HOLOFUSION: Towards Photo-realistic 3D Generative Modeling

Animesh Karnewar
UCL
a.karnewar@ucl.ac.uk

Niloy J. Mitra
UCL
n.mitra@ucl.ac.uk

Andrea Vedaldi
Meta AI
vedaldi@meta.com

David Novotny
Meta AI
dnovotny@meta.com

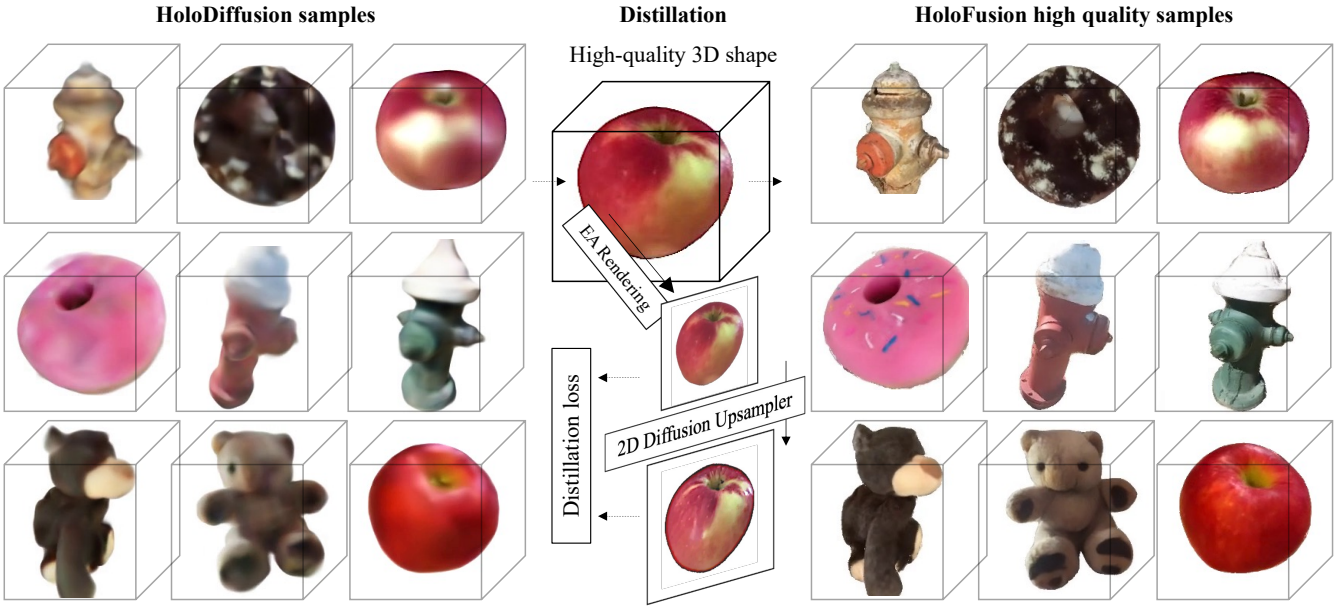

Figure 1: We propose HOLOFUSION to generate photo-realistic 3D radiance fields by extending the HoloDiffusion method with a jointly trained 2D 'super resolution' network. The independently super-resolved images are fused back into the 3D representation to improve the fidelity of the 3D model via distillation, while preserving the consistency across view changes.

Abstract

Diffusion-based image generators can now produce high-quality and diverse samples, but their success has yet to fully translate to 3D generation: existing diffusion methods can either generate low-resolution but 3D consistent outputs, or detailed 2D views of 3D objects but with potential structural defects and lacking view consistency or realism. We present HOLOFUSION*, a method that combines the best of these approaches to produce high-fidelity, plausible, and diverse 3D samples while learning from a collection of multi-view 2D images only. The method first generates coarse 3D samples using a variant of the recently proposed HoloDiffusion generator. Then, it independently renders and upsamples a large number of views of the coarse 3D model, super-resolves them to add detail, and distills those into a single, high-fidelity implicit 3D representation, which also ensures view-consistency of the final renders. The super-resolution network is trained as an integral part of* HOLOFUSION*, end-to-end, and the final distillation uses a new sampling scheme to capture the space of super-resolved signals. We compare our method against existing baselines, including DreamFusion, Get3D, EG3D, and HoloDiffusion, and achieve, to the best of our knowledge, the most realistic results on the challenging CO3Dv2 dataset.*

1. Introduction

Diffusion models [32, 7, 31] are at the basis of state-of-the-art 2D image generators which can now produce very high-quality and diverse outputs. However, their success has yet to be translated to 3D and there is no generator that can produce 3D assets of a comparable quality.

Recent attempts at extending diffusion to 3D generation have reported mixed success. Some authors have attempted to apply diffusion directly in 3D [18], or still in 2D but using a 3D-aware neural network [42, 1]. This requires solving two problems: first, finding a suitable 3D representation (*e.g.*, triplane features [4], mesh [19], voxels [18]) that scales well with resolution and is amenable to diffusion;

and, second, obtaining a large amount of 3D training data, for example using synthetic models [41, 27], or training the model using only 2D images [18], often via differentiable (volume) rendering [13, 26]. However, the quality of results so far is limited, especially when training on real images.

