Viewset Diffusion: (0-)Image-Conditioned 3D Generative Models from 2D Data

Stanislaw Szymanowicz   Christian Rupprecht   Andrea Vedaldi

Visual Geometry Group — University of Oxford
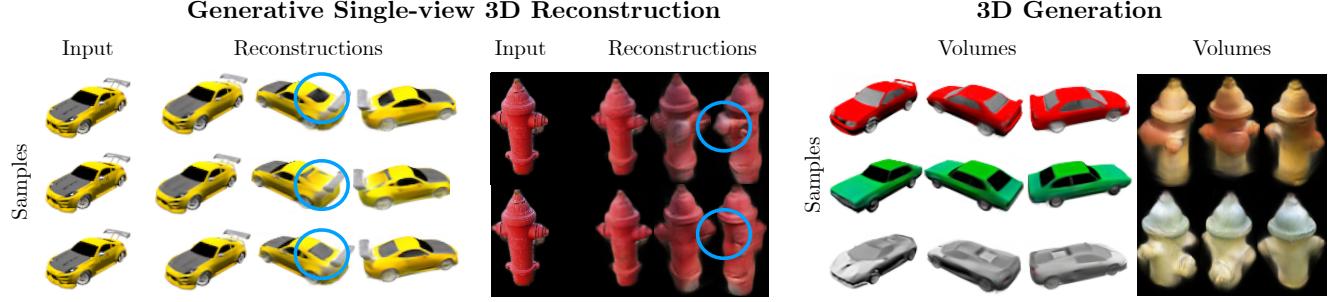
{stan,chrisr,vedaldi}@robots.ox.ac.uk

Figure 1: Viewset Diffusion. Our category-specific models perform both 'generative' 3D reconstruction and unconditional 3D generation. In single-view 3D reconstruction (left) our models generate plausible explanations of occluded regions (car's back, hydrant's occluded side). The same models are able to generate varied 3D objects (right) in a feed-forward manner while being trained only on 2D data.

Abstract

We present Viewset Diffusion*, a diffusion-based generator that outputs 3D objects while only using multi-view 2D data for supervision. We note that there exists a one-to-one mapping between viewsets,* i.e.*, collections of several 2D views of an object, and 3D models. Hence, we train a diffusion model to generate viewsets, but design the neural network generator to reconstruct internally corresponding 3D models, thus generating those too. We fit a diffusion model to a large number of viewsets for a given category of objects. The resulting generator can be conditioned on zero, one or more input views. Conditioned on a single view, it performs 3D reconstruction accounting for the ambiguity of the task and allowing to sample multiple solutions compatible with the input. The model performs reconstruction efficiently, in a feed-forward manner, and is trained using only rendering losses using as few as three views per viewset. Project page:* szymanowiczs.github.io/ viewset-diffusion.

1. Introduction

Image-based 3D reconstruction, *i.e.*, recovering the 3D shape of the world from 2D observations, is a fundamental problem in computer vision. In this work, we study the problem of reconstructing the 3D shape and appearance of individual objects from as few as one image. In fact, we cast this as image-conditioned 3D generation, and also consider the case of unconditional generation (fig. 1).

Single-view 3D reconstruction is inherently ambiguous because projecting a 3D scene to an image loses the depth dimension. The goal is thus not to recover the exact 3D shape and appearance of the object, particularly of its occluded parts, but to generate *plausible* reconstructions. This can only be achieved by learning a prior over the likely 3D shapes and appearances of the objects. Here, we do so for one category of objects at a time.

Leveraging 3D object priors for reconstruction has been explored by several works [11, 48]. Most of these tackle 3D reconstruction in a deterministic manner, outputting one reconstruction per object. This is limiting in the presence of ambiguity, as a deterministic reconstructor can only predict either (1) a single most likely solution, which is plausible but usually incorrect, or (2) an average of all possible reconstructions, which is implausible (fig. 2).

Thus, in this work, we tackle the problem of modelling ambiguity in few-view 3D reconstruction. Our goal is to learn a conditional generator that can sample *all plausible 3D reconstructions* consistent with a given image of an ob-

ject from a given viewpoint.

We approach this problem using Denoising Diffusion Probabilistic Models (DDPM) [14] due to their excellent performance for image generation [6]. However, while DDPMs are trained on billions of images, 3D training data is substantially more scarce. We thus seek to *learn a 3D DDPM using only multi-view 2D data* for supervision. The challenge is that DDPMs assume that the training data is in the same modality as the generated data. In our setting, the 3D model of the object can be thought of as an unobserved *latent variable*, learning which is beyond the scope of standard DDPMs. We solve this problem starting from the following important observation.

Given a 3D model of an object, we can render all possible 2D views of it. Likewise, given a sufficiently large set of views of the object, called a *viewset*, we can recover, up to an equivalence class, the corresponding 3D model. Because of this bijective mapping, generating 3D models is equivalent to generating viewsets. The advantage of the latter is that we often have access to a source of suitable 2D multi-view data for supervision. Similarly to 2D image generation, our corresponding DDPM takes as input a partially noised viewset and produces as output a denoised version of it. For generation, this denoising process is iterated starting from a Gaussian noise sample.

Our second key intuition is that the bijective mapping between viewsets and 3D models *can be integrated in the denoising network* itself. Namely, our DDPM is designed to denoise the input viewset by reconstructing a full radiance field of the corresponding 3D object (see fig. 3). This has the advantage of producing the 3D model we are after and ensuring that the denoised viewset is 3D consistent (the lack of 3D consistency is an issue for some multi-view generators [22, 47]). Furthermore, by allowing different views in the viewset to be affected by different amounts of noise, the same model supports conditional generation from any number of input views (including zero). This conditional generation is achieved by setting the noise level of the available conditioning images to zero.

