\mathcal{P}_{V_I}(I, X_t)$ would be to simply project the 3D points on the image and take the corresponding image features. However, this is akin to assuming that the point cloud is transparent. Instead, we found it beneficial to choose a projection function that properly accounts for self-occlusion. We implement it by rasterizing the points, assuming that they have some small, non-zero radius ρ . That is, \mathcal{P}_{V_I} is a rasterization procedure [33]. Due to highly optimized rasterization routines in modern software, this process is very efficient; it accounts for a negligible fraction of training time and computation. Full details about the rasterization process and the projected information are provided in the supplementary material.

Finally, we find that we can apply *exactly the same* projection-based conditioning to reconstruct object color in addition to object shape.¹ Specifically, we learn a separate coloring network $c_\theta : \mathbb{R}^{(3+C)N} \rightarrow \mathbb{R}^{CN}$, which takes as input a point cloud augmented with projection conditioning, and outputs the colors of each point. Empirically, we found that a single-step coloring model produced results comparable to a diffusion model. To reduce computational complexity, we hence utilize the single step model. :

¹Learning a coloring network is beneficial compared to a simple projection of colors from the image to points because a coloring network could learn to color an object backside differently from its front side (for example, in the case of a teddy bear).

Category	[1]	[56]	[55]	PC ²	PC ² -FM
airplane	0.225	0.215	0.266	0.473	0.589
bench	0.198	0.241	0.266	0.305	0.334
cabinet	0.256	0.308	0.317	0.203	0.211
car	0.211	0.220	0.268	0.359	0.372
chair	0.194	0.217	0.246	0.290	0.309
display	0.196	0.261	0.279	0.232	0.268
lamp	0.186	0.220	0.242	0.300	0.326
loudspeaker	0.229	0.286	0.297	0.204	0.210
rifle	0.356	0.364	0.410	0.522	0.585
sofa	0.208	0.260	0.277	0.205	0.224
table	0.263	0.305	0.327	0.270	0.297
telephone	0.407	0.575	0.582	0.331	0.389
watercraft	0.240	0.283	0.316	0.324	0.341
Average	0.244	0.289	0.315	0.309	0.343

Table 2. Comparison of single-view 3D reconstruction performance with prior work on ShapeNet-R2N2. F-Score is calculated using a distance threshold of 0.01 [54]. Our method performs similarly to prior work without filtering and outperforms prior work using filtering (both PC²-FA and PC²-FM); this filtering step leverages the method’s ability to produce multiple reconstructions for a given input image.

Category	PC ²	PC ² -FA	PC ² -FM	Oracle
airplane	0.473	0.517	0.589	0.681
bench	0.305	0.316	0.334	0.444
cabinet	0.203	0.246	0.211	0.303
car	0.359	0.369	0.372	0.420
chair	0.290	0.312	0.309	0.377
display	0.232	0.277	0.268	0.357
lamp	0.300	0.320	0.326	0.399
loudspeaker	0.204	0.235	0.210	0.288
rifle	0.522	0.538	0.585	0.686
sofa	0.205	0.242	0.224	0.298
table	0.270	0.293	0.297	0.420
telephone	0.331	0.400	0.389	0.523
watercraft	0.324	0.322	0.341	0.424
Average	0.309	0.337	0.343	0.432

Table 3. Single-view 3D reconstruction performance on ShapeNet-R2N2 for different levels of filtering. PC² produces a single sample for each input image. PC²-FM (see Sec. 3) uses 5 samples for automated IoU-based filtering. The “Oracle” method evaluates the upper bound of filtering: it computes the F-score of 5 samples using the ground truth point cloud and chooses the best sample for each input image. Filtering, both without object masks (PC²-FA) and with masks (PC²-FM) improves results substantially, but does not fully close the gap to the oracle.

Samples	1	2	3	4	5
Avg. F-Score	0.310	0.330	0.335	0.340	0.343

Table 4. Performance of PC²-FM on ShapeNet-R2N2 when filtering different numbers of samples. There is a large improvement (0.02 F-score) from a single sample (*i.e.*, no filtering) to two samples. Performance continues to increase with diminishing returns.