Other authors have proposed to *distill* 3D objects from pre-trained 2D image generators. For instance Score Distillation Sampling (SDS) [29] can sample 3D objects from a high-quality off-the-shelf 2D diffusion model while requiring no (re)training. However, without any 3D guidance, distillation methods often produce implausible results; for example, they suffer from the 'Janus effect', where details of the front of the object are replicated on its back. They also create overly-smooth outputs that average out inconsistencies arising from the fact that the signal obtained from the 2D model is analogous to sampling independent views of the object (see Sec. 4.2 for examples). Furthermore, distillation methods do not support unconditional sampling, even if the underlying image generator does, as strong language guidance is required to stabilise the 3D reconstruction.

In this work, we propose HOLOFUSION, a method that combines the best of both approaches. We start from HoloDiffusion [18], a diffusion-based 3D generator. This model can be trained using only a multiview image dataset like [30] and produces outputs that are 3D consistent. However, the output resolution is limited by computation and memory. We augment the base model with a lightweight super-resolution network that upscales the initial renders. Crucially, the 2D super-resolution model is integrated and trained jointly with the 3D generator, end-to-end.

The super-resolution network outputs detailed views of the 3D object, and the underlying 3D generator ensures that the coarse structure of these views is indeed consistent (*e.g.*, avoiding the Janus effect and other structural artifacts). However, the 2D upscaling still progresses independently for different views, which means that fine grained details may still be inconsistent between views. We address this issue by distilling a single, coherent, high quality 3D model of the object from the output of the upsampler. For this, we propose a new distillation technique that efficiently combines several putative super-resolved views of the object into a single, coherent 3D reconstruction.

With this, we are able to train a high-quality 3D generator model purely from real 2D data. This model is capable of generating consistent and detailed 3D objects, which in turn result in view-consistent renderings (see Fig. 1) at a quality not achievable by prior methods.

We evaluate HOLOFUSION on real images (CO3Dv2 dataset [30]) and compare with a variety of competing alternatives (*e.g.*, HoloDiffusion [18], Get3D [9], EG3D [4], DreamFusion [40]) demonstrating that view-consistent high-quality 3D generation is possible using our simple, effective, easy-to-implement hybrid approach.

2. Related Work

3D generators that use adversarial learning. Generative Adversarial Learning (GAN) [10] learns a generator network so that its "fake" samples cannot be distinguished from real images by a second discriminator network. Approaches such as PlatonicGAN [13], HoloGAN [28], and PrGAN [8] introduced 3D structure into the generator network, achieving 3D shape generation with only image-level supervision. Our method is related to those as it renders images from a generated voxel grid, as well as to HoloGAN [28], which renders features and then converts them into an image by a lightweight 2D convolutional network. Other voxel-based 3D generators include VoxGRAF [34] and NeuralVolumes [21].

More recently, 3D generators have built on neural radiance fields [26]. GRAF [33] was the first to adopt the NeRF framework; analogous to PlatonicGAN, they generate the parameters of an MLP which renders realistic images of the object from a random viewpoint. This idea has been improved in StyleNeRF [11] and EG3D [4] by adding a 2D convolutional post-processing step after emission-absorption rendering, which is analogous to our super-resolution network. EG3D also introduced a novel 'triplane' representation of the radiance field which, in a memory efficient manner, factorises the latter into a triplet of 2D feature planes. EG3D inspired several improvements such as GAUDI [2] and EpiGRAF [37].

Mesh-based 3D generators have been explored in [43]. Recently, GET3D [9] replaced the radiance field with a signed distance function to regularise the representation of geometry. The latter is converted into a mesh and rendered in a differentiable manner by using the marching tetrahedral representation [35].

Modeling 3D with diffusion. Diffusion methods [38] have recently became the go-to framework for generative modeling of any kind, including 3D generative modeling. The first applications of diffusion to 3D considered point-cloud generators trained on synthetic data [23, 45, 44].

3D distillation of 2D diffusion models. More recently, DreamFusion [29] ported the idea of distillation to diffusion models: they extract a neural radiance field so that its renders match the belief of a pre-trained 2D diffusion generator [32, 7, 31]. They introduce the Score Distillation Sampling (SDS) loss which makes distillation relatively efficient (but still in the order of several minutes for a single 3D sample). Their generation can be conditioned by an image or by a textual description, making the process rather flexible. Magic3D [19] further increases the quality of the output by distilling a mesh-based 3D representation instead of a radiance field.

Image-conditioned 3D diffusion. The idea of distillation has been applied to few-view conditioned reconstruction in

[42, 12, 25, 46, 6]. SparseFusion [46] employs a 3D-based new-view synthesis model [39] followed by a 2D diffusion upsampler. They complete the process by 3D distillation, ensuring that the generated views of the object are consistent. NeRFDiff [12] and 3DiM [42] bypass an explicit 3D model and directly generate new views of an object using a 2D image generator and, in the case of NeRFDiff, refine the results using distillation.

While SparseFusion and NeRFDiff need to be trained on a dataset of object-centric multi-view images with pose information, RealFusion [25] and NeRDi [6] can be used for zero-shot monocular 3D reconstruction, starting from a pre-trained 2D diffusion model. Given a single image as input, they automatically generate a prompt for the diffusion model, using a form of prompt inversion, and then use distillation to extract a radiance field.