We call our method *Viewset Diffusion* and, with it, make several contributions: (i) The idea of generating viewsets as a way to apply DDPMs to the generation of 3D objects even when only multi-view 2D supervision is available. (ii) An ambiguity-aware 3D reconstruction model that is able to sample different plausible reconstructions given a single input image, and which doubles as an unconditional 3D generator. (iii) A network architecture that enables our reconstructions to match the conditioning images, aggregate information from an arbitrary number of views in an occlusion-aware manner and estimate plausible 3D geometries. (iv) A new synthetic benchmark dataset, designed for evaluating the performance of single-image reconstruction techniques in ambiguous settings.
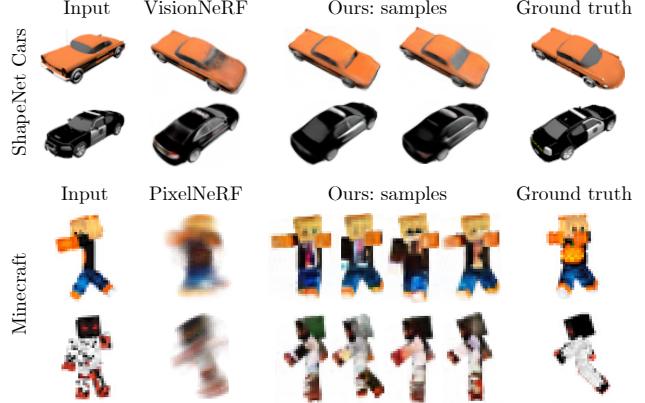

Figure 2: **Ambiguities.** Under occlusion, deterministic methods blur possible shapes (orange car's back, Minecraft characters' poses) and colours (black car's back, occluded sides of Minecraft characters). Our method samples plausible 3D reconstructions.

2. Related Work

Works most related to ours consider the problem of few-view 3D reconstruction and the ensuing ambiguities.

Reconstructing Neural Fields. Few-view reconstruction methods that use Neural Radiance Fields [25] include Sin-NeRF [51] and DietNeRF [15]. They use semantic pseudo-labels in unseen views to provide multi-view pseudo-constraints from pre-trained 2D networks, effectively leveraging a 2D prior. Other works learn a 3D prior instead and represent individual shapes using global latent codes [16, 27, 31, 36]. Codes be optimised at test-time via auto-decoding, akin to latent space inversion in priors learned with 3D GANs [2]. The latent codes can also be local [13, 21, 53] or simultaneously global and local [21], which tends to improve high-frequency details. While our method borrows the idea of local conditioning and learning a 3D prior from such prior works, a key difference is that we *sample different plausible reconstructions*, while most prior works only output a *single reconstruction*, which tends to regress to the mean, usually falling outside the data distribution, being blurry, and mixing several modes in one. In contrast, our method samples multiple sharp reconstructions, each of which is different yet plausible.

Reconstruction Beyond Neural Fields. Many other possible 3D representations have been explored, including geometry-free representations [30, 39, 40, 45], occupancy voxel grids grids [4, 42, 52], textured meshes [18, 49] or hybrid implicit-explicit representations [33, 50]. While our work is currently based on a neural radiance field, it is compatible with any differentiable formulation.

Ambiguity in 3D Reconstruction. Single-view 3D reconstruction is an ill-posed problem because a 2D input only partially constrains the 3D output. A 3D prior can re-

duce but not eliminate the ambiguity. In fact, even when the 3D reconstruction is constrained to be plausible according to the given prior, it does not mean that it is unique [48]. For instance, when we reconstruct a person seen from the front, even knowing that it is a person is insufficient to exactly predict their back. The goal is to obtain one or several plausible solutions that are compatible with the given observations, for example via constrained optimization [11] or using an adversarial loss during training [48]. Ignoring ambiguity may result in distortions, including blurry reconstructions without fine details [5].

We embrace ambiguity in 3D reconstruction and train a network to (1) output a 3D object that matches an observation such that (2) the 3D object is a plausible member of a given category. This setting is similar to Wu *et al.* [48], but we allow sampling different plausible reconstructions via a conditional diffusion model, rather than finding a single plausible solution via a constrained optimisation approach.

3D Modelling with Diffusion Models   Recently, Denoising Diffusion Probabilistic Models [14] (DDPM) have been applied to modelling 3D shape distributions by diffusing directly in the space of 3D representations, including point clouds [23], triplanes [10, 34, 46] and other radiance fields [17, 26] Shortcomings of these approaches include (1) assuming an available 3D dataset (see also section 3.2) and (2) requiring heuristics for dealing with 'floater' artefacts [26, 34] common in volumes reconstructed from multi-view data in an optimisation setting.

Others leverage pre-trained 2D diffusion models via Score Distillation Loss for text-to-3D generation [29, 44] with test-time optimisation, and extensions allow image-based reconstruction [20, 24]. Concurrently to our work, several others [3, 22, 47] learn image-conditioned diffusion models. The outputs are 2D, and their 3D consistency is only approximate, with frequent flickers [3, 22, 47]. 3D consistency can be enforced via costly (sometimes 1 hour [55]) test-time optimization of a 3D representation [8, 55]. In contrast to prior and concurrent works, our method (1) can be trained on **2D data**, requiring only 3 views of an object, (2) is guaranteed to be 3D consistent, and (3) is feed-forward, and therefore much faster than test-time distillation methods [8, 24, 55]. HoloDiffusion [19] also learns 3D generative models from 2D data, but it only considers unconditional generation, while we propose a principled, unified framework for conditional and unconditional generation.

The closest work to ours is RenderDiffusion [1], discussed in detail in section 3.3.

3. Method

We consider the problem of learning a distribution over 3D objects, supporting both unconditional sampling and sampling conditioned on one or more views of the object. We approach this problem using a 3D DDPM and rethink the training setup to allow training it from multi-view 2D data, without access to 3D ground-truth.

3.1. DDPMs: background and notation

Consider the problem of learning a distribution $p(x)$ over 2D images $x \in \mathbb{R}^{3 \times H \times W}$ (or, with little changes, a distribution $p(x|y)$ conditioned on additional information y such as a text description). The DDPM approach generates a sequence of increasingly noisier versions of the data. This sequence starts from $x_0 = x$ and adds progressively more Gaussian noise. such that the conditional distribution $p(x_t|x_0)$ at step t can be characterised by writing $x_t = \sqrt{1 - \sigma_t^2} x_0 + \sigma_t \epsilon_t$, $t = 1, \ldots, T$, where σ_t is a sequence of noise standard deviations increasing from 0 to 1, ϵ_t is normally distributed. The marginal distribution $p(x_t)$ does not, in general, have a closed-form solution but for large $t \approx T$ it approaches a normal distribution.