3.5. PC²-FM and PC²-FA: Filtering for 3D Reconstruction

Due to the ill-posed nature of the single-view 3D reconstruction problem, there are many possible ground truth shapes corresponding to each input image. Differently from most prior works [54–56, 58], our model is inherently probabilist, and thus admits sampling multiple different reconstructions from a single input.

We take advantage of this property by proposing to generate *multiple independent samples* for each input image and then filter these according to an automated criterion. Thus, we are able to generate a diverse set of outputs and choose the one which is “most plausible” for a given input image.

In practice, we propose two simple criteria for filtering, both of which involve the object silhouette. One of these uses additional mask supervision, while the other does not. In both cases, we begin by rendering each point cloud $\{X_0^{(i)}\}_{i=1}^N$ from the input camera view V . We render the points as balls with a small fixed radius, such that we obtain the silhouette $\hat{M}^{(i)}$ of our point cloud sample.

With mask supervision (PC²-FM). We compare each silhouette \hat{M} with the object mask M , which is extracted automatically using Mask-RCNN [8]. We then calculate their intersection-over-union $IoU(\hat{M}^{(i)}, M)$ and select the sample with the highest IoU score.

Without mask supervision (PC²-FA). Rather than using an object mask, we filter based on the *mutual agreement* between our predictions. Specifically, we compute the IoU of each prediction with all of the *other predictions* and select the mask with the highest average IoU . In this way, we are able to select high-quality masks without any additional supervision.

Finally, we emphasize that our filtering approach is a general one; the two methods explored here are by no means comprehensive.

4. Experiments

We will first describe our datasets, model, and the implementation details. We then discuss and analyze our results.

ShapeNet. The ShapeNet Dataset is a collection of 3D CAD models corresponding to categories in the WordNet lexical database. Due to its synthetic nature, ShapeNet is the standard dataset for quantitative evaluation of single-view

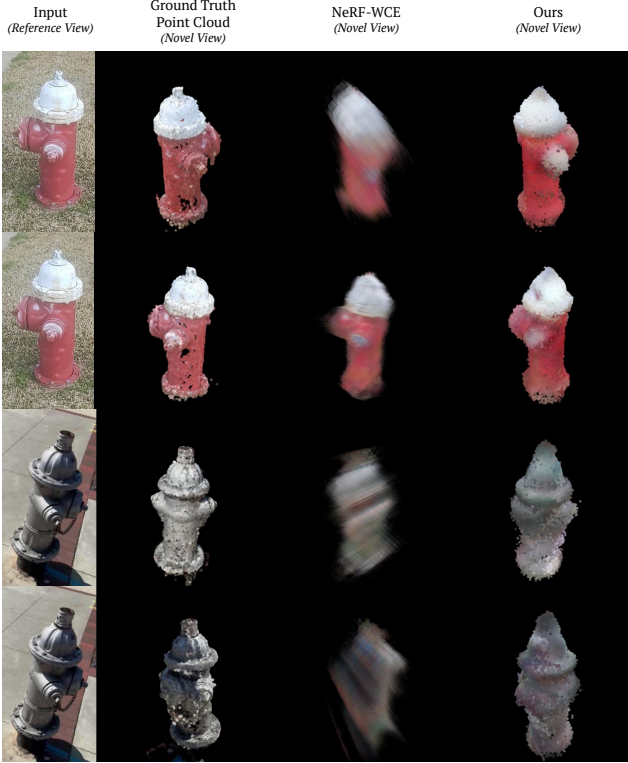


Figure 4. Comparison with NeRF-WCE [9] on the hydrants category of Co3D [34]. The first column shows the reference image, while the following three images show renders from a novel view (sampled randomly from a circle). The second column shows a render of the ground truth point cloud (obtained from COLMAP on the entire video). The third and fourth columns show renders from NeRF-WCE and our model. NeRF-WCE produces highly blurry outputs due to its deterministic nature. By contrast, our model produces realistic shapes from any viewpoint.