Unconditional generation. Most relevantly to us, unconditional generation, *i.e.*, generation which does not require either text or image conditioning, was explored in [41, 36, 27, 18]. While [27, 36, 41] train generators given synthetic 3D ground truth, similar to us, HoloDiffusion [18] is supervised only with real object-centric images and camera poses. While HoloDiffusion was the first to demonstrate successful training on real image data, its renders contain considerably lower amount of detail than samples from a conventional 2D image generator that uses diffusion. We thus leverage a 2D diffusion upsampler, conditioned on the lower-fidelity HoloDiffusion renders, to distill higher resolution images and, eventually, 3D models.

3. HoloFusion

We present HoloFusion, a method that can learn a high-quality diffusion-based 3D generator from a collection of multiview 2D images. HoloFusion first obtains an unconditional low-resolution 3D sample using diffusion and then distills a high-resolution 3D radiance field representing a higher-quality version of the generated object. We first summarize the Denoising Diffusion Probabilistic Models (DDPMs) [16] that we utilize in Sec. 3.1. Then, we discuss the low-resolution 3D generator in Sec. 3.2 followed by super-resolution distillation in Sec. 3.3.

3.1. Preliminaries: DDPMs

Let $x = x_0$ be a random vector whose probability distribution $p(x|y)$ we seek to model. The DDPM [16] defines a hierarchy of latent variables x_t, $t = 0, \dots, T$ and an encoder q comprising a sequence of Gaussian distributions

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)\mathbb{I}), \qquad (1)$$

where $\alpha_t, \dots, \alpha_T$ is a predefined 'noising schedule'. Given knowledge of x_0, a sample x_t can be drawn in a closed-form directly from $q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbb{I})$, where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Hence, we can express x_t as:

$$x_t = \hat{\epsilon}_t(x_0) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t \qquad (2)$$

The noising schedule is chosen such that $\bar{\alpha}_T \approx 0$. In this manner, $q(x_T|x_0) \approx \mathcal{N}(x_T; 0, \mathbb{I})$ is approximately normal, and so is $q(x_T)$. To generate a sample $x = x_0$, we start by sampling x_T from this normal distribution and then sample the intermediate latent variables backward. This is done by using a variational approximation of the probabilities $q(x_{t-1}|x_t)$ given by the Gaussian factors:

$$p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}}D_\theta(x_t, t), (1-\bar{\alpha}_{t-1})\mathbb{I}) \quad (3)$$

where D_θ is a neural network with parameters θ.

The network D_θ is trained by maximizing the ELBO (Evidence Lower Bound), which reduces to the denoising objective [22]:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\epsilon,x}\left[\frac{\bar{\alpha}_{t-1}}{2(1-\bar{\alpha}_{t-1})^2}\|D_\theta(\hat{\epsilon}_t(x), t) - x\|^2\right], \quad (4)$$

where t is sampled uniformly from $U[1, T]$. Hence, $D_\theta(x_t, t)$ approximates the clean sample x_0 given the noisy sample x_t (obtained using (2)).

3.2. HoloDiffusion revisited

Given a large dataset of 3D models, the framework of Sec. 3.1 could be used to train a corresponding probability distribution. However, such a dataset is not available, and we must instead learn from 2D images of physical 3D objects. Given a dataset containing several views of a large number of objects, we could use image-based reconstruction (*e.g.*, using neural rendering) to obtain corresponding 3D models first, and then use those to train a diffusion model. Instead, we adopt, and slightly upgrade, the HoloDiffusion method [18], which learns a 3D diffusion model *directly* from the 2D images.

Training data. HoloDiffusion learns from a collection $\mathcal{D}$ of N image sequences $s_i = (I_j^i, C_j^i)_{j=1}^{N_{\text{frame}}}$, $i = 1, \dots, N$, where frame $I_j^i \in \mathbb{R}^{3\times H\times W}$ is an RGB image and $C_j^i \in \mathbb{R}^{4\times 4}$ is the corresponding camera projection matrix, collectively defining the motion of the camera.

3D representation and rendering. The shape and appearance of the object are represented by a voxel grid $V \in \mathbb{R}^{d\times S\times S\times S}$ with resolution S containing a d-dimensional feature vector per voxel. Given a 3D point $\mathbf{p} \in \mathbb{R}^3$, its opacity $\sigma(\mathbf{p}) \in \mathbb{R}_+$ and color $c(\mathbf{p}) \in \mathbb{R}^3_{[0,1]}$ are obtained from the voxel grid by an MLP $M_\eta(V(\mathbf{p}))$ that takes as input the d-dimensional feature vector $V(\mathbf{p})$ extracted form the grid via trilinear interpolation [20]. The usual emission-absorption model [26, 24] is then used to implement a differentiable rendering function R_η, mapping the voxel grid V and the camera viewpoint C into an image $\hat{I} = R_\eta(V, C)$, where η are the parameters of the MLP.

