

RealFusion

360° Reconstruction of Any Object from a Single Image

Luke Melas-Kyriazi Iro Laina Christian Rupprecht Andrea Vedaldi

Visual Geometry Group, Department of Engineering Science, University of Oxford

{luke, iro, chrisr, vedaldi}@robots.ox.ac.uk

<https://lukemelas.github.io/realfusion>



Figure 1. **RealFusion** generates a full 360° reconstruction of any object given a *single image* of it (left column). It does so by leveraging an existing diffusion-based 2D image generator. From the given image, it synthesizes a prompt that causes the diffusion model to “dream up” other views of the object. It then extracts a neural radiance field from the original image and the diffusion model-based prior, thereby reconstructing the object in full. Both appearance and geometry are reconstructed faithfully and extrapolated in a plausible manner (see the textured and shaded reconstructions from different viewpoints).

Abstract

We consider the problem of reconstructing a full 360° photographic model of an object from a single image of it. We do so by fitting a neural radiance field to the image, but find this problem to be severely ill-posed. We thus take an off-the-self conditional image generator based on diffusion and engineer a prompt that encourages it to “dream up” novel views of the object. Using the recent DreamFusion method, we fuse the given input view, the conditional prior, and other regularizers in a final, consistent reconstruction. We demonstrate state-of-the-art reconstruction results on benchmark images when compared to prior methods for monocular 3D reconstruction of objects. Qualitatively, our reconstructions provide a faithful match of the input view and a plausible extrapolation of its appearance and 3D shape, including to the side of the object not visible

in the image.

1. Introduction

We consider the problem of obtaining a 360° photographic reconstruction of *any* object given a *single image* of it. The challenge is that a single image *does not* contain sufficient information for 3D reconstruction. Without access to multiple views, an image only provides weak evidence about the 3D shape of the object, and only for one side of it. Even so, there is proof that this task *can* be solved: any skilled 3D artist can take a picture of almost any object and, given sufficient time and effort, create a plausible 3D model of it. The artist can do so by tapping into her vast knowledge of the natural world and of the objects it contains, making up for the information missing in the image.

To solve this problem algorithmically, one must then marry visual geometry with a powerful statistical model of the 3D world. The recent explosion of 2D image generators like DALL-E [36], Imagen [42], and Stable Diffusion [40] suggests that such models might not be far behind. By using diffusion, these methods can solve highly-ambiguous generation tasks, obtaining plausible 2D images from textual descriptions, semantic maps, partially-complete images, or simply unconditionally from random noise. Clearly, these models possess high-quality priors—if not of the 3D world, then at least of the way it is represented in 2D images. Hence, in theory, a 3D diffusion model trained on vast quantities of 3D data should be capable of producing 3D reconstructions, either unconditionally or conditioned on a 2D image. However, training such a model is infeasible because, while one can access billions of 2D images [43], the same cannot be said about 3D data.

The alternative to training a 3D diffusion model is to extract 3D information from an existing 2D model. A 2D image generator can in fact be used to sample or validate multiple views of a given object; these multiple views can then be used to perform 3D reconstruction. With early GAN-based generators, authors showed some success for simple data like faces and synthetic objects [3, 9, 12, 30, 31, 54]. With the availability of large-scale models like CLIP [34] and, more recently, diffusion models, increasingly complex results have been obtained. The most recent example is DreamFusion [33], which generates high-quality 3D models from textual descriptions alone.

Despite these advances, the problem of single-image 3D reconstruction remains largely unsolved. In fact, these recent methods do not solve this problem. They either sample random objects, or, like in the case of DreamFusion, start from a textual description.

A problem in extending generators to reconstruction is *coverage* (sometimes known as mode collapse). For example, high-quality face generators based on GANs are usually difficult to invert: they may be able to generate *many* different high-quality images, and yet are usually unable to generate *most* images [1]. Conditioning on an image provides a much more detailed and nuanced specification of the object than, say, a textual description. It is not obvious if the generator model would be able to satisfy all such constraints.

In this paper, we study this problem in the context of diffusion models. We express the object’s 3D geometry and appearance by means of a neural radiance field. Then, we train the radiance field to reconstruct the given input image by minimizing the usual rendering loss. At the same time, we sample random other views of the object, and constrain them with the diffusion prior, using a technique similar to DreamFusion.

We find that, out of the box, this idea does not work well. Instead, we need to make a number of improvements and

modifications. The most important change is to adequately condition the diffusion model. The idea is to configure the prior to “dream up” or sample images that may *plausibly constitute other views of the given object*. We do so by engineering the diffusion prompt from random augmentations of the given image. Only in this manner does the diffusion model provide sufficiently strong constraints to allow meaningful 3D reconstruction.

In addition to setting the prompt correctly, we also add some regularizers: shading the underlying geometry and randomly dropping out texture (also similar to DreamFusion), smoothing the normals of the surface, and fitting the model in a coarse-to-fine fashion, capturing first the overall structure of the object and only then the fine-grained details. We also focus on efficiency and base our model on Instant-NGP [29]. In this manner, we achieve reconstructions in the span of hours instead of days if we were to adopt traditional MLP-based NeRF models.

We assess our approach by using random images captured in the wild as well as existing benchmark datasets. Note that we do *not* train a fully-fledged 2D-to-3D model and we are *not* limited to specific object categories; rather, we perform reconstruction on an image-by-image basis using a pretrained 2D generator as a prior. Nonetheless, we can surpass quantitatively and qualitatively previous single-image reconstructors, including Shelf-Supervised Mesh Prediction [58], which uses supervision tailored specifically for 3D reconstruction.

More impressively, and more importantly, we obtain plausible 3D reconstructions that are a good match for the provided input image (Fig. 1). Our reconstructions are not perfect, as the diffusion prior clearly does its best to explain the available image evidence but cannot always match all the details. Even so, we believe that our results convincingly demonstrate the viability of this approach and trace a path for future improvements.

To summarize, we make the following **contributions**: (1) We propose RealFusion, a method that can extract from a single image of an object a 360° photographic 3D reconstruction without assumptions on the type of object imaged or 3D supervision of any kind; (2) We do so by leveraging an existing 2D diffusion image generator via a new single-image variant of textual inversion; (3) We also introduce new regularizers and provide an efficient implementation using InstantNGP; (4) We demonstrate state-of-the-art reconstruction results on a number of in-the-wild images and images from existing datasets when compared to alternative approaches.

2. Related work

Image-based reconstruction of appearance and geometry.

Much of the early work on 3D reconstruction is based on principles of multi-view geometry [11]. These classic meth-

ods use photometry only to match image features and then discard it and only estimate 3D shape.