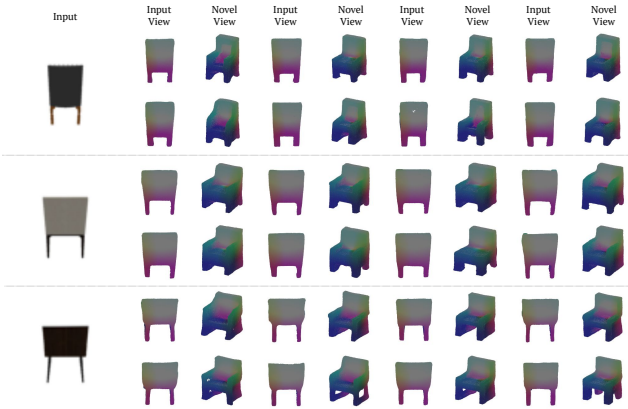


Figure 5. Diversity. The leftmost column shows a reference image from the chair category of ShapeNet, chosen specifically for its high shape ambiguity. The following images show generations produced by our model for eight random seeds. We see that our model is capable of producing significant variations in shape while still matching the input image from the reference viewpoint.



Figure 6. Examples of reconstructions produced by our method along with prior work on seven classes from the ShapeNet-R2N2 dataset [1]. The leftmost image in each row is the conditioning image. Given the simplicity and synthetic nature of ShapeNet, all models produce adequate results; however, our model produces reconstructions with the highest level of detail. We perform particularly well on challenging categories such as airplane and rifle.

reconstruction methods. We use the subset of 13 ShapeNet categories from 3D-R2N2 [1] along with their standard set of camera viewpoints, renderings, and train-test splits, which we collectively refer to as the ShapeNet-R2N2 benchmark. For evaluation, we use the widely-adopted F-score metric proposed by Tatarchenko [48]. See the supplementary material for further details on the ShapeNet and evaluation.

Co3D. The Co3D dataset is a challenging dataset consisting of multi-view images of real-world objects from common object categories. For training data, we use point clouds generated from running COLMAP [39, 40] on each image sequence. Through these experiment, we demonstrate that our method is able to reconstruct *real-world* objects using training data that is *derived only from multi-view images processed with COLMAP*.

We show results for three categories: hydrants, teddy bears, and toy trucks. Since there is no prior work on single-view reconstruction of Co3D, we train a NeRF-WCE [9] as

a baseline and present qualitative results (Fig. 4).

Implementation Details. For our diffusion model s_θ , we adopt a Point-Voxel CNN (PVCNN) [21] based on its success as described in [60]. This model processes a point cloud X_i using two branches simultaneously: a point-based branch which processes points individually and a voxel-based branch which aggregates nearby points in a hierarchical manner. In this way, the network is able to understand both the global and local structure of a point cloud.

We implement our model in PyTorch [29] and use the PyTorch3D library [33] for rasterization during the projection conditioning phase. Our point-voxel diffusion model is trained with batch size 16 for a total of 100,000 steps. For feature extraction, we use MAE [7]. For optimization, we use AdamW [15] with a $\beta = (0.9, 0.999)$ and a learning rate which is decayed linearly from 0.0002 to 0 over the course of training. We use images of size 137×137 px and point clouds with 8192 points, because prior work computes the F-Score metric using 8192 points. For our diffusion noise schedule, we use a linear schedule with warmup in which beta increases from $1 \cdot 10^{-5}$ to $8 \cdot 10^{-3}$. All experiments are performed on a single GPU.

Quantitative Results. In Tab. 2, we show results for the 13 classes in the widely-used ShapeNet-R2N2 benchmark. Without filtering, \mathbf{PC}^2 performs on-par with prior work. Examining the performance across categories, we see that our method performs better on categories with objects that have fine details, such as rifle and airplane.

With filtering, which is only possible due to the probabilistic nature of our conditional diffusion process, \mathbf{PC}^2 -FM consistently outperforms prior work; it improves upon the state-of-the-art on the majority of the object categories and on average.