In order to draw a sample x_0, one starts backwards, drawing first a sample x_T from the marginal $p(x_T)$, and then taking samples x_t from $p(x_{t-1}|x_t)$, until x_0 is obtained. The key observation is that these are comparatively simple distributions to learn. Various slightly different formulations are possible; here, we learn a denoising network $\hat{x}_0(x_t, t)$ that tries to estimate the "clean" sample x_0 from its noisy version x_t. Given a training set $\mathcal{X}$ of images, such a network is trained by minimizing the loss

$$\mathcal{L}(\hat{x}_0, t) = \frac{1}{|\mathcal{X}|} \sum_{x_0 \in \mathcal{X}} w(\sigma_t) \, \mathbb{E}_{p(x_t|x_0)} \|\hat{x}_0(x_t, t) - x_0\|^2$$

where the weight $w(\sigma_t)$ depends on the noise/timestep [12].

3.2. The challenge of a 3D extension

We now consider using a DDPM to learn a distribution $p(v)$ of 3D models v of objects (in practice radiance fields, see section 3.4). In oder to train a DDPM to generate 3D models v, we would require a dataset $\mathcal{V}$ of such models. Differently from 2D images, however, 3D models are not readily available. We assume instead to have access to 2D multi-view training data. Each training sample is a *viewset* $(\boldsymbol{x}, \Pi)$, *i.e.*, a collection $\boldsymbol{x} \in \mathbb{R}^{N \times 3 \times H \times W}$ of N views of a 3D object with known camera poses $\Pi = \{\pi^{(i)}\}_{i=1}^N$. The 3D model $\boldsymbol{v}$ is not observed but a *latent variable*.

The fact that $\boldsymbol{v}$ is a latent variable suggests adopting the *latent diffusion* approach, which has been very successful for 2D images [32], and thus simply replace the input data $\boldsymbol{x}$ with corresponding codes $\boldsymbol{v}$. Unfortunately, doing so requires to know the encoder $\boldsymbol{x} \mapsto \boldsymbol{v}$, mapping the input data $\boldsymbol{x}$ to the latents $\boldsymbol{v}$. In our case, this mapping amounts to image-based 3D reconstruction, which is non-trivial.

One way of implementing the mapping $\boldsymbol{x} \mapsto \boldsymbol{v}$ is to use an optimization method like NeRF, which can recover the
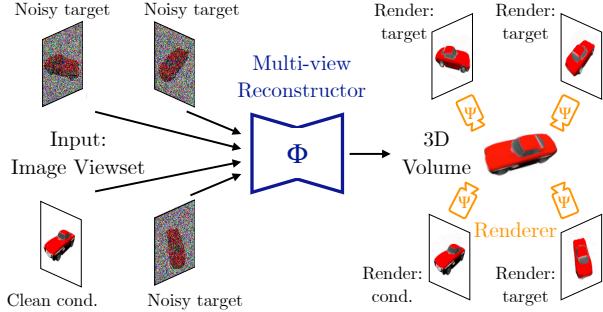
Figure 3: Viewset Diffusion takes in *any* number of clean conditioning images and generated images with Gaussian noise (section 3.3). The denoising function is defined as reconstructing (section 3.4) and rendering a 3D volume. When there is at least one clean conditioning view, Viewset Diffusion samples plausible 3D reconstructions. When all input views are noisy, it samples a 3D object form the prior.

radiance field v given a sufficiently large viewset x. This is the approach taken by several prior works [10, 46], but it has several shortcomings. First, with no prior information, reconstructing a model v from a viewset x is ambiguous due to visibility (the interior of an object does not matter to its appearance) and over-parameterization. While we can dismiss these ambiguities as classes of equivalent models[1], nevertheless they increase the irregularity of the distribution $p(v)$ that we wish to learn [46]. Second, good independent reconstructions require a fairly large (≥ 50) number of views per object, which are not always available, and even then they may still contain defects such as floaters [34]. Lastly, optimization-based reconstruction is slow (hours per sample) and must happen for every sample $x \in \mathcal{X}$ before training the distribution $p(v)$ can even start.

3.3. Viewset diffusion

In this section, we seek to directly train a DDPM to generate 3D models using only 2D supervision. Our approach is centred around a few simple but powerful observations. First, we note that it is easy to apply DDPMs to viewsets x because, differently from the 3D radiance fields v, they are assumed to be observable. Second, while DDPMs do not directly support generating latent variables, we can interpret the latent 3D model v as an *intermediate viewset representation* learned by the neural network which implements the DDPM — we simply do not apply diffusion to it.

Concretely, we write the DDPM denoising function as the composition of two functions. The first is an encoder network

$$v = \Phi(x_t, \Pi, \sigma_t) \qquad (1)$$

which, given a *noised viewset* (x_t, Π), produces as output a

3D model v. This 3D model is then *decoded* into an estimate

$$\hat{x}_0(x_t) = \Psi(v, \Pi) = \Psi(\Phi(x_t, \Pi, \sigma_t), \Pi) \qquad (2)$$

of the *clean viewset* by the decoder Ψ that implements differentiable rendering. This is the same formulation as standard image-based diffusion, except that (1) one generates a set of views in parallel instead of a single image and (2) the denoiser network has a particular structure and geometric interpretation. The training loss is the same as for standard diffusion:

$$\mathcal{L}(\Phi, x_0, x_t, \Pi, t) = w(\sigma_t) \left\| \Psi(\Phi(x_t, \Pi, \sigma_t), \Pi) - x_0 \right\|^2$$

where $x_t = \sqrt{1 - \sigma_t^2}\, x_0 + \sigma_t \varepsilon_t$ is a noised version of the (clean) input viewset x_0.

Single and few-view reconstruction. With the model above, we can learn *simultaneously* unconditional 3D generation as well as single and few-view reconstruction with almost no changes. Given a conditioning viewset (y, Π'), in fact, we can sample $p(x|\Pi, y, \Pi')$ by feeding into the network Φ a mixture of noised and clean views:

$$v = \Phi(x_t \oplus y, \Pi \oplus \Pi', \sigma_t \oplus \mathbf{0})$$

where $\oplus$ denotes concatenation along the view dimension. Here $\sigma_t \oplus \mathbf{0}$ means that we treat σ_t as a vector of noise variances, one for each view in the viewset, and append zeros to denote the fact that the conditioning views y are "clean".