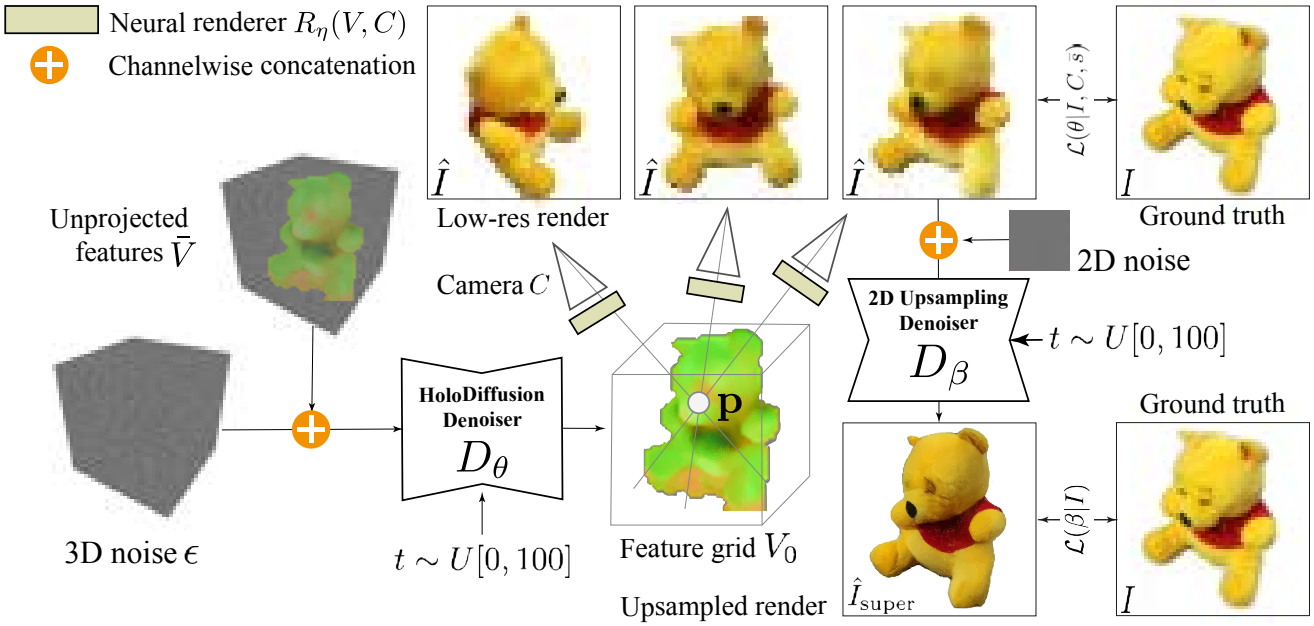

Figure 2: **Overview.** HOLOFUSION, which trains the 3D denoiser network D_θ, is augmented with the 2D 'super-resolution' diffusion model D_β. Both models are trained end-to-end by supervising their outputs with 2D photometric error.

Training scheme. HoloDiffusion leverages the DDPM framework (revised in the previous paragraphs) to recover the density $p(V)$ over voxel grids $x = V$ encoding plausible real-life objects. In order to train a DDPM on such 3D data, we would need access to ground-truth 3D models V, which are not available. HoloDiffusion addresses this problem by making three changes to DDPM.

First, it replaces the data denoising loss with a photometric reconstruction loss. Given a pair $(I, C) \in s$ from one of the training sequences s, it replaces Eq. (4) with $\mathbb{E}_{t,\epsilon,C}\left[\|I - R_\eta(D_\theta(\hat{\epsilon}_t(V), t), C)\|^2\right]$ where the goal is not to reconstruct the 'clean' volume V (which is unknown), but rather its image I (which is known).

Second, also because the 'clean' volume V is not available, we cannot use Eq. (3) to generate the noisy volumes V_t to denoise; the only exception is the last sample V_T, which is pure noise. This suggests to adopt a 'double denoising' step. First, pure noise V_T is fed into the denoiser to obtain an (approximate) version of $V_0 = D_\theta(V_T, T)$ of the clean volume $V_0 = V$. Then, noise is applied to obtain $V_t = \hat{\epsilon}_t(V_0) = \sqrt{\bar{\alpha}_T} V_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_t$ according to Eq. (3), and the latter is fed back into the denoiser as above.

Finally, there is the issue that unconditional generation of the clean volume V_0 from pure noise V_T is difficult, especially in the first iterations of training. On the other hand, the problem of *view-conditioned* generation is considerably easier. Hence, the third idea is to learn a *conditional* generator, using a variable number of input views. Specifically, given a training sequence s, the method extracts a random subset of frames $\bar{s} \subset s$ (which could be empty, which corresponds to unconditional generation). Then, a feature volume $\bar{V} = \Phi(\bar{s}) \in \mathbb{R}^{d \times S \times S \times S}$ is obtained from the selected frames. This extracts 2D image features using a pre-trained and frozen 2D image encoder and then pools them in 3D via 'unprojection' [17, 14] into $\bar{V}$, where $\bar{V} = 0$ if $\bar{s}$ is empty. Finally, these pooled features are used to condition the denoising network $V_0 = D_\theta(V_T, \bar{V}, T)$, which, on average, leads to a simpler reconstruction problem.