Qualitative Results. Figure 3 and Fig. 4 show qualitative results of our method on the real-world Co3D dataset. Figure 3 visually demonstrates the intermediate steps of the diffusion process as well as the final coloring step. Figure 4 compares our generations to those from NeRF-WCE [9]. Since NeRF-WCE is deterministic, it struggles to model highly uncertain regions, and produces blurry images for novel views which are far from the reference view. By contrast, our method produces realistic object shapes that can be viewed from any viewpoint. Relative to prior methods, our method is able to generate shapes with a significantly finer level of detail.

Many additional qualitative results are shown in the supplementary material.

Diversity of Generations. One key benefit of our probabilistic approach to 3D reconstruction is that we are able to generate multiple plausible shapes for a given reference view, as shown in Fig. 5 on the ShapeNet benchmark. This figure contains eight generations with different random seeds for

three highly-ambiguous images in the chair category. Our method produces meaningful variations in shape, while always remaining consistent with the input image from the given view. This ability to sample makes it possible for us to develop our filtering method, which we analyze below.

Filtering Analysis. In Tab. 3, we compare the performance of our two filtering methods against our single-sample performance and an oracle. The oracle chooses the best sample for each input image according to its F -score; it provides an upper-bound to the improvements attainable via filtering. Our mask-free filtering method (\mathbf{PC}^2 -FA) improves results substantially and using masks (\mathbf{PC}^2 -FM) improves them further. However, gap between our method and the oracle suggests that there remains room for further improvement; this represents an interesting potential avenue for future work.

In Tab. 4, we compare our performance when filtering with different numbers of images. We find that using just two images significantly improves results, and adding additional images leads to further performance gains. Naturally, more and more samples further increase the performance but with diminishing returns.

Limitations. The primary limitation of our model is the need for point cloud ground truth for training. As mentioned in the introduction, this data is more accessible than commonly imagined, as it can be extracted from multi-view images or videos. However, point clouds obtained from these methods can be noisy; in Co3D, for example, many of the ground-truth point clouds for the hydrant category contain holes in locations that were not observed in the original video sequence. We find that our method is relatively robust to this type of noisy data, but we feel it is still important to discuss this limitation.

5. Conclusions

In this paper, we have proposed \mathbf{PC}^2 , a novel diffusion-based method for single-view 3D shape reconstruction. Our method iteratively reconstructs a shape by projecting image features onto a partially-denoised point cloud during the diffusion process. Empirically, our experiments demonstrate the effectiveness of \mathbf{PC}^2 at both reconstructing geometry and point color. Quantitatively, we outperform prior methods on synthetic benchmarks. Qualitatively, we are able to reconstruct objects with high levels of detail from challenging real-world images. With regard to future work, it would be interesting to scale our method to larger datasets and models. Given the success of scaling diffusion models of 2D images (e.g., DALL-E-2 [32]), we hope this work is a step along the path to developing similar models for 3D reconstruction.

Ethics. We use the ShapeNet and CO3D datasets in a manner compatible with their terms. The images do not contain personal data. ShapeNet models are used in a man-

ner compatible with the Data Analysis Permission. Please see <https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html> for further information on ethics.

Acknowledgments. L. M. K. is supported by the Rhodes Trust. A. V. and C. R. are supported by ERC-UNION-CoG-101001212. C. R. is also supported by VisualAI EP/T028572/1.