Discussion. The approach above learns a distribution $p(x)$ over viewsets rather than a distribution $p(v)$ over 3D models. As noted in section 1, however, viewsets and 3D models can be thought to be in one-to-one correspondence, so sampling one is equivalent to sampling the other. While this statement is correct in the limit of infinitely-large viewsets,[2] crucially reconstruction in our case is performed by a network Φ. The benefit is that this reconstruction network can learn a 3D data prior and use it to perform 3D reconstruction with much greater data efficiency. In fact, we use as few as 3 images per viewset, which are far from sufficient to optimise a radiance field from scratch.

Our approach is also related to RenderDiffusion (RD) [1], but with substantial theoretical and practical differences. First, using our notation, their approach amounts to reducing the size of the viewset to a *single view*, which is insufficient to adequately represent a 3D object v. In our case, by using a non-trivial viewset, the generation of successive denoised samples ensures coherent and plausible appearance and shape from *all* viewpoints, which is not guaranteed in RD, which only denoises a *single* viewpoint. We also introduce architectural advancements in the form of local conditioning and multi-view attention-based aggregation, further improving quality.

---

[1]In the sense that these models all produce the same images.

[2]And ignoring inconsequential reconstruction ambiguities

Input: 2D features

Output: Explicit Radiance Field

Canonically aligned 3D features

■ Per-frame features ◤ Geometric unprojection
■ Per-category features → 3D convolutions + self-attention
■ Per-viewset features ⤍ Cross-attention queries
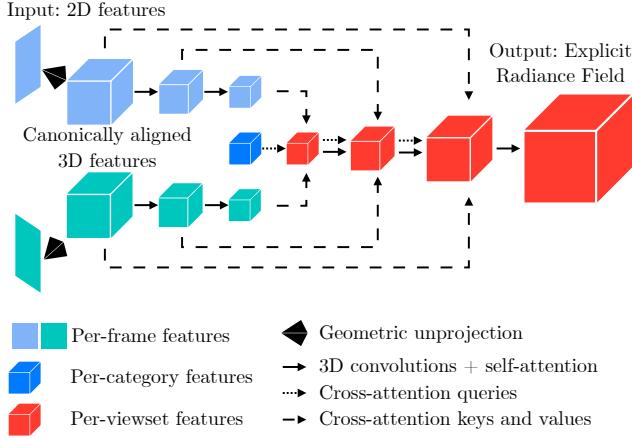 ⤍ Cross-attention keys and values

Figure 4: Architecture. 2D input views are unprojected along camera rays to a canonical feature volume. Multi-scale features are extracted and aggregated with an attention mechanism to output a single radiance field. The number of input views can be variable.

3.4. Radiance fields and network architecture

Having discussed the learning formulation, we now describe key implementation details. A 3D model $v = (\rho, c)$ is a *radiance field*, i.e., a pair of functions ρ and c mapping a 3D point $q \in \mathbb{R}^3$ to an opacity value $\rho(q) \geq 0$ and an RGB color $c(q) \in [0, 1]^3$. For simplicity, we discretize the radiance field over a 3D grid, expressing it as a tensor $v \in \mathbb{R}^{4 \times H \times W \times D}$, and evaluate $(\rho(q), c(q))$ using trilinear interpolation and padding as needed. Similar to DVGO [41], the voxel grid stores colors and opacities pre-activation, and activations are applied only after trilinear interpolation. Given the camera π (specified by rotation, translation and intrinsic parameters such as the focal length), then eq. (2) *renders* the image $x = \Psi(v, \pi)$ in a differentiable manner via ray casting.

The goal of the network Φ of eq. (1) is to output the 3D model v given as input the viewset x_t (including the cameras Π) and the noise variance σ_t for each view. Our network consists of the following stages (see fig. 4):

1. 2D feature extraction. A small 5-layer 2D convolutional subnetwork f outputs a feature map $F^{(i)} = f(x_t^{(i)})$ for each image $x_t^{(i)}$ in out of N images in viewset x_t.

2. Geometric unprojection. For each feature map $F^{(i)}$, associated camera pose $\pi^{(i)} = (R^{(i)}, T^{(i)})$ and the camera intrinsic matrix K, we form a volume $V^{(i)} \in \mathbb{R}^{C \times H \times W \times D}$. Voxel with centre at location $q = (i, j, k)$ holds feature $F^{(i)} \left[K \left(R^{(i)} | T^{(i)} \right) \tilde{q} \right]$, where $[\cdot]$ denotes bilinear interpolation, $\tilde{\cdot}$ denotes homogeneous coordinates and $(\cdot | \cdot)$ denotes column-wise matrix concatenation. After this step, volumes $V^{(i)}$ for different images $x_t^{(i)}$ share the same global reference frame, so they "line up". Unprojection allows the vol-

ume to easily match the conditioning image.

3. Per-frame 3D U-Net encoder. We use the same U-Net encoder as in DDPM [14], but replace the 2D convolutional and self-attention blocks with their 3D equivalents, similarly to [26]. The encoder outputs multi-scale feature maps $\{W_j^{(i)}\}_{j=1}^M$, with $j = 1$ being the finest and $j = M$ the coarsest feature map, for each input volume $V^{(i)}$. We also pass the timestep t (and thus, implicitly, the noise level σ_t) to the encoder after via the Transformer sinusoidal positional embedding [43]. Similarly to DDPM [14], the timestep modulates the U-Net Convolutional blocks via FiLM [7]. Each input volume $V^{(i)}$ is processed independently by the encoder, hence it accepts the individual noise level $\sigma^{(i)}$ for corresponding image $I^{(i)}$.

4. Multi-view 3D U-Net decoder. The decoder acts as a multi-scale multi-view feature aggregator. At each level j of the U-Net, the decoder aggregates features the feature maps $\{W_j^{(i)}\}_{i=1}^N$ at level j with an attention mechanism: $W_{j-1}' = \text{Attn}(Q = Q_j, K = V = \{W_j^{(i)}\}_{i=1}^N)$. The query Q_M at the coarsest level is fixed and learnt per-class. Attention operates at each voxel location independently to minimise computational complexity. Learnt attention-based aggregation (instead of mean-averaging) means that the combination of features across views can depend on, for example, occlusion. At each feature map level j, the aggregated features W_{j-1}' are then upscaled, passed through convolutional and self-attention blocks, identically to usual U-Nets used in diffusion models [14] to output $Q_{j-1} = h(W_{j-1}')$, before aggregation at the finer level $j - 1$.