Putting it all together, the training loss becomes:

$$\mathcal{L}(\theta|I, C, \bar{s}) = \mathbb{E}_{t,\hat{\epsilon},V_T}\left[\|\hat{I} - I\|^2\right], \tag{5}$$

$$\text{where } \hat{I} = R_\eta(D_\theta(V_t, \bar{V}, t), C), \tag{6}$$

$$V_t = \hat{\epsilon}_t(V_0),$$

$$V_0 = D_\theta(V_T, \bar{V}, T),$$

$$\bar{V} = \Phi(\bar{s}).$$

This loss is averaged over training sequences s, subsequences $\bar{s} \subset s$, and views $(I, C) \in s$ therein. Note that this is slightly different than the original HoloDiffusion, where feature volume $\bar{V}$ and reconstructed volumes V_t overlap as arguments of the denoiser; we found that keeping them separated in the formulation leads to more stable training and additionally allows for view-conditioned generation.

3.3. HOLOFUSION

The method of Sec. 3.2 learns to generate 3D objects from 2D image supervision only, but the fidelity of the output is limited by the resolution at which the operations are carried out. Increasing resolution is difficult due to the GPU

memory impact of the voxel-based representation, so we seek a more efficient way to do so. The idea is to incorporate a 2D super-resolution network (Sec. 3.3.1), trained end-to-end, that improves the output from the base model. The super-resolved images are eventually fused back in an improved 3D model, which also has the benefit of further increasing view consistency (Sec. 3.3.2).

3.3.1 Integrating super-resolution

As shown in Fig. 2, we augment the method of Sec. 3.2 with a lightweight refinement post-processor network that takes the 2D image $\hat{I}$ generated by the base model and outputs a higher quality version $\hat{I}_{\text{super}}$ of the same. This can be thought of as a form of super-resolution; however, due to the particular statistics of the input ('low-res') images $\hat{I}$ that HoloDiffusion generates, it is necessary to train this super-resolution network in an end-to-end fashion with HoloDiffusion, integrating the two models.

To make this integration seamless, we formulate super-resolution as another diffusion process that runs 'in parallel' with 3D reconstruction. Hence, the super-resolved image $\hat{I}_{\text{super}} = D_\beta(I_t, \hat{I}, t)$ is the output of a denoiser network (a lightweight U-Net), which takes as input the noised target image $I_t = \hat{\epsilon}_t(I)$ and is also conditioned on the 'low-res' output $\hat{I} = R_\eta(V, C)$ of HoloDiffusion from Eq. (6). This denoiser is trained with the DDPM loss:

$$\mathcal{L}(\beta|I) = \mathbb{E}_{t,\hat{\epsilon}}\left[\|D_\beta(\hat{\epsilon}_t(I), \hat{I}, t) - I\|^2\right]. \tag{7}$$

Training details The overall model (D_β and D_θ) is trained end-to-end by optimising the loss $\mathcal{L}(\theta|I, C, \bar{s}) + \mathcal{L}(\beta|I)$ obtained by summing Eqs. (5) and (7).

As training data, we use a large dataset of images capturing object-centric scenes ([30]). In each training batch, we pick a random training scene s and sample 15 different source images $\bar{s}_{\text{src}} \subset s$ which are unprojected to generate the feature volume conditioning $\bar{V}$. Then, $\bar{V}$ is rendered into 4 random target views $\bar{s}_{\text{tgt}} \subset (s \setminus \bar{s}_{\text{src}})$ which allows to optimize the training image reconstruction loss $\mathcal{L}(\theta|I, C, \bar{s}) + \mathcal{L}(\beta|I)$. The latter uses the Adam optimizer with an initial learning rate of $5 \cdot 10^{-5}$ decaying tenfold whenever the loss plateaus until convergence.

3.3.2 Fusing super-resolved views in 3D

The method of Sec. 3.3.1 leaves us with high-resolution views $\hat{I}_{\text{super}}$ of the generated 3D object. However, we would like to obtain a single, high-quality 3D model, not just individual views of it. In this section, we discuss how to take the super-resolved images and fuse them into such a model, while addressing the issues that these images are not perfectly view-consistent.

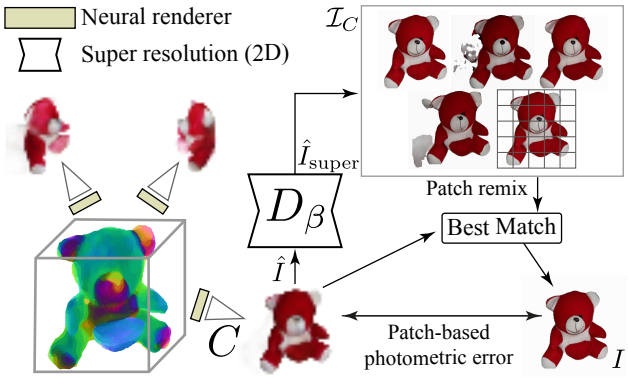

Figure 3: **Distillation**. HOLOFUSION distills a single high-resolution voxel grid V_0^H by minimizing a top-k patch-remix loss $\mathcal{L}_{\text{distil}}$ between the grid renders $R_{\eta'}(V_0^H, C)$ and a bank $\mathcal{I}_C$ of $K = 5$ high-res images output by the 2D diffusion upsampler D_β for each scene camera C.