References

- [1] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *Proc. ECCV*, pages 628–644, 2016. 2, 3, 5, 6, 7
- [2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794, 2021. 3
- [3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5
- [4] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 2
- [5] Richard I. Hartley and Frederik Schaffalitzky. Reconstruction from projections using grassmann tensors. In *Proc. ECCV*, LNCS. Springer-Verlag, 2004. 2
- [6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2
- [7] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *Proc. CVPR*, volume abs/2111.06377, 2021. 8
- [8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *Proc. ICCV*, 2017. 6
- [9] Philipp Henzler, Jeremy Reizenstein, Patrick Labatut, Roman Shapovalov, Tobias Ritschel, Andrea Vedaldi, and David Novotny. Unsupervised learning of 3d object categories from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4709, June 2021. 2, 3, 7, 8
- [10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv:2006.11239*, 2020. 3
- [11] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12949–12958, 2021. 2
- [12] Wonbong Jang and Lourdes Agapito. CodeNeRF: Disentangled neural radiance fields for object categories. In *Proc. ICCV*, 2021. 2, 3
- [13] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Nips*, pages 364–375, 2017. 3
- [14] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. *Proc. CVPR*, 2019. 2
- [15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 8
- [16] Erwin Kruppa. *To determine an object from two perspectives with inner orientation*. Hölder, 1913. 2
- [17] Jonáš Kulháněk, Erik Derner, Torsten Sattler, and Robert Babuška. ViewFormer: NeRF-free neural rendering from few images using transformers. In *Proc. ECCV*, 2022. 3
- [18] Chen-Hsuan Lin, Oliver Wang, Bryan C. Russell, Eli Shechtman, Vladimir G. Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. In *Proc. CVPR*, June 2019. 2
- [19] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2021. 3
- [20] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 2
- [21] 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. 8, 12
- [22] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 3
- [23] Zhaoyang Lyu, Zhifeng Kong, Xudong XU, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. In *International Conference on Learning Representations*, 2022. 3
- [24] S. J. Maybank. The angular velocity associated with the optical flow field arising from motion through a rigid environment. *Proc. R. Soc. Lond.*, 401, 1985. 2
- [25] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 2
- [26] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2
- [27] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, 2020. 3
- [28] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2, 3

- [29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. [8](#)
- [30] 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. [3](#), [12](#)
- [31] 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. [12](#)
- [32] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. [2](#), [8](#)
- [33] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. [5](#), [8](#)
- [34] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021. [2](#), [4](#), [7](#)
- [35] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common Objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *Proc. CVPR*, 2021. [3](#)
- [36] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. Sharf: Shape-conditioned radiance fields from a single view. In *International Conference on Machine Learning*, pages 8948–8958. PMLR, 2021. [2](#)
- [37] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. ShaRF: Shape-conditioned radiance fields from a single view. In *Proc. ICML*, 2021. [2](#), [3](#)
- [38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. [2](#), [3](#)
- [39] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [7](#)
- [40] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. [7](#)
- [41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [3](#)
- [42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, 2020. [3](#)
- [43] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. volume 32, 2019. [3](#)
- [44] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. volume 33, pages 12438–12448, 2020. [3](#)
- [45] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. [3](#)
- [46] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018. [2](#)
- [47] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019. [2](#)
- [48] 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. [5](#), [7](#)
- [49] Bram Wallace and Bharath Hariharan. Few-shot generalization for single-image 3d reconstruction via priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3818–3827, 2019. [2](#)
- [50] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. [2](#)
- [51] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv.cs, abs/2106.10689*, 2021. [3](#)
- [52] Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In *International Conference on Learning Representations*, 2021. [3](#)
- [53] Yan Xia, Cheng Wang, Yusheng Xu, Yu Zang, Weiquan Liu, Jonathan Li, and Uwe Stilla. Realpoint3d: Generating 3d point clouds from a single image of complex scenarios. *Remote Sensing*, 11(22), 2019. [2](#)
- [54] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2690–2698, 2019. [2](#), [3](#), [5](#), [6](#)
- [55] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun. Pix2vox++: multi-scale context-aware 3d object reconstruction from single and multiple images. *International Journal of Computer Vision*, 128(12):2919–2935, 2020. [2](#), [3](#), [5](#), [6](#)
- [56] Farid Yagubbayli, Alessio Tonioni, and Federico Tombari. Legoformer: Transformers for block-by-block multi-view 3d reconstruction, 2021. [2](#), [3](#), [6](#)

- [57] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2, 3
- [58] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *Proc. CVPR*, 2021. 2, 6
- [59] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *Proc. CVPR*, June 2019. 2
- [60] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5826–5835, October 2021. 3, 8

6. Implementation Details

Here, we provide additional details about the model and projection procedure.

First, we discuss the Point-Voxel [21] model which is used to process the partially-denoised point cloud at each step of the diffusion process. As its name suggests, this 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 [30, 31] (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 [21], the 3D U-Net consists of four downsampling (“Set Abstraction”) layers followed by four upsampling (“Feature Propagation”) 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.