5. Upscaling. Finally, a small 5-layer 3D convolutional subnetwork g performs upscaling $v = g(Q_0)$ to output the reconstructed volume v. The output of the network is a single volume v for the N input views in viewset x_t.

We validate our design choices, including the use of unprojection and attention-based aggregation in table 4.

3.5. Training and inference details

Training. To train our model, we consider a dataset $\mathcal{X}$ of viewsets (for example, from CO3D). We further subsample each viewset extracting at random N_{train} views (x, Π) that will be passed at input to the network, where $N_{\text{train}} \in 1, 2$, so that $x \in \{\{x^{(1)}\}, \{x^{(1)}, x^{(2)}\}\}$, and an additional unseen view (x_u, π_u), which is unavailable to the network. We sample the noise level t, and with it the scalar noise variance $\bar{\sigma}_t$ according to the cosine schedule of [28]. We then randomly apply Gaussian noise to some of the input views, such that noise standard deviations can be in one of three states $\sigma_t \in \{\{\bar{\sigma}_t\}, \{\bar{\sigma}_t, \bar{\sigma}_t\}, \{\bar{\sigma}_t, 0\}\}$, corresponding to one noised view, two noised views, or one clean and one noised view, respectively. These three options are sampled with probability $[0.45, 0.45, 0.1]$, respectively. The noised

viewset $\boldsymbol{x}_t = \alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\varepsilon}_t$ is input to the network[3].

Loss. We optimise the network Φ by minimising the photometric $L2$ loss (section 3.3), of the renders from reconstructed volume:

$$\mathcal{L} = \sum_{i=1}^{N} w(\sigma^{(i)}) \|\Psi(\Phi(\boldsymbol{x}_t, \Pi, \sigma_t), \pi^{(i)}) - x^{(i)}\|^2$$
$$+ \lambda_t \|\Psi(\Phi(\boldsymbol{x}_t, \Pi, \sigma_t), \pi_u) - x_u\|^2. \quad (3)$$

The weights $w(\sigma^{(i)})$ are set according to the Min-SNR-5 strategy [12]. The loss $\mathcal{L}$ also includes a penalty on the unseen view (x_u, π_u), as a regularisation strategy to encourage 3D consistency. The weight of the unseen view $\lambda_t = \lambda \cdot \min_i w(\sigma^{(i)})$ is also noise-dependent, where λ is a hyperparameter.

Inference. Inference is performed by progressive denoising of the viewset $\boldsymbol{x}$. The size of the viewset N_{inf} used at inference depends on the dataset and its complexity — we use $N_{\text{inf}} = 5, 3, 4$ for CO3D, ShapeNet and Minens, respectively. When performing single-view reconstruction, the viewset $\boldsymbol{x}$ includes the clean conditioning view, which is unchanged during denoising the viewset. In unconditional 3D generation, images in the viewset $\boldsymbol{x}$ are initialised to samples from Gaussian distribution. We use DDIM [38] sampling with 250 steps.

4. Experiments

Datasets. We evaluate our method at 128×128 resolution on *ShapeNet-SRN Cars* [37] using the standard train/val/test split [37] and protocol: view 64 is used for conditioning and the remaining 250 views as novel, unseen prediction targets. We also evaluate our method on four object classes from *CO3D* [30]: Hydrant, Teddybear, Plant and Vase. For each object class, we form a small testing set with 100 examples of randomly sampled image pairs and associated camera poses, one for conditioning and the other as the target, from randomly sampled test instances. Pre-processing details for CO3D are in the sup. mat.

We also introduce *Minens* (see fig. 5), a new dataset that makes it easier to evaluate ambiguity and diversity in 3D reconstruction. We design it to be large in the number of instances, while sufficiently small for rigorous experimentation using academic resources. Each object in Minens consists of a torso with randomly articulated arms, legs and head and is textured with one of 3,000 skins. We render 40,000 training and 5,000 validation examples with OpenGL at 256×256 resolution and downsample to 48×48 with Lanczos interpolation. Each example consists of 3 images and associated camera poses Π. We form two test sets

Figure 5: Minens dataset. Textured meshes are articulated and rendered from random camera viewpoints, allowing for procedural generation of a large number of instances.

with the Minens dataset: 'Random', with randomly sampled skins, poses and camera viewpoints, and 'Ambiguous', with manually-selected 3D poses that are ambiguous when seen from a single viewpoint, *e.g.*, due to one arm being occluded by the torso. We use different skins for training and testing. Similar to our subset of CO3D above, we select 100 test samples, consisting of one conditioning image and one target image, from different viewpoints. Code and the Minens dataset are available at `szymanowiczs.github.io/viewset-diffusion`.

Evaluation protocol. We render the reconstructed object from the target viewpoint(s) and measure the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) and Learned Perceptual Image Patch Similarity [54] (LPIPS), measured with a pre-trained VGG Net [35], when compared to the ground truth image. However, deterministic baselines like PixelNeRF produce a single average reconstruction which is optimal for squared-error metrics like PSNR: the 'average' sample is blurry (fig. 2), but closer in PSNR to the ground truth than most samples taken from the correct posterior distribution.[4] Hence, by only looking at metrics like PSNR, it is simply not possible to measure the benefits of modelling ambiguity. To measure the latter, we take multiple samples of reconstructions of every object and report the *best* PSNR and SSIM from these samples. For Minens and CO3D we use 100 samples per testing instance. For ShapeNet-SRN, we take 20 samples due to the computational burden (the benchmark tests 175,000 generated views). As LPIPS is not as strongly affected by this property since it measures perceived visual similarity, here we report the *average* across all 100/20 samples. For completeness, we also report the best LPIPS across all samples and the average PSNR and SSIM, but we report them in brackets '$(\cdot)$' as they do not measure well what we want.

Baselines. Our primary aim is to show the importance of modelling reconstruction ambiguity by showing that this results in sharper and possibly more accurate reconstructions

---

[3]With a slight abuse of notation, this applies a given noise level to the corresponding image. Also note that some elements of σ_t can be 0, therefore $\boldsymbol{x}_t$ can also contain clean (non-noised) images.