The problem of reconstructing photometry and geometry together has been dramatically revitalized by the introduction of neural radiance fields (RFs). NeRF [26] in particular noticed that a coordinate MLP provides a compact and yet expressive representation of 3D fields, and can be used to model RFs with great effectiveness. Many variants of NeRF-like models have since appeared. For instance, some [24, 48, 50] use sign distance functions (SDFs) to recover cleaner geometry. These approaches assume that dozens if not hundreds of views of each scene are available for reconstruction. Here, we use them for single-image reconstruction, using a diffusion model to “dream up” the missing views.

Few-view reconstruction. Many authors have attempted to improve the statistical efficiency of NeRF-like models, by learning or incorporating various kinds of priors. Quite related to our work, NeRF-on-a-Diet [17] reduces the number of images required to learn a NeRF by generating random views and measuring their “semantic compatibility” with the available views via CLIP embeddings [35], but they still require several input views.

While CLIP is a general-purpose model learned on 2D data, other authors have learned deep networks specifically for the goal of inferring NeRFs from a small number of views. Examples include IBRNet [51], NeRF-WCE [13], PixelNeRF [60], NeRFormer [38], and ViewFormer [22]. These models still generally require more than one input view at test time, require multi-view data for training, and are often optimized for specific object categories.

Single-view reconstruction. Some authors have attempted to recover full radiance fields from single images, but this generally requires multi-view data for training, as well as learning models that are specific to a specific object category. 3D-R2N2 [5], Pix2Vox [55, 55], and LegoFormer [57] learn to reconstruct volumetric representation of simple objects, mainly from synthetic data like ShapeNet [4]. More recently, CodeNeRF [19] predicts a full radiance field, including reconstructing the photometry of the objects. AutoRF [28] learns a similar autoencoder specifically for cars.

Extracting 3D models from 2D generators. Several authors have proposed to extract 3D models from 2D image generators, originally using GANs [3, 9, 12, 30, 31, 54].

More related to our work, CLIP-Mesh [20] and Dream Fields [16] do so by using the CLIP embedding and can condition 3D generation on text. Our model is built on the recent Dream Fusion approach [33], which builds on a similar idea using a diffusion model as prior.

However, these models have been used as either pure generators or generators conditioned on vague cues such as

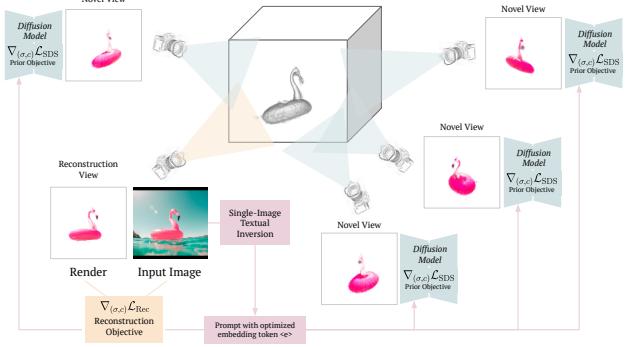


Figure 2. Method diagram. Our method optimizes a neural radiance field using two objectives simultaneously: a reconstruction objective and a prior objective. The reconstruction objective ensures that the radiance field resembles the input image from a specific, fixed view. The prior objective uses a large pre-trained diffusion model to ensure that the radiance field looks like the given object from randomly sampled novel viewpoints. The key to making this process work well is to condition the diffusion model on a prompt with a custom token $\langle e \rangle$, which is generated prior to reconstruction using single-image textual inversion. This diagram does not display our coarse-to-fine training strategy or regularization terms, both of which improve qualitative results.

class identity or text. Here, we build on similar ideas, but we apply them to the case of single-view reconstruction.

Recently, the authors of [53] have proposed to directly generate multiple 2D views of an object, which can then be reconstructed in 3D using a NeRF-like model. This is also reminiscent of our approach, but their model requires multi-view data for training, is only tested on synthetic data, and requires to explicitly sample multiple views for reconstruction (in our case they remain implicit).

Diffusion Models. Diffusion denoising probabilistic models are a class of generative models based on iteratively reversing a Markovian noising process. In vision, early works formulated the problem as learning a variational lower bound [14], or framed it as optimizing a score-based generative model [45, 46] or as the discretization of a continuous stochastic process [47]. Recent improvements includes the use of faster and deterministic sampling [14, 25, 52], class-conditional models [7, 46], text-conditional models [32], and modeling in latent space [41].

3. Method

We provide an overview and notation for the background material first (Sec. 3.1), and then discuss our RealFusion method (Sec. 3.2).

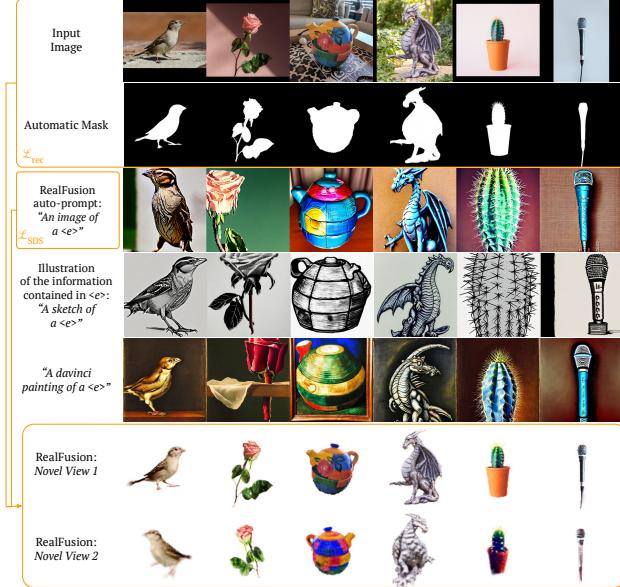


Figure 3. Examples demonstrating the level of detail of information captured by the optimized embedding $\langle e \rangle$. Rows 1-2 show input images and masks. The images are used to optimize $\langle e \rangle$ via our single-image textual inversion process. Rows 3-5 show examples of 2D images generated using $\langle e \rangle$ in new prompts, which we hope demonstrate the type of information encoded in $\langle e \rangle$. Rows 6-7 show RealFusion’s output, optimized using the prompt “An image of a $\langle e \rangle$ ”.

3.1. Radiance fields and DreamFusion

Radiance fields. A *radiance field* (RF) is a pair of functions $(\sigma(\mathbf{x}), c(\mathbf{x}))$ mapping a 3D point $\mathbf{x} \in \mathbb{R}^3$ to an opacity value $\sigma(\mathbf{x}) \in \mathbb{R}_+$ and a color value $c(\mathbf{x}) \in \mathbb{R}^3$. The RF is called *neural* when these two functions are implemented by a neural network.