The basic idea is simple. We can generate a certain number (*e.g.*, 100) high-resolution images of the object from different viewpoints C and then use a technique, akin to neural rendering, to fuse them back into a single 3D model. However, there is a problem with this idea: The model of Sec. 3.3.1 generates high-quality views I_{super}, but these are *view-dependent samples* from the distribution $p(\hat{I}_{\text{super}}|\hat{I})$ where $\hat{I} = R_\eta(V, C)$ is the 'low-res' output form HoloDiffusion. Because super-resolving details is intrinsically ambiguous, there is no reason why samples I_{super} taken from different viewpoints C would be consistent (Fig. 7). Fusing them into a single 3D model would then result in a blurry appearance yet again.

As described in Fig. 3, we address this issue in a principled manner by considering *several* possible super-resolved images $\mathcal{I}_C = \{I_{\text{super}} \sim p(\hat{I}_{\text{super}}|\hat{I})\}$ sampled from each given viewpoint C. Then, we optimize a high-resolution voxel grid V_0^H by minimizing the photometric loss:

$$\mathcal{L}_{\text{distil}}(\eta', V_0^H|\mathcal{I}_C) = \mathbb{E}_C\left[\min_{I_{\text{super}} \in \mathcal{I}_C} \|I_{\text{super}} - R_{\eta'}(V_0^H, C)\|^2\right] \tag{8}$$

where $R_{\eta'}(V_0^H, C)$ is the render of a high-resolution voxel grid $V_0^H \in \mathbb{R}^{d \times S' \times S' \times S'}$, $S' > S$ using the learnable renderer $R_{\eta'}$ with scene specific parameters η'. Minimizing with respect to I_{super} means that the 3D model must be consistent with at least *one* of the possible super-resolved images, drawn from the distribution of super-resolved samples, for each view C.

Patch remix. In practice, this approach requires a very large number of super resolved images $\mathcal{I}_C$ to be effective. We found that we can significantly improve the statistical efficiency by performing the minimization at the level of individual patches. Namely, we produce a stack of only

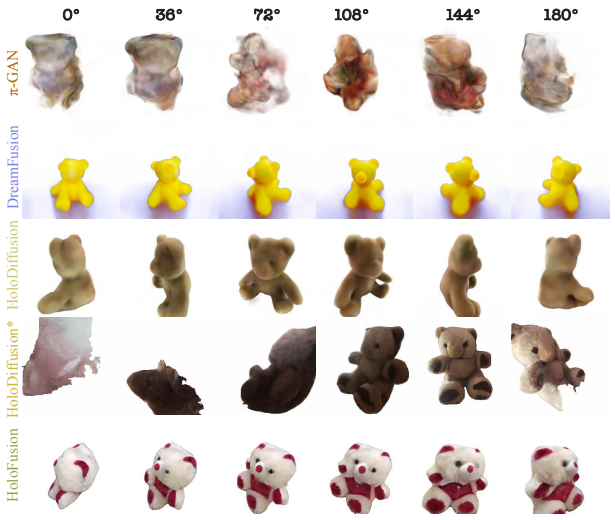

Figure 4: **Generated 3D samples visualized from a moving camera.** π-GAN and HoloDiffusion* fail to produce 3D view consistent samples, while DreamFusion suffers from the "Janus" problem (multiple heads).

$K = |\mathcal{I}_C| = 5$ super resolved images and perform the minimization in Eq. (8) at the level of small 16×16 patches independently (effectively allowing super-resolved images to 'remix' as needed to fit the generated view $R_{\eta'}(V_0^H, C)$).

Distillation details. $\mathcal{L}_{\text{distil}}$ is optimized independently for each generated scene with Adam (lr=$2 \cdot 10^{-4}$) for 25K steps until convergence. While η' is initialized using the pre-trained multi-sequence weights η, V_0^H is initialized by trilinearly upsampling the low-resolution volume V_0 output by HoloDiffusion. Cameras C are sampled at uniform azimuths with elevation fixed at object's equator.

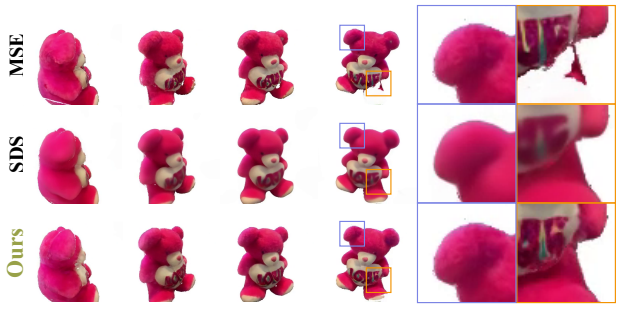

Figure 5: **Fusing views.** Our patch-remix (Sec. 3.3.2) compared to the **SDS** and **MSE** distillation. **MSE** has "floaters" and viewpoint inconsistencies, **SDS** over-smooths the texture. Ours is robust and produces superior quality.

4. Experiments

We begin with a description of the experiments conducted in Sec. 4.1, followed by an analysis and discussion of the results in Sec. 4.2.