[4]By definition, the mean of a distribution has the minimum average square distance to all samples, and this distance is significantly less than the average squared distance between pairs of samples, particularly in high-dimensional spaces like images.

Method	Random		Ambiguous	
	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
RenderDiffusion	19.85	0.213	16.33	0.236
PixelNeRF	21.55	0.220	17.86	0.250
RenderDiffusion++	24.18	0.157	19.92	0.210
Ours w/o $\mathcal{D}$	24.63	0.115	20.26	0.156
Ours w $\mathcal{D}$ - best	**24.82**	(0.072)	**21.50**	(0.081)
Ours w $\mathcal{D}$ - mean	(22.81)	**0.107**	(18.62)	**0.130**

Table 1: **Single view reconstruction - Minens.** Ours achieves larger gains in the Ambiguous subset, showcasing the strength of probabilistic modelling.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
3DiM	21.01	0.57	-
LFN	22.42	0.89	-
SRN	22.25	0.88	0.129
CodeNeRF	22.73	0.89	0.128
FE-NVS	22.83	(0.91*)	(0.099*)
VisionNeRF	22.88	<u>0.90</u>	**0.084**
PixelNeRF	23.17	0.89	0.146
Ours w/o $\mathcal{D}$	<u>23.21</u>	<u>0.90</u>	0.116
Ours w $\mathcal{D}$ - best	**23.29**	**0.91**	(0.094)
Ours w $\mathcal{D}$ - mean	(22.72)	(0.90)	<u>0.099</u>

Table 2: **Single view reconstruction — ShapeNet Cars.** Ours achieves the best PSNR, with the additional benefit of probabilistic treatment. *FE-NVS optimises SSIM in training, affecting perceptual sharpness.

than deterministic predictors. We compare against other reconstructors trained using 2D data: the fully-deterministic PixelNeRF [53], our reimplementation of RenderDiffusion (RD) [1], and our improvement over it, RD++. We train PixelNeRF using the publicly available code with a tuned learning rate and softplus activation for improved training stability and reimplement RD based on publicly available information. Since RD uses only single images for training (using our notation from section 3.5: $N_{train} = 1$, $N_{inf} = 1$, $\sigma_t = \{\bar{\sigma}_t\}$ and $\lambda = 0$) and a weaker architecture, for fairness we also consider the variant RD++. RD++ uses the architecture from section 3.4 (like Viewset Diffusion), takes as input a single noised image (like RD) and is trained with multi-view supervision (like Viewset Diffusion): $N_{train} = 1$, $\sigma_t = \{\bar{\sigma}_t\}$ and $\lambda \neq 0$. At inference time RD++, like RD, is capable of 3D generation by diffusing over a single view $N_{inf} = 1$, starting from pure Gaussian noise. Like RD, RD++ performs single-view reconstruction in a deterministic manner by accepting one clean input view $N_{inf} = 1$, $\sigma_t = \{0\}$. Finally, to evaluate the importance of a probabilistic model, we include a baseline of our method without diffusion (*i.e.*, one that receives a clean image and directly regresses a 3D volume): $N_{train} \in \{1, 2\}$, $\sigma_t \in \{\{0\}, \{0, 0\}\}$, $N_{inf} = 1$.

On ShapeNet-SRN we compare to single-view reconstruction deterministic works which report scores on this standard benchmark [9, 16, 21, 36, 37, 47, 53].

4.1. Single-view reconstruction on Minens

In table 1 we compare the reconstruction quality using PSNR and LPIPS. Sampling multiple plausible reconstructions via views diffusion improves the PSNR of the best sample in both 'Ambiguous' and 'Random' subsets. The significant gain in PSNR in the 'Ambiguous' dataset shows that diffusion can effectively single-view reconstruction ambiguities. Renders of samples of the reconstructed volumes in fig. 2 show how diverse poses and textures are sampled under the presence of ambiguity.

In table 1 it is also seen that using views diffusion leads

to a decrease in *average* LPIPS (lower is better), suggesting that *all* samples from our method are more perceptually plausible than the results from the baselines. Finally, the improvement in the metrics is further accompanied by qualitative comparison in figs. 2 and 6 where our samples are seen to be much sharper than the baseline results.

4.2. Single-view reconstruction: ShapeNet & CO3D

Quantitative results in ShapeNet and CO3D are given in tables 2 and 3. Like in Minens, in ShapeNet-SRN the best sample from Viewset Diffusion has higher PSNR than the baselines, which indicates that the model can sample from the correct distribution. This is further confirmed by Viewset Diffusion's lower (better) LPIPS than all baselines in CO3D (table 3) and almost all baselines in ShapeNet (table 2). VisionNeRF [21] outperforms our method in LPIPS, likely due to their use of the much stronger ViT-based 2D feature extraction (our 2D feature extractor is much smaller, consisting of only 5 convolutional layers). On challenging CO3D classes (Teddybear, Plant, Vase), the deterministic baselines achieve better PSNR, possibly because more than 100 samples are required to adequately sample the space of reconstructions for these more complex objects.

4.3. Unconditional generation

Viewset Diffusion supports unconditional generation by setting the number of clean input views to 0. To generate 3D prior works [26, 34, 46] require 3D ground truth at training time, while we only use 3 views per object[5] (at test time, we can generate any number of 3D consistent views). In fig. 7 we compare samples from our method (network

---

[5]For Minens dataset this is precisely true. In CO3D and ShapeNet different viewsets may come from the same objects due to data limitations, but they are treated independently by the training algorithm.