The RF represents the shape and appearance of an object. In order to generate an image of it, one *renders* the RF using the emission-absorption model. Let $I \in \mathbb{R}^{3 \times H \times W}$ be an image, so that $I(u) \in \mathbb{R}^3$ is the color of pixel u . In order to compute $I(u)$, one casts a ray r_u from the camera center through the pixel, interpreted as a point on the 3D image plane (this implicitly accounts for the camera viewpoint $\pi \in SE(3)$). Then, one takes a certain number of samples $(\mathbf{x}_i \in r_u)_{i \in \mathcal{N}}$, for indices $\mathcal{N} = \{1, \dots, N\}$ taken with constant spacing Δ . The color is obtained as:

$$I(u) = \mathcal{R}(u; \sigma, c) = \sum_{i \in \mathcal{N}} (T_{i+1} - T_i) c(\mathbf{x}_i), \quad (1)$$

where $T_i = \exp(-\Delta \sum_{j=0}^{i-1} \sigma(\mathbf{x}_j))$ is the probability that a photon is transmitted from point \mathbf{x}_i back to the camera sensor without being absorbed by the material.

Importantly, the rendering function $R(u; \sigma, c)$ is differentiable, which allows training the model by means of a

standard optimizer. Specifically, the RF is fitted to a dataset $\mathcal{D} = \{(I, \pi)\}$ of images I with known camera parameters by minimizing the L^2 image reconstruction error

$$\mathcal{L}_{\text{rec}}(\sigma, c; \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(I, \pi) \in \mathcal{D}} \|I - R(\cdot; \sigma, c, \pi)\|^2. \quad (2)$$

In order to obtain good quality results, one typically requires a dataset of dozens or hundreds of views.

Here, we consider the case in which we are given *exactly one* input image I_0 corresponding to some (unknown) camera π_0 . In this case, we can also assume *any* standard viewpoint π_0 for that single camera. Optimizing Eq. (2) with a single training image leads to severe over-fitting: it is straightforward to find a pair (σ, c) that has zero loss and yet does not capture any sensible 3D model of the object. Below we will leverage a pre-trained 2D image prior to (implicitly) dream up novel views of the object and provide the missing information for 3D reconstruction.

Diffusion models. A *diffusion model* draws a sample from a probability distribution $p(I)$ by inverting a process that gradually adds noise to the image I . The diffusion process is associated with a variance schedule $\{\beta_t \in (0, 1)\}_{t=1}^T$, which defines how much noise is added at each time step. The noisy version of sample I at time t can then be written $I_t = \sqrt{\alpha_t} I + \sqrt{1 - \bar{\alpha}_t} \epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, is a sample from a Gaussian distribution (with the same dimensionality as I), $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. One then learns a denoising neural network $\hat{\epsilon} = \Phi(I_t; t)$ that takes as input the noisy image I_t and the noise level t and tries to predict the noise component ϵ .

In order to draw a sample from the distribution $p(I)$, one starts by drawing a sample $I_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, one progressively denoises the image by iterated application of Φ according to a specified sampling schedule [15, 25, 44], which terminates with I_0 sampled from $p(I)$.

Modern diffusion models are trained on large collections $\mathcal{D}' = \{I\}$ of images by minimizing the loss

$$\mathcal{L}_{\text{diff}}(\Phi; \mathcal{D}') = \frac{1}{|\mathcal{D}'|} \sum_{I \in \mathcal{D}'} \|\Phi(\sqrt{\bar{\alpha}_t} I + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) - \epsilon\|^2. \quad (3)$$

This model can be easily extended to draw samples from a distribution $p(\mathbf{x}|e)$ conditioned on a *prompt* e . Conditioning on the prompt is obtained by adding e as an additional input of the network Φ , and the strength of conditioning can be controlled via classifier-free guidance [7].

DreamFusion and Score Distillation Sampling (SDS). Given a 2D diffusion model $p(I|e)$ and a prompt e , DreamFusion extracts from it a 3D rendition of the corresponding concept, represented by a RF (σ, c) . It does so by randomly sampling a camera parameter π , rendering a corresponding view I_π , assessing the likelihood of the view based on the

model $p(I_\pi | e)$, and updating the RF to increase the likelihood of the generated view based on the model.

In practice, DreamFusion uses the denoiser network as a frozen critic and takes a gradient step

$$\nabla_{(\sigma,c)} \mathcal{L}_{\text{SDS}}(\sigma, c; \pi, e, t) = E_{t,\epsilon} \left[w(t)(\Phi(\alpha_t I + \sigma_t \epsilon; t, e) - \epsilon) \cdot \nabla_{(\sigma,c)} I \right], \quad (4)$$

where $I = R(\cdot; \sigma, c, \pi)$. is the image rendered from a given viewpoint π and prompt e . This process is called *Score Distillation Sampling* (SDS).

Note that Eq. (4) differs from simply optimizing the standard diffusion model objective because it does not include the Jacobian term for Φ . In practice, removing this term both improves generation quality and reduces computational and memory requirements.

One final aspect of DreamFusion is essential for understanding our contribution in the following section: DreamFusion finds that it is necessary to use classifier-free guidance [7] with a very high guidance weight of 100, much larger than one would use for image sampling, in order to obtain good 3D shapes. As a result, the generations tend to have limited diversity; they produce only the most likely objects for a given prompt, which is incompatible with our goal of reconstructing any given object.

3.2. RealFusion

Our goal is to reconstruct a 3D model of the object contained in a single image I_0 , utilizing the prior captured in the diffusion model Φ to make up for the missing information. We will achieve this by optimizing a radiance field using two simultaneous objectives: (1) a reconstruction objective Eq. (2) from a fixed viewpoint, and (2) a SDS-based prior objective Eq. (4) on novel views randomly sampled at each iteration. Figure 2 provides a diagram of the entire system.

Single-image textual inversion as a substitute for alternative views. The most important component of our method is the use of single-image textual inversion as a substitute for alternative views. Ideally, we would like to condition our reconstruction process on multi-view images of the object in I_0 , *i.e.* on samples from $p(I|I_0)$. Since these images are not available, we instead synthesize a text prompt $e^{(I_0)}$ specifically for our image I_0 as a proxy for this multi-view information.