4.1. Details

Dataset. We experiment on the challenging large-scale Co3Dv2 [30] dataset which is a popular choice for a real-world 3D reconstruction benchmark. More specifically, 4 categories are selected, `Apple`, `Hydrant`, `TeddyBear`, and `Donut`, with 500 3D-scenes per category for training. Each 3D scene contains $\sim$ 200 images of the object of interest along with poses of their corresponding cameras.

Baselines. We use two sets of baselines for comparison (Tab. 1): (i) general 3D generative modeling baselines and (ii) diffusion distillation based baselines. π-GAN [5], EG3D [4], GET3D [9], and HoloDiffusion [18] are considered as the 3D generative baselines. Along with HoloDiffusion, we also test the super-resolution integrated model (described in Sec. 3.3.1) HoloDiffusion*. For the distillation-based baselines, we consider the open-source implementation of DreamFusion [29] titled Stable-DreamFusion [40]. For the latter, scenes are generated by conditioning on prompts comprising names of Co3Dv2 categories extended with color and style modifier phrases leading to $\sim$ 200 prompts / 3D shapes per class. More details regarding the prompt creation are in the supplementary.

Metrics. We use FID [15] and KID [3] to compare the quality of our 2D renders, as these are commonly used to assess 2D and 3D generators.

4.2. Quantitative and qualitative analysis

Tab. 1 evaluates quantitatively while Fig. 6 qualitatively. Furthermore, Fig. 4 compares rendering view-consistency.

HoloFusion (Ours) yields better FID/KID scores than the general 3D generative baselines except for π-GAN on `Apple` and `Donut` classes. However, since π-GAN does not guarantee view consistency by design, it essentially acts as a 2D image GAN, and thus does better on the 2D FID/KID metrics, but it generates significantly view-inconsistent renders (see Fig. 4 and the supplementary).

We observed that the other 3D-GAN baselines, EG3D and GET3D, are prone to collapsing to a single adversarial sample leading to poor FID/KID scores. The latter is probably due to the 3D misalignment of the CO3Dv2 sequences across instances, which makes training harder.

HoloFusion also outperforms the text-to-3D Stable-DreamFusion on both FID/KID. Stable-DreamFusion yields good shapes, but produces synthetic-looking and overly-smooth textures and thus performs poorly when compared to the real-world images of Co3Dv2. As evi-

Figure 6: 3D samples generated by our HoloFusion compared to π-GAN, EG3D, GET3D, HoloDiffusion, HoloDiffusion*, and the text-to-3D Stable-DreamFusion.

Table 1: FID (↓) and KID (↓) on 4 classes of Co3Dv2 [30]. We compare with 3D generative modeling baselines (rows 1–5); with an SDS distillation-based Stable-DreamFusion (row 6); and with ablations of our HoloFusion (rows 7–8). The column "VP" denotes whether renders of a method are 3D view-consistent or not.

method	VP	Apple		Hydrant		TeddyBear		Donut		Mean	
		FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓
π-GAN [5]	✗	49.3	0.042	92.1	0.080	125.8	0.118	99.4	0.069	91.7	0.077
EG3D [4]	✓	170.5	0.203	229.5	0.253	236.1	0.239	222.3	0.237	214.6	0.233
GET3D [9]	✓	179.1	0.190	303.3	0.380	244.5	0.280	209.9	0.230	234.2	0.270
HoloDiffusion [18]	✓	94.5	0.095	100.5	0.079	109.2	0.106	115.4	0.085	122.5	0.102
HoloDiffusion*	✗	55.9	0.045	62.6	0.045	116.6	0.101	99.6	0.079	83.7	0.068
Stable-DreamFusion [40]	✓	139.0	0.104	185.2	0.132	183.4	0.125	169.3	0.114	169.2	0.119
HoloFusion (MSE)	✗	72.7	0.067	62.2	0.045	87.2	0.076	109.0	0.099	82.8	0.072
HoloFusion (SDS)	✓	123.0	0.105	77.1	0.058	117.8	0.090	142.8	0.087	115.2	0.085
HoloFusion (Ours)	✓	69.2	0.063	66.8	0.047	87.6	0.075	109.7	0.098	83.3	0.071

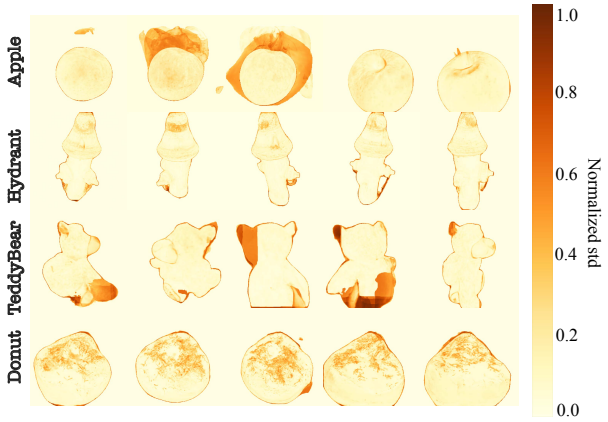

Figure 7: Heatmaps illustrating the per-pixel color variance of $K = 10$ hypothesis produced by the upsampler D_β. Some samples contain artifacts around the object boundaries which correspond to the high-variance regions in the figure. Our top-K patch-remix increases robustness by allowing the loss to discard such artifacts during distillation.

dent from the TeddyBear samples, the method also suffers from the "Janus" issue.