Method	Hydrant			Teddybear			Plant			Vase		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
RenderDiffusion	17.43	0.70	0.263	14.71	0.48	0.444	17.30	0.46	0.467	18.92	0.68	0.288
PixelNeRF	18.07	0.67	0.297	15.01	0.43	0.451	17.62	0.41	0.460	17.98	0.61	0.329
RenderDiffusion++	21.61	0.69	0.282	19.58	0.65	0.303	19.85	0.49	0.399	21.51	0.65	0.292
Ours w/o $\mathcal{D}$	22.06	0.78	0.217	**19.73**	0.65	**0.309**	**20.33**	0.51	0.382	**21.89**	0.68	0.264
Ours w $\mathcal{D}$ - best	**22.36**	**0.80**	(0.176)	19.68	**0.70**	(0.267)	20.23	**0.58**	(0.339)	21.36	**0.75**	(0.210)
Ours w $\mathcal{D}$ - mean	(20.52)	(0.77)	**0.199**	(17.28)	(0.64)	**0.309**	(19.47)	(0.40)	**0.366**	(20.05)	(0.71)	**0.237**

Table 3: Single view reconstruction - CO3D. Our method improves over baselines in CO3D classes on SSIM and LPIPS.
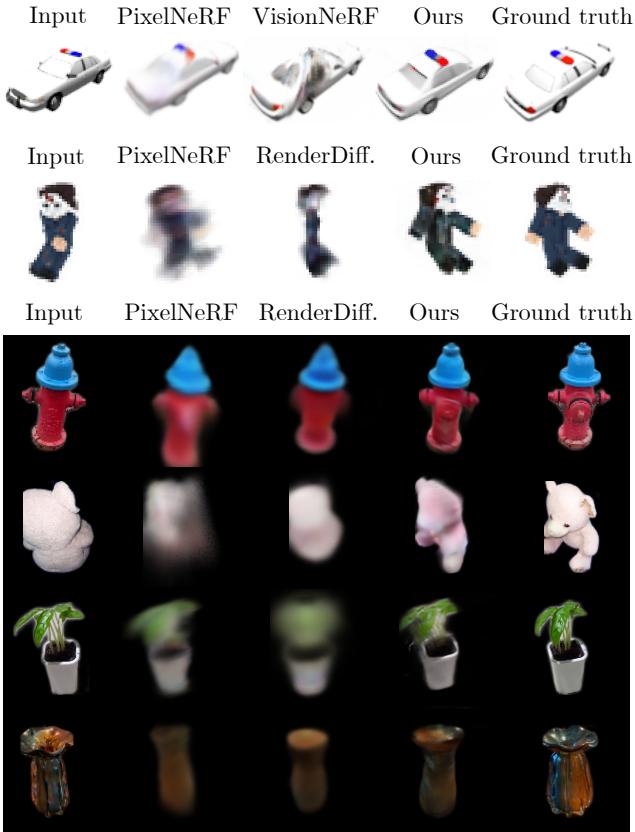

Figure 6: Single view reconstruction. Our method outputs sharper shapes than prior work. The solutions are ambiguous, therefore our samples do not match the ground truth exactly but are more plausible than deterministic baselines.

	PSNR ↑	LPIPS ↓
Full model	**20.36**	**0.075**
$\ominus$ diffusion $\mathcal{D}$	18.85	0.101
$\ominus$ attention in aggregation	19.54	0.100
$\ominus$ unprojection	18.26	0.164

Table 4: Ablations. Impact of removing component from our method on reconstruction quality.

4.4. Ablations

We assess the importance of different components of our method: input image unprojection, attention-based aggregation of features from different views, and using diffusion ($\mathcal{D}$). We use the 'Ambiguous' Minens dataset and evaluate best PSNR and average LPIPS in the unseen novel views. We train smaller models (half the number of U-Net convolutional layers and no self-attention layers) for fewer iterations (60k) due to the computational cost. Results are reported in table 4. Not using diffusion ($\sigma_t = \{0\}$, using notation from section 3.5) leads to a drop in PSNR and worse perceptual quality due to the reconstructions being blurry in the presence of ambiguity. Removing attention-based feature aggregation (section 3.4, 4.) across frames and aggregating them with a simple mean prohibits the network from reasoning about viewpoints and occlusion when pooling the features from different views. Finally, removing unprojection (section 3.4, 2.) hinders the learning process due to the removal of local conditioning which is known to improve the learning process [53].

5. Conclusions

We have presented Viewset Diffusion, a method to learn a model for probabilistic single-view 3D reconstruction and generation. By diffusing viewsets, we can learn a DDPM from multi-view 2D supervision and still learn to generate 3D objects, having only 3 views per object and no access to 3D ground truth. Viewset Diffusion also unifies 3D reconstruction and generation, and enables feed-forward
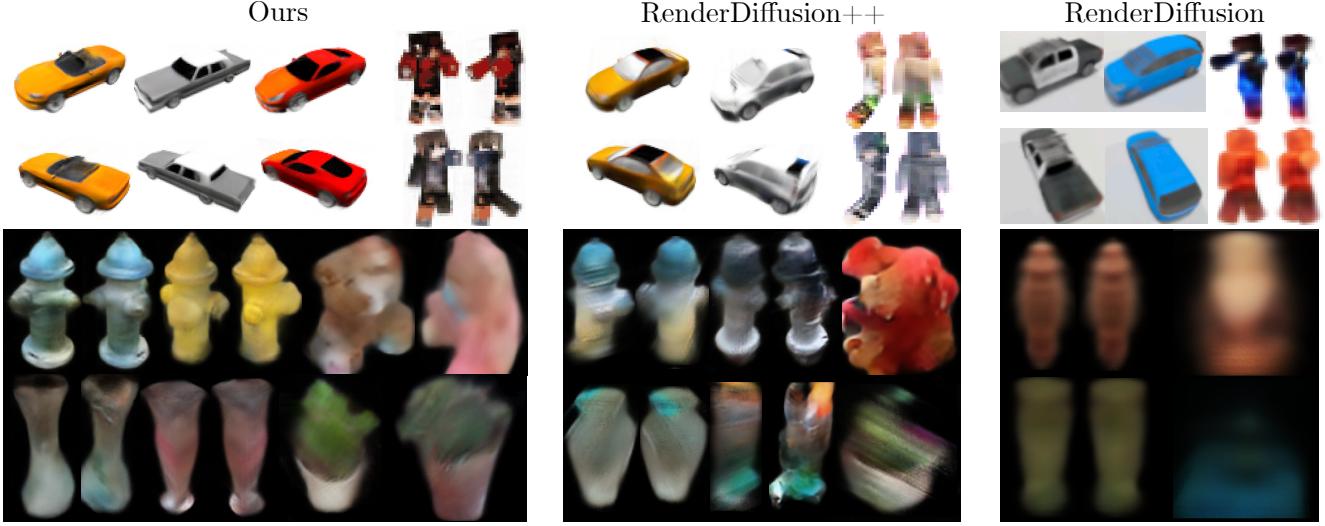
from section 3.4, $N_{\text{inf}} > 1$), RD++ (network from section 3.4, $N_{\text{inf}} = 1$) and RD (network from [1], $N_{\text{inf}} = 1$). Viewset Diffusion samples are sharper from all viewpoints, which demonstrates the advantage of diffusing more than one view jointly. Intuitively, in the last step of diffusion, Viewset Diffusion essentially performs reconstruction from $3 \leq N_{\text{inf}} \leq 5$ (nearly) clean views of the object, whereas RD and RD++ do so from a single view, which results in blurry images from other viewpoints.