7. Additional Qualitative Examples

We provide additional qualitative examples of our method in Figs. 7, 8 and 10 to 12. Figures 7 to 9 show examples of reconstructions on additional categories of Co3D, including hydrants, teddybears, glasses, remotes, motorcycles, hairdryers, plants, and donuts. Fig. 10 contains a selection of the best reconstructions produced by our model for each category of ShapeNet, as ranked by F-score. Fig. 12 contains random examples of reconstructions produced by our model on ShapeNet. Finally, Fig. 11 shows a selection of the worst examples produced by our model for each category on ShapeNet, as ranked by F-score.

8. 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 Section 6. 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.

9. Analysis of Failure Cases

Failure cases of our model are shown in Fig. 11. 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 8th column of the 2nd row of the figure, it appears that the model generates a chair rather than a box. Similarly, in the 12th row of the 5th 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 7. **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.

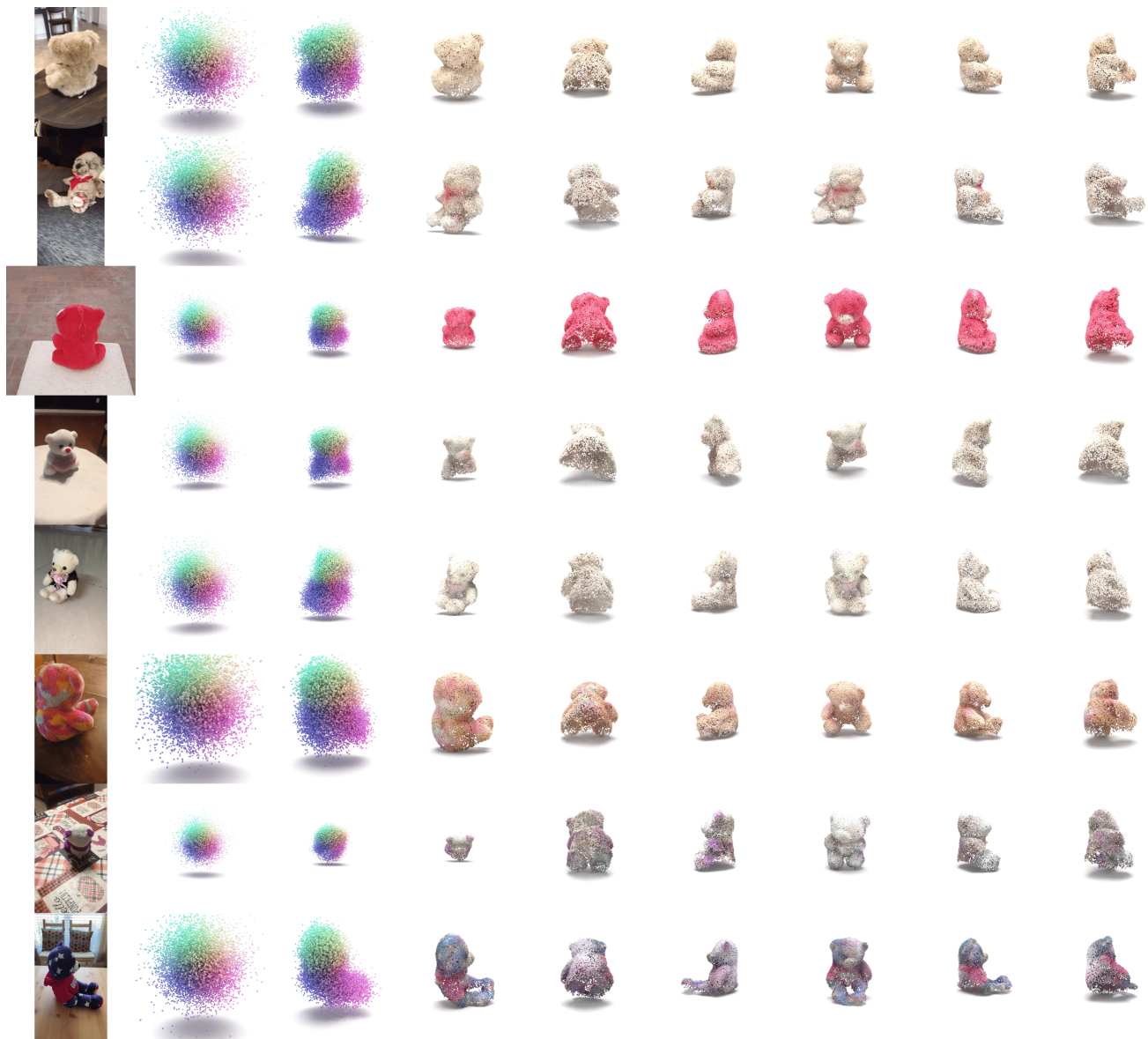


Figure 8. **Additional qualitative examples.** Examples from the teddy bear category of Co3D. The first column in each row shows the input image. The second and third columns show intermediate steps in the diffusion process. The fourth column shows the final reconstructed point cloud with color. The remaining five rows show the final predicted point cloud from novel views.

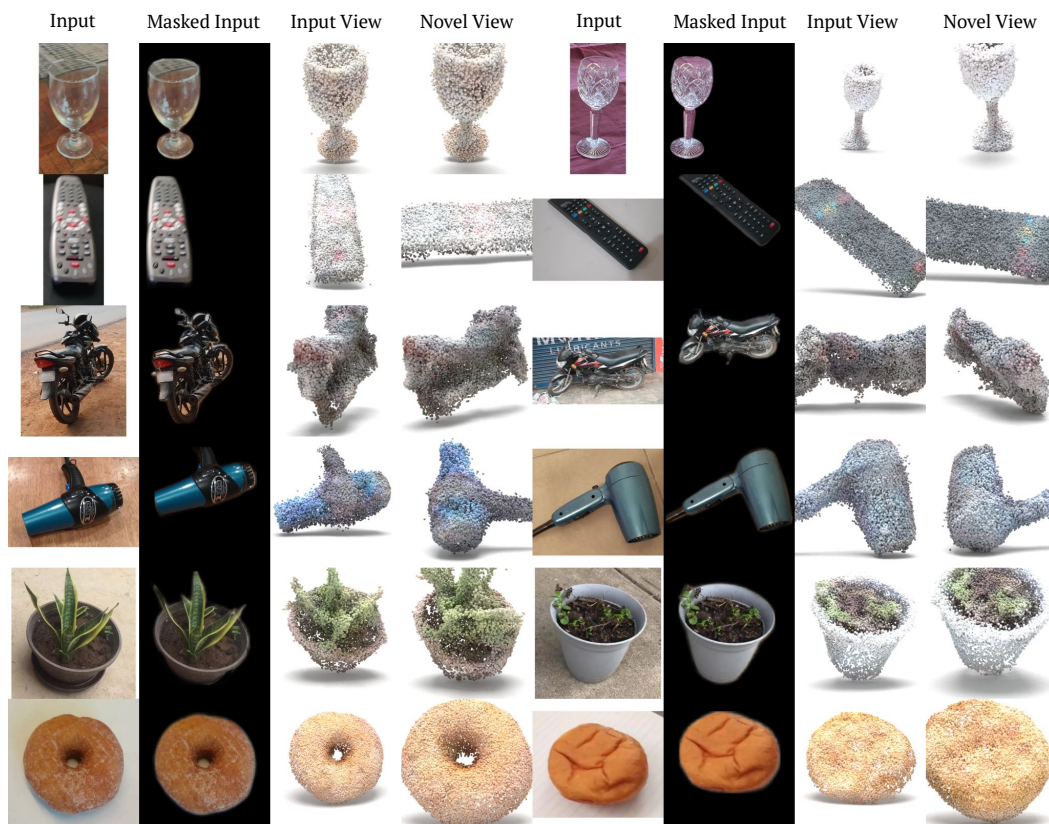


Figure 9. **Additional qualitative examples.** Examples from six additional categories: glasses, remotes, motorcycles, hairdryers, plants, and donuts.



Figure 10. **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 11. **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).