Compared to HoloDiffusion, we improve the FID/KID scores by a significant margin, mainly due to the more photo-realistic renders that include high-frequency details. Although the 2D Diffusion upsampler of HoloDiffusion* produces renders with the highest amount of details yielding scores similar to ours, they are not 3D view-consistent (as apparent from Fig. 4 and as explained in Sec. 3.3).

Ablations. In Tab. 1 and in Fig. 5 we ablate components of our HoloFusion to verify their contribution.

The first variant, HoloFusion (SDS), replaces the Top-k patch-remixed distillation loss with the score distillation sampling (SDS) gradient as proposed in [29]. As apparent from Fig. 5 and from the lower scores, SDS washes out all the high-frequency details in the textures.

Secondly, HoloFusion (MSE) reduces the number of upsampling hypotheses $\mathcal{I}$ to the minimum of $|\mathcal{I}| = 1$. Even though this slightly improves the 2D metrics, as can be seen from Fig. 5, the samples lack view-consistency and introduce "floaters". In Fig. 7 we further illustrate the variability of the upsampling hypotheses.

5. Conclusion

We have presented a hybrid diffusion-based method that can generate high-quality 3D neural radiance fields of real-life object categories. Our method starts by producing coarse 3D models whose renders are independently super-resolved, and finally consolidated using a robust distillation process. We evaluated our method on the Co3D v2 dataset and presented 3D-consistent, diverse, and high-quality results superior to all competing baselines.

Our method suffers from limitations that can be addressed in future work. First, our method is slow to sample from as the sampling process takes about 30 mins for each generation, because it is still a distillation-based method. An interesting extension would be to train another network to directly distill a set of super-resolved images, without requiring explicit optimization during inference. Second, we do not produce an explicit surface representation (*e.g.*, a mesh), which could be done by integrating a differentiable mesh render in the loop as done in some prior work.

6. Acknowledgements

Animesh and Niloy were partially funded by the European Union's Horizon 2020 research and innovation pro-

gramme under the Marie Skłodowska-Curie grant agreement No. 956585. This research has been partly supported by MetaAI and the UCL AI Centre. Finally Animesh is grateful to The Rabin Ezra Scholarship Fund being a recipient of their esteemed fellowship for the year 2023.

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. *arXiv.cs*, abs/2211.09869, 2022. 1

[2] Miguel Ángel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, Afshin Dehghan, and Josh M. Susskind. GAUDI: A neural architect for immersive 3d scene generation. *arXiv.cs*, abs/2207.13751, 2022. 2

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 6

[4] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 1, 2, 6, 8

[5] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021. 6, 8

[6] Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. *arXiv preprint arXiv:2212.03267*, 2022. 3

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1, 2

[8] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3D shape induction from 2D views of multiple objects. In *arXiv*, 2016. 2

[9] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*, 2022. 2, 6, 8

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[11] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3d-aware generator for high-resolution image synthesis. *CoRR*, abs/2110.08985, 2021. 2

[12] Jiatao Gu, Alex Trevithick, Kai-En Lin, Josh Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. *arXiv preprint arXiv:2302.10109*, 2023. 3

[13] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9984–9993, 2019. 2

[14] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3

[17] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. *Advances in neural information processing systems*, 30, 2017. 4

[18] Animesh Karnewar, Andrea Vedaldi, David Novotny, and Niloy Mitra. Holodiffusion: Training a 3D diffusion model using 2D images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023. 1, 2, 3, 6, 8

[19] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3d content creation. *arXiv.cs*, abs/2211.10440, 2022. 1, 2

[20] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 3

[21] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 2

[22] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022. 3

[23] 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*, pages 2837–2845, 2021. 2

[24] Nelson L. Max. Optical models for direct volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 1(2):99–108, 1995. 3

[25] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Realfusion: 360° reconstruction of any object from a single image. *arXiv e-prints*, pages arXiv–2302, 2023. 3

[26] 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, 3

[27] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. DiffRF: Rendering-guided 3d radiance field diffusion. In *Proc. CVPR*, 2022. 2, 3

[28] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3D representations from natural images. *arXiv.cs*, abs/1904.01326, 2019. 2

[29] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 6, 8

[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. CVPR*, 2021. 2, 5, 6, 8

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1, 2

[32] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1, 2

[33] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: generative radiance fields for 3D-aware image synthesis. *arXiv.cs*, abs/2007.02442, 2020. 2

[34] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. *ARXIV*, 2022. 2

[35] 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

[36] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. *arXiv preprint arXiv:2211.16677*, 2022. 3

[37] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. EpiGRAF: Rethinking training of 3d gans. *CoRR*, abs/2206.10535, 2022. 2

[38] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2

[39] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 156–174. Springer, 2022. 3

[40] Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. https://github.com/ashawkey/stable-dreamfusion. 2, 6, 8

[41] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. *arXiv preprint arXiv:2212.06135*, 2022. 2, 3

[42] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv.cs*, abs/2210.04628, 2022. 1, 3

[43] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2020. 2

[44] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022. 2

[45] 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*, pages 5826–5835, 2021. 2

[46] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. *arXiv preprint arXiv:2212.00792*, 2022. 3