Our idea, then, is to engineer a prompt $e^{(I_0)}$ to provide a useful approximation of $p(I|I_0)$. We do so by generating random augmentations $g(I_0)$, $g \in G$ of the input image, which serve as pseudo-alternative-views. We use these augmentations as a mini-dataset $\mathcal{D}' = \{g(I_0)\}_{g \in G}$ and optimize the diffusion loss Eq. (3) $\mathcal{L}_{\text{diff}}(\Phi(\cdot; e^{(I_0)}))$ with respect to the prompt $e^{(I_0)}$, while freezing all other text embeddings and model parameters.

In practice, our prompt is derived automatically from templates like “an image of a $\langle e \rangle$ ”, where “ $\langle e \rangle$ ” ($= e^{(I_0)}$) is a new token introduced to the vocabulary of the text encoder of our diffusion model (see Appendix A for details). Our optimization procedure mirrors and generalizes the recently-proposed textual-inversion method of [10]. Differently from [10], we work in the single-image setting and utilize image augmentations for training rather than multiple views.

To help convey the intuition behind $\langle e \rangle$, consider an attempt at reconstructing an image of a fish using the generic text prompt “An image of a fish” with losses Eqs. (3) and (4). In our experience, this often produces a reconstruction which looks like the input fish from the input viewpoint, but looks like some *different, more-generic* fish from the backside. By contrast, using the prompt “An image of a $\langle e \rangle$ ”, the reconstruction resembles the input fish from all angles. An example of exactly this case is shown in Figure 7.

Finally, Figure 3 demonstrates the amount of detail captured in the embedding $\langle e \rangle$.

Coarse-to-fine training. In order to describe our coarse-to-fine training methodology, it is necessary to first briefly introduce our underlying RF model, a InstantNGP [29]. InstantNGP is a grid-based model which stores features at the vertices of a set of feature grids $\{G_i\}_{i=1}^L$ at multiple resolutions. The resolution of these grids is chosen to be a geometric progression between the coarsest and finest resolutions, and feature grids are trained simultaneously.

We choose a InstantNGP over a conventional MLP-based NeRF due to its computational efficiency and training speed. However, the optimization procedure occasionally produces small irregularities on the surface of the object. We find that training in a coarse-to-fine manner helps to alleviate these issues: for the first half of training we only optimize the lower-resolution feature grids $\{G_i\}_{i=1}^{L/2}$, and then in the second half of training we optimize all feature grids $\{G_i\}_{i=1}^L$. Using this strategy, we obtain the benefits of both efficient training and high-quality results.

Normal vector regularization. Next, we introduce a new regularization term to encourage our geometry to have smooth normals. The introduction of this term is motivated by the observation that our RF model occasionally generated noisy-looking surfaces with low-level artifacts. To address these artifacts, we encourage our RF to have smoothly varying normal vectors. Notably, we perform this regularization in 2D rather than in 3D.

At each iteration, in addition to computing RGB and opacity values, we also compute normals for each point along the ray and aggregate these via the raymarching equa-

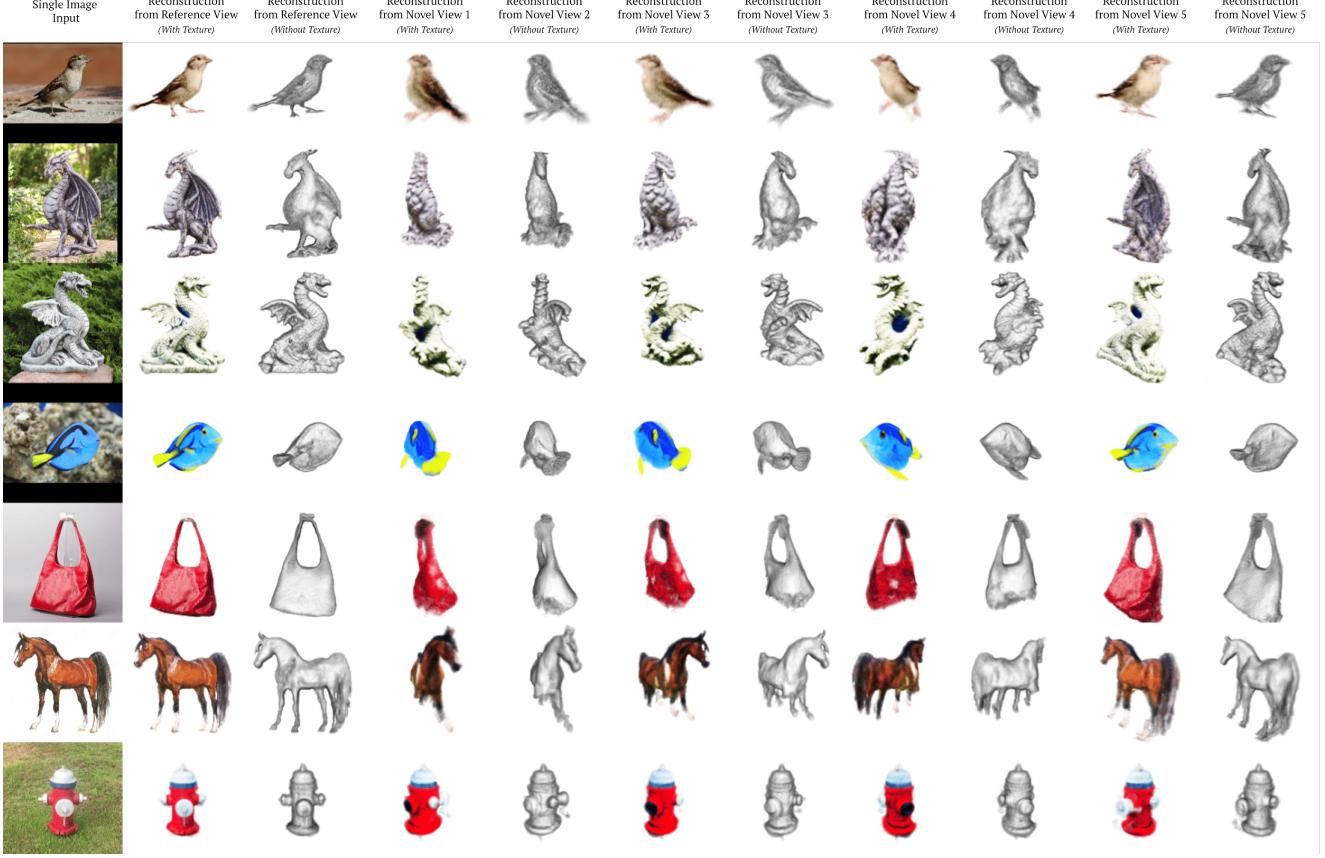


Figure 4. **Qualitative results.** RealFusion reconstructions from a single input view. Each pair of columns shows the textured object and the underlying 3D shape, as a shaded surface. Different pairs of columns show different viewpoints.

tion to obtain normals $N \in \mathcal{R}^{H \times W \times 3}$.¹ Our loss is:

$$\mathcal{L}_{\text{normals}} = \|N - \text{stopgrad}(\text{blur}(N, k))\|^2 \quad (5)$$

where stopgrad is a stop-gradient operation and blur(\cdot, k) is a Gaussian blur with kernel size k (we use $k = 9$).