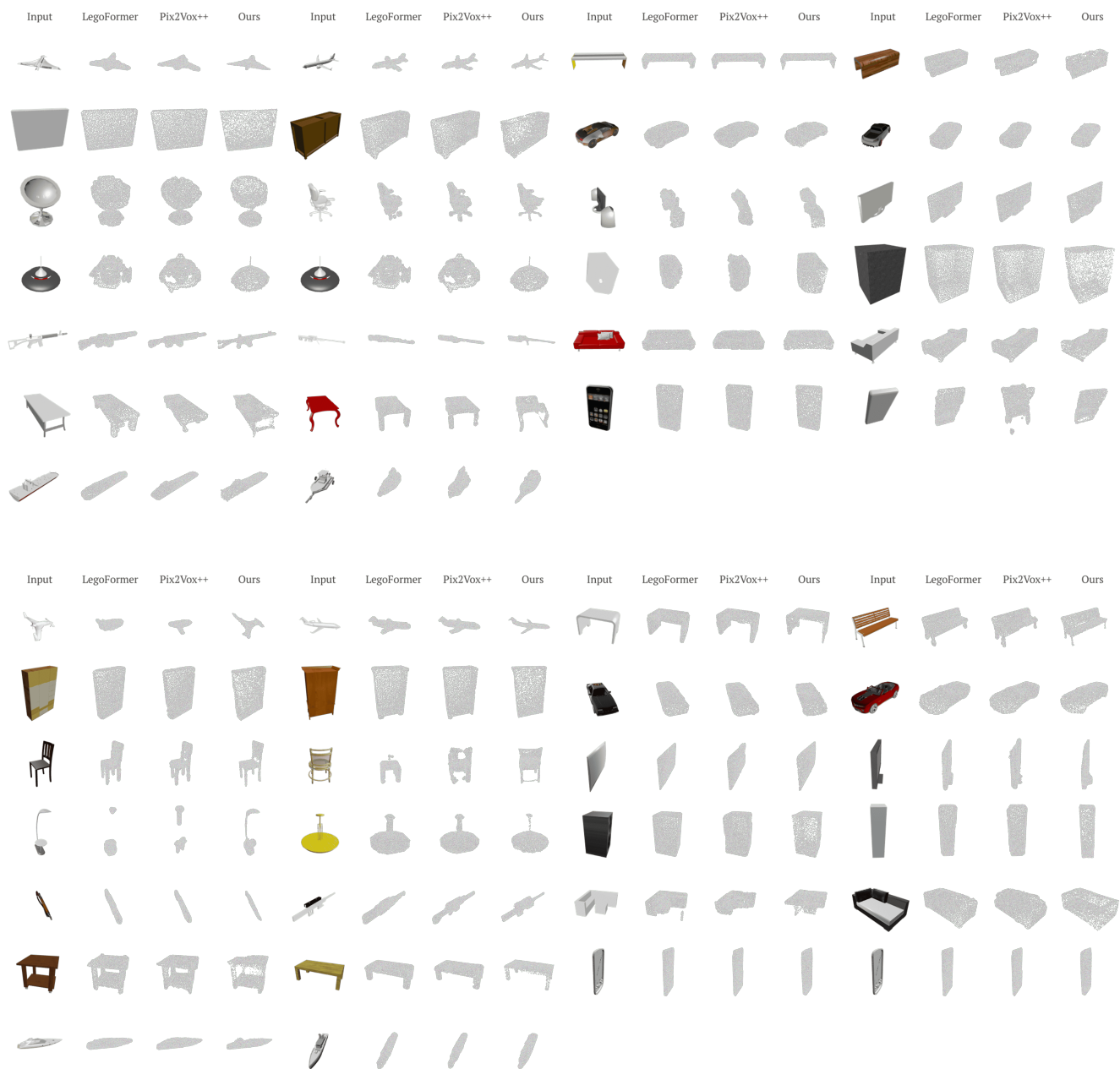


Figure 12. **Random examples** of our method along with prior work. The leftmost image in each set of images is the input image. Note that there are no images in the last row of the right half of the figure because we show examples for all 13 ShapeNet-R2N2 classes (seven on the left and six on the right).