Figure 7: Unconditional generation - Cars, Minens, Hydrants, Teddybears, Vases, Plants. Samples from our method show higher visual detail than RenderDiffusion [1] and our improvement over it, RenderDiffusion++.

probabilistic 3D reconstruction with diffusion models. We have shown empirically that a probabilistic approach to the single-view reconstruction problem leads to higher-quality results and less blurry solutions than deterministic alternatives.

Ethics. We use the ShapeNet and CO3D datasets in a manner compatible with their terms. The images used in this research do *not* contain personal information such as faces. For further details on ethics, data protection, and copyright please see `https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html`.

References

[1] Titas Anciukevicius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J. Mitra, and Paul Guerrero. RenderDiffusion: Image diffusion for 3D reconstruction, inpainting and generation. In *Proc. CVPR*, 2023. 3, 4, 7, 8, 9

[2] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *Proc. CVPR*, 2022. 2

[3] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In *arXiv*, 2023. 3

[4] 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*, 2016. 2

[5] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proc. CVPR*, 2017. 3

[6] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Proc. NeurIPS*, 2021. 2

[7] Vincent Dumoulin, Ethan Perez, Nathan Schucher Florian, Strub Harm de Vries, Aaron Courville, and Yoshua Bengio. Feature-wise transformations. In *Distill*, 2018. 5

[8] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In *Proc. ICML*, pages 11808–11826. PMLR, 2023. 3

[9] Pengsheng Guo, Miguel Angel Bautista, Alex Colburn, Liang Yang, Daniel Ulbricht, Joshua M. Susskind, and Qi Shan. Fast and explicit neural view synthesis. In *Proc. WACV*, 2022. 7

[10] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oguz. 3DGen: Triplane latent diffusion for textured mesh generation. *corr*, abs/2303.05371, 2023. 3, 4

[11] JunYoung Gwak, Christopher B Choy, Manmohan Chandraker, Animesh Garg, and Silvio Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In *Proc. 3DV*, 2017. 1, 3

[12] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *arXiv*, 2023. 3, 6

[13] 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 *Proc. CVPR*, 2021. 2

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Proc. NeurIPS*, 2020. 2, 3, 5

[15] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proc. ICCV*, pages 5885–5894, October 2021. 2

[16] Wonbong Jang and Lourdes Agapito. CodeNeRF: Disentangled neural radiance fields for object categories. In *Proc. ICCV*, 2021. 2, 7

[17] Heewoo Jun and Alex Nichol. Shape-E: Generating conditional 3d implicit functions. *arXiv*, 2023. 3

[18] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proc. ECCV*, 2018. 2

[19] Animesh Karnewar, Andrea Vedaldi, David Novotny, and Niloy Mitra. Holodiffusion: Training a 3D diffusion model using 2D images. In *Proc. CVPR*, 2023. 3

[20] Gang Li, Heliang Zheng, Chaoyue Wang, Chang Li, Changwen Zheng, and Dacheng Tao. 3ddesigner: Towards photorealistic 3d object generation and editing with text-guided diffusion models. *arXiv*, 2022. 3

[21] Kai-En Lin, Lin Yen-Chen, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. Vision transformer for nerf-based view synthesis from a single input image. In *Proc. WACV*, 2023. 2, 7

[22] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *arXiv*, 2023. 2, 3

[23] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proc. CVPR*, 2021. 3

[24] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Realfusion: 360° reconstruction of any object from a single image. In *Proc. CVPR*, 2023. 3

[25] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. ECCV*, 2020. 2

[26] Norman Müller, , Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proc. CVPR*, 2023. 3, 5, 7

[27] Norman Müller, Andrea Simonelli, Lorenzo Porzi, Samuel Rota Bulò, Matthias Nießner, and Peter Kontschieder. AutoRF: Learning 3D object radiance fields from single view observations. *CoRR*, abs/2204.03593, arXiv.cs. 2

[28] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proc. ICLR*, 2021. 5

[29] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *Proc. ICLR*, 2023. 3

[30] 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. ICCV*, 2021. 2, 6

[31] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. ShaRF: Shape-conditioned radiance fields from a single view. In *Proc. ICML*, 2021. 2

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. CVPR*, 2022. 3

[33] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *Proc. NeurIPS*, 2021. 2

[34] J. Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proc. CVPR*, 2023. 3, 4, 7

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 6

[36] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *Proc. NeurIPS*, 2021. 2, 7

[37] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Proc. NeurIPS*, 2019. 6, 7

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *Proc. ICLR*, 2021. 6

[39] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *Proc. ECCV*, 2022. 2

[40] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *Proc. CVPR*, 2023. 2

[41] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proc. CVPR*, 2022. 5

[42] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proc. CVPR*, 2017. 2

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 5

[44] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proc. CVPR*, 2023. 3

[45] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proc. CVPR*, 2021. 2

[46] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, and Baining Guo. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proc. CVPR*, 2023. 3, 4, 7

[47] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *Proc. ICLR*, 2023. 2, 3, 7

[48] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T. Freeman, and Joshua B. Tenenbaum. Learning shape priors for single-view 3D completion and reconstruction. In *Proc. ECCV*, 2018. 1, 3

[49] Shangzhe Wu, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. Dove: Learning deformable 3d objects by watching videos. *IJCV*, pages 1–12, 2023. 2

[50] Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. MagicPony: Learning articulated 3d animals in the wild. In *Proc. CVPR*, 2023. 2

[51] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. SinNeRF: Training neural radiance fields on complex scenes from a single image. In *Proc. ECCV*, 2022. 2

[52] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Proc. NeurIPS*, 2016. 2

[53] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *Proc. CVPR*, 2021. 2, 7, 8

[54] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. CVPR*, 2018. 6

[55] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *Proc. CVPR*, 2023. 3