Although it may be more common to regularize normals in 3D, we found that operating in 2D reduced the variance of the regularization term and led to superior results.

Mask loss. In addition to the input image, our model also utilizes a mask of the object that one wishes to reconstruct. In practice, we use an off-the-shelf image matting model to obtain this mask for all images.

We incorporate this mask in a simple manner by adding a simple L^2 loss term on the difference between the rendered opacities from the fixed reference viewpoint $\mathcal{R}(\sigma, \pi_0) \in \mathcal{R}^{H \times W}$ and the object mask M : $\mathcal{L}_{\text{rec,mask}} = \|O - M\|^2$

¹Normals may be computed either by taking the gradient of the density field or by using finite differences. We found that using finite differences worked well in practice.

Our final objective then consists of four terms:

$$\begin{aligned} \nabla_{\sigma,c}\mathcal{L} &= \nabla\mathcal{L}_{\text{SDS}} + \lambda_{\text{normals}} \cdot \nabla\mathcal{L}_{\text{normals}} \\ &\quad + \lambda_{\text{image}} \cdot \nabla\mathcal{L}_{\text{image}} + \lambda_{\text{mask}} \cdot \nabla\mathcal{L}_{\text{mask}} \end{aligned} \quad (6)$$

where the top line in the equation above corresponds to our prior objective and the bottom line corresponds to our reconstruction objective.

4. Experiments

4.1. Implementation details

Regarding hyperparameters, we use *essentially the same set of hyper-parameters for all experiments*—there is no per-scene hyper-parameter optimization.² For our diffusion model prior, we employ the open-source *Stable Diffusion* model [41] trained on the LAION [43] dataset of text-image pairs. For our InstantNGP [29] model, we use a model with

²There is one small exception to this rule, which is that for a few number of images where the camera angle was clearly at an angle higher than 15°, we took a camera angle of 30 or 40deg.



Figure 5. Qualitative comparison with prior work. We show the results of our method and the category-level method of [59] on real-world images from the CO3D dataset [38]. Each pair of rows show two novel views produced by [59] and our method. For [59], we use category-specific models for each CO3D category (in this case, motorcycles, cups, and backpacks). Despite not requiring any category-specific information, our method is able to reconstruct objects at a higher level of detail than [59].

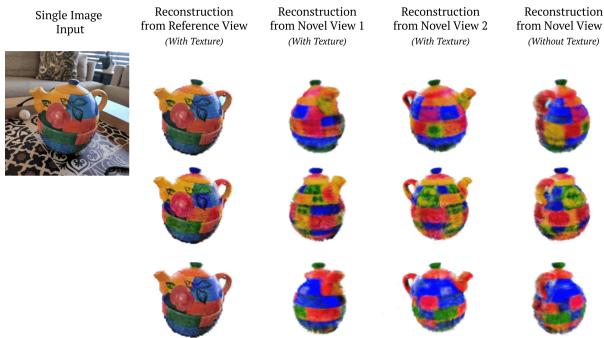


Figure 6. A demonstration of multi-modal image reconstruction. Above, we see our method’s ability to generate a diverse set of object reconstructions given the same input image. In particular, the method produces different textures on the backsides of the generated objects, despite all objects matching the input image from the reference view.

16 resolution levels, a feature dimension of 2, and a maximum resolution of 2048, trained in a coarse-to-fine manner as explained above.

The camera for reconstruction is placed looking at the origin on a sphere of radius 1.8, at an angle of 15deg above the plane. At each optimization step, we first render from the reconstruction camera and compute our reconstruction losses \mathcal{L}_{rec} and $\mathcal{L}_{\text{rec},\text{mask}}$. We then render from a randomly sampled camera to obtain a novel view, and use this view for \mathcal{L}_{sds} and $\mathcal{L}_{\text{normals}}$. We use $\lambda_{\text{image}} = 5.0$, $\lambda_{\text{mask}} = 0.5$, and $\lambda_{\text{normal}} = 0.5$.

Regarding camera sampling, lighting, and shading, we keep nearly all parameters the same as [33]. This in-

Table 1. Quantitative comparison. We compare our method with Shelf-Supervised [59] on seven object categories. The F-score and CLIP-similarity metrics are designed to measure the quality of reconstruction shape and appearance, respectively. For both metrics, higher is better. Metrics are averaged over three images per category. Our method outperforms [59] in aggregate, despite the fact that [59] uses a *different category-specific model* for each category.

	F-score		CLIP-similarity	
	[59]	Ours	[59]	Ours
Backpack	7.58	12.22	0.72	0.74
Chair	8.26	10.23	0.65	0.76
Motorcycle	8.66	8.72	0.69	0.70
Orange	6.27	10.16	0.71	0.74
Skateboard	7.74	5.89	0.74	0.74
Teddybear	12.89	10.08	0.73	0.82
Vase	6.30	9.72	0.69	0.71
Mean	8.24	9.58	0.70	0.74

cludes the use of diffuse and textureless shading stochastic throughout the course of optimization, after an initial warmup period of albedo-only shading. Complete details regarding this and other aspects of our training setup are provided in the supplementary material.

4.2. Quantitative results

There are only few methods that attempt to reconstruct arbitrary objects in 3D. The most recent and best-performing of these is Shelf-Supervised Mesh Prediction [58], which we compare here. They provide 50 pre-trained category-level models for 50 different categories in OpenImages [23]. Since we aim to compute metrics using 3D or multi-view ground truth, we evaluate on seven categories in the CO3D dataset [39] with corresponding Open-Images categories. For each of these seven categories, we select three images at random and run both RealFusion and Shelf-Supervised to obtain reconstructions.

We first test the quality of the recovered 3D shape in Fig. 5. Shelf-Supervised directly predicts a mesh. We extract one from our predicted radiance fields using marching cubes. CO3D comes with sparse point-cloud reconstruction of the objects obtained using multi-view geometry. For evaluation, we sample points from the reconstructed meshes and align them optimally with the ground truth point cloud by first estimating a scaling factor and then using Iterated Closest Point (ICP). Finally, we compute F-score with threshold 0.05 to measure the distance between the predicted and ground truth point clouds. Results are shown in Tab. 1.

In order to evaluate the quality of the reproduced appearance, we also compare novel-view renderings from our and their method (Tab. 1). Ideally, these renderings should pro-

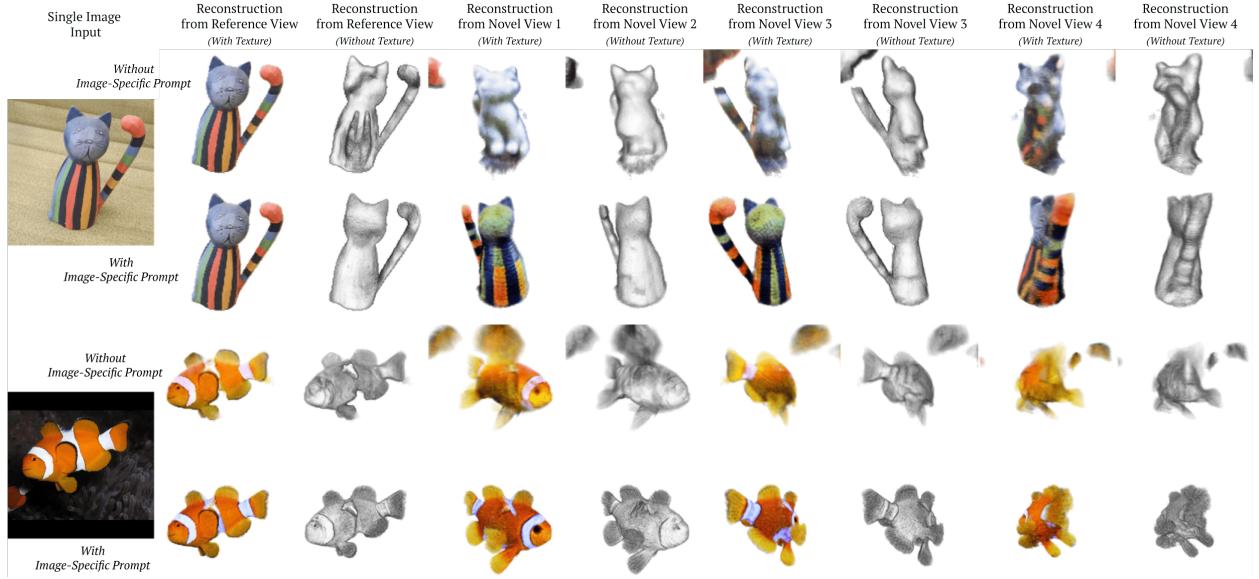


Figure 7. **A visualization of the effect of single-image textual inversion on reconstruction quality.** In each pair of rows, the top row shows the result of utilizing a standard text prompt for our diffusion-model-based loss (*e.g.* “An image of a statue of a cat”). The bottom row shows the result of utilizing a text prompt optimized for the input image in a fully-automatic manner; this textual inversion process dramatically improves object reconstruction.

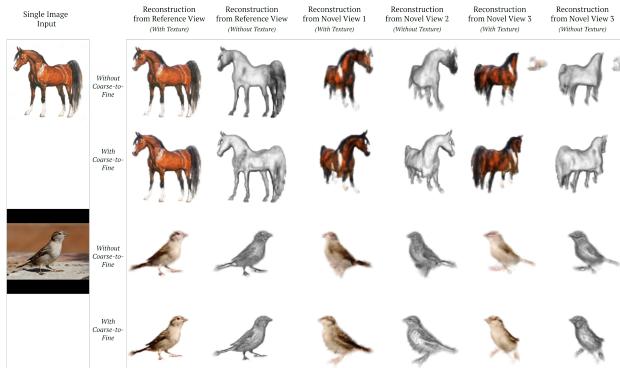


Figure 8. **Effect of coarse-to-fine training.** The top row of each pair is generated by optimizing all levels of a multi-resolution 3D feature grid from the first optimization step, whereas the bottom row is optimized in a coarse-to-fine manner.

duce views that are visually close to the real views. In order to test this hypothesis, we check whether the generated views are close or not to the other views given in CO3D. We then report the CLIP embedding similarity of the generated images with respect to the closest CO3D view available (*i.e.* the view with maximum similarity).

4.3. Qualitative results

Figure 4 shows additional qualitative results from multiple viewpoints. Having a single image of an object means that several 3D reconstructions are possible. Figure 6 explores the ability of RealFusion to sample the space of pos-



Figure 9. **Effect of normal smoothness on reconstruction quality.** Each pair of rows show the reconstruction without and with the normal smoothness regularization term equation 5. The regularizer improves the visual appearance of surfaces and reduces the number of irregularities on the surface of reconstructed objects. In most cases, we also find that it helps to improve the overall realism of the reconstructed shape.

sible solutions by repeating the reconstruction several times, starting from the same input image. There is little variance in the reconstructions of the front of the object, but quite a large variance for its back, as expected.

Figure 11 shows two typical failure modes of RealFusion: in some cases the model fails to converge, and in others it copies the front view to the back of the object, even if this is not semantically correct.

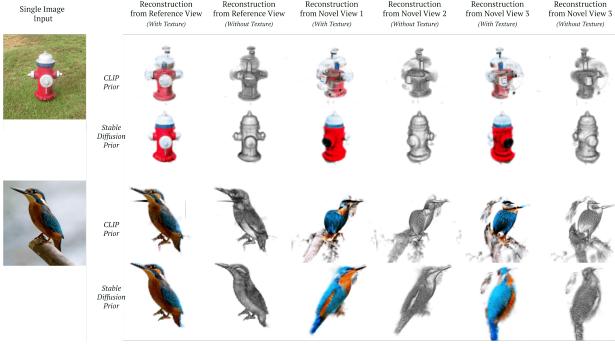


Figure 10. **Comparing Stable Diffusion and CLIP priors.** Results from two different priors: Stable Diffusion [41] and CLIP [34]. Stable Diffusion yields much higher-quality reconstructions, capturing more plausible object shapes.

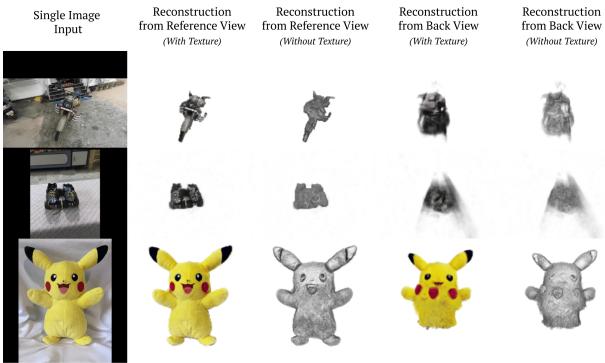


Figure 11. **Failure cases.** In the first two examples, the model simply fails to properly reconstruct the object geometry, and produces a semi-transparent scene which lacks a well-defined geometry. The third case is different in that the geometry is highly realistic, but the texture paints two Pikachu faces, one on each side of the object; this problem is sometimes called the Janus problem, after the two-faced Roman god.

4.4. Analysis and Ablations

One of the key components of RealFusion is our use of single-image textual inversion, which allows the model to correctly imagine novel views of a specific object. Figure 7 shows that this component plays indeed a critical role in the quality of the reconstructions. Without textual inversion, the model often reconstructs the backside of the object in the form of a generic instance from the object category. For example, the backside of the cat statue in the top row of Fig. 7 is essentially a different statue of a more generic-looking cat, whereas the model trained with textual inversion resembles the true object from all angles.

Other components of the model are also significant. Figure 9 shows that the normal smoothness regularizer of Eq. (5) results in smoother, more realistic meshes and reduces the number of artifacts. Figure 8 shows that coarse-

to-fine optimization reduces the presence of low-level artifacts and results in smoother, visually pleasing surfaces. Fig. 10 shows that using Stable Diffusion works significantly better than relying on an alternative such as CLIP.

5. Conclusions

We have introduced RealFusion, a new approach to obtain full 360° photographic reconstructions of any object given a single image of it. Given an off-the-shelf diffusion model trained using only 2D images and no special supervision for 3D reconstruction, as well as a single view of the target object, we have shown how to select the model prompt to imagine other views of the object. We have used this conditional prior to learn an efficient, multi-scale radiance field representation of the reconstructed object, incorporating an additional regularizer to smooth out the reconstructed surface. The resulting method can generate plausible 3D reconstructions of objects captured in the wild which are faithful to the input image. Future works include specializing the diffusion model for the task of new-view synthesis and incorporating dynamics to reconstruct animated 3D scenes.

Ethics. We use the CO3D dataset in a manner compatible with their terms. CO3D does not contain personal information. Please see <https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html> for further information on ethics.

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

References

- [1] David Bau, Jun-Yan Zhu, Jonas Wulff, William S. Peebles, Bolei Zhou, Hendrik Strobelt, and Antonio Torralba. Seeing what a GAN cannot generate. In *Proc. ICCV*, 2019. 2
- [2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. ICCV*, 2021. 15
- [3] 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, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *Proc. CVPR*, 2022. 2, 3
- [4] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet an information-rich 3d model repository. *arXiv.cs*, abs/1512.03012, 2015. 3
- [5] 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. 3
- [6] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 13
- [7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794, 2021. 3, 4, 5
- [8] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998. 13
- [9] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3D shape induction from 2D views of multiple objects. In *arXiv*, 2016. 2, 3
- [10] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 5, 13
- [11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2
- [12] Philipp Henzler, Niloy J. Mitra, and Tobias Ritschel. Escaping plato’s cave using adversarial training: 3D shape from unstructured 2D image collections. In *Proc. ICCV*, 2019. 2, 3
- [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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3
- [14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. NeurIPS*, 2020. 3
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020. 4
- [16] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proc. CVPR*, 2022. 3
- [17] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proc. ICCV*, 2021. 3
- [18] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5885–5894, October 2021. 13
- [19] Wonbong Jang and Lourdes Agapito. CodeNeRF: Disentangled neural radiance fields for object categories. In *Proc. ICCV*, 2021. 3
- [20] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Pop Tiberiu. CLIP-Mesh: Generating textured meshes from text using pretrained image-text models. *SIGGRAPH Asia*, 2022. 3
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12, 13
- [22] Jonáš Kulhánek, Erik Derner, Torsten Sattler, and Robert Babuška. ViewFormer: NeRF-free neural rendering from few images using transformers. In *Proc. ECCV*, 2022. 3
- [23] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 7
- [24] Chen-Hsuan Lin, Chaoyang Wang, and Simon Lucey. SDF-SRN: learning signed distance 3d object reconstruction from static images. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Proc. NeurIPS*, 2020. 3
- [25] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2021. 3, 4
- [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. 3
- [27] 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 *ECCV*, 2020. 13
- [28] Norman Müller, Andrea Simonelli, Lorenzo Porzi, Samuel Rota Bulò, Matthias Nießner, and Peter Kortscheder. Autorf: Learning 3d object radiance fields from single view observations. *CoRR*, abs/2204.03593, arXiv.cs. 3
- [29] Thomas Muller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 2, 5, 6
- [30] 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, 3
- [31] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy J. Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *Proc. NeurIPS*, 2020. 2, 3
- [32] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 3
- [33] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv.cs*, abs/2209.14988, 2022. 2, 3, 7, 12, 13, 14
- [34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2, 9
- [35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya

- Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. ICML*, volume 139, pages 8748–8763, 2021. 3
- [36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. 2
- [37] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv:1907.01341*, 2019. 14
- [38] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021. 3, 7
- [39] 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. 7
- [40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2
- [41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. CVPR*, 2022. 3, 6, 9, 12, 17
- [42] 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. 2
- [43] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 2, 6
- [44] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 4
- [45] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Proc. NeurIPS*, 2019. 3
- [46] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. 3
- [47] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proc. ICLR*, 2021. 3
- [48] Itsuki Ueda, Yoshihiro Fukuhara, Hirokatsu Kataoka, Hiroyuki Aizawa, Hidehiko Shishido, and Itaru Kitahara. Neural Density-Distance fields. In *Proc. ECCV*, 2022. 3
- [49] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 12
- [50] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv.cs*, abs/2106.10689, 2021. 3
- [51] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibnet: Learning multi-view image-based rendering. In *Proc. CVPR*, 2021. 3
- [52] Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In *International Conference on Learning Representations*, 2021. 3
- [53] 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. 3
- [54] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Proc. NeurIPS*, 2016. 2, 3
- [55] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun. Pix2vox++: Multi-scale context-aware 3d object reconstruction from single and multiple images. *Int. J. Comput. Vis.*, 128(12), 2020. 3
- [56] 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. 2022. 13
- [57] Farid Yagubayli, Alessio Tonioni, and Federico Tombari. Legoformer: Transformers for block-by-block multi-view 3d reconstruction. *arXiv.cs*, abs/2106.12102, 2021. 3
- [58] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild. *CoRR*, abs/2102.06195, 2021. 2, 7
- [59] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 7
- [60] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 3, 13
- [61] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 14

A. Implementation Details

In this section, we provide full implementation details which were omitted from the main text due to space constraints. Most of these details follow [33], but a few are slightly modified.

Shading. We consider three different types of shading: albedo, diffuse, and textureless. For albedo, we simply render the RGB color of each ray as given by our model:

$$I(u) = I_\rho(u) = \mathcal{R}(u; \sigma, c)$$

For diffuse, we also compute the surface normal n as the normalized negative gradient of the density with respect to u . Then, given a point light l with color l_ρ and an ambient light with color l_a , we render

$$I(u) = I_\rho(u) \circ (l_\rho \circ \max(0, n \cdot \frac{l-u}{\|l-u\|} + l_a))$$

For textureless, we use the same equation with $I_\rho(u)$ replaced by white $(1, 1, 1)$.

For the reconstruction view, we only use albedo shading. For the random view (i.e. the view used for the prior objectives), we use albedo shading for the first 1000 steps of training by setting $l_a = 1.0$ and $l_\rho = 0.0$. Afterwards we use $l_a = 0.1$ and $l_\rho = 0.9$, and we select stochastically between albedo, diffuse, and textureless with probabilities 0.2, 0.4, and 0.4, respectively.

We obtain the surface normal using finite differences:

$$n = \frac{1}{2 \cdot \epsilon} \begin{pmatrix} I(u + \epsilon_x) - I(u - \epsilon_x) \\ I(u + \epsilon_y) - I(u - \epsilon_y) \\ I(u + \epsilon_z) - I(u - \epsilon_z) \end{pmatrix}$$

where $\epsilon_x = (\epsilon, 0, 0)$, $\epsilon_y = (0, \epsilon, 0)$, and $\epsilon_z = (0, 0, \epsilon)$

Density bias. As in [33], we add a small Gaussian blob of density to the origin of the scene in order to assist with the early stages of optimization. This density takes the form

$$\sigma_{\text{init}}(\mu) = \lambda \cdot e^{-\|\mu\|^2/(2\nu^2)}$$

with $\lambda = 5$ and $\nu = 0.2$.

Camera. The fixed camera for reconstruction is placed at a distance of 1.8 from the origin, oriented toward the origin, at an elevation of 15° above the horizontal plane. For a small number of scenes in which the object of interest is clearly seen from overhead, the reconstruction camera is placed at an elevation of 40° .

The camera for the prior objectives is sampled randomly at each iteration. Its distance from the origin is sampled uniformly from $[1.0, 1.5]$. Its azimuthal angle is sampled uniformly at random from the 360° around the object. Its elevation is sampled uniformly in degree space from -10° to 90° with probability 0.5 and uniformly on the upper hemisphere with probability 0.5. The field of view is uniformly

sampled between 40 and 70. The camera is oriented toward the origin. Additionally, every tenth iteration, we place the prior camera near the reconstruction camera: its location is sampled from the prior camera’s location perturbed by Gaussian noise with mean 0 and variance 1.

Lighting. We sample the position of the point light by adding a noise vector $\eta \sim \mathcal{N}(0, 1)$ to the position of the prior camera.

View-Dependent Prompt. We add a view-dependent suffix to our text prompt based on the location of the prior camera relative to the reconstruction camera. If the prior camera is placed at an elevation of above 60° , the text prompt receives the suffix “overhead view.” If it is at an elevation below 0° , the text receives “bottom view.” Otherwise, for azimuthal angles of $\pm 30^\circ$, $\pm 30 - 90^\circ$, or $\pm 90 - 180^\circ$ in either direction of the reconstruction camera, it receives the suffices “front view,” “side view,” or “bottom view,” respectively.

InstantNGP. Our InstantNGP parameterizes the density and albedo inside a bounding box around the origin with side length 0.75. It is a multi-resolution feature grid with 16 levels. With coarse-to-fine training, only the first 8 (lowest-resolution) levels are used during the first half of training, while the others are masked with zeros. Each feature grid has dimensionality 2. The features from these grids are stacked and fed to a 3-layer MLP with 64 hidden units.

Rendering and diffusion prior. We render at resolution 96px. Since Stable Diffusion [41] is designed for images with resolution 512px, we upsample renders to 512px before passing them to the Stable Diffusion latent space encoder (i.e. the VAE). We add noise in latent space, sampling $t \sim \mathcal{U}(0.02, 0.98)$. We use classifier-free guidance strength 100. We found that results with classifier-free guidance strength above 30 produced good results; below 30 led to many more geometric deformities. Although we do not backpropagate through the Stable Diffusion UNet for \mathcal{L}_{SDS} , we do backpropagate through the latent space encoder.

Optimization. We optimize using the Adam [21] optimizer with learning rate $1e - 3$ for 5000 iterations. The optimization process takes approximately 45 minutes on a single V100 GPU.

Background model. For our background model, we use a two-layer MLP which takes the viewing direction as input. This model is purposefully weak, such that the model cannot trivially optimize its objectives by using the background.

Additional regularizers. We additionally employ two regularizers on our density field. The first is the orientation loss from Ref-NeRF [49], also used in DreamFusion [33], for which we use $\lambda_{\text{orient}} = 0.01$. The second is an entropy loss which encourages points to be either fully transparent or fully opaque: $\mathcal{L}_{\text{entropy}} = (w \cdot \log_2(w) - (1-w) \cdot \log_2(1 - w))$

```

transform = T.Compose([
    T.RandomApply([T.RandomRotation(degrees=10, fill=255)], p=0.75),
    T.RandomResizedCrop(image_size, scale=(0.70, 1.3)),
    T.RandomApply([T.ColorJitter(0.04, 0.04, 0.04, 0.04)], p=0.75),
    T.RandomGrayscale(p=0.10),
    T.RandomApply([T.GaussianBlur(5, (0.1, 2))], p=0.10),
    T.RandomHorizontalFlip(),
])

```

Figure 12. PyTorch code for the image augmentations used for single-image textual inversion.

w) where w is the cumulative sum of density weights computed as part of the NeRF rendering equation (Equation 1).

Single-image textual inversion. Our single-image textual inversion step, which is a variant of textual inversion [10], entails optimizing a token e introduced into the diffusion model text encode to match an input image. The key to making this optimization successful given only a single image is the use of heavy image augmentations, shown in Fig. 12. We optimize using these augmentations for a total of 3000 steps using the Adam optimizer [21] with image size 512px, batch size 16, learning rate $5 \cdot 10^{-4}$, and weight decay $1 \cdot 10^{-2}$.

The embedding e can be initialized either randomly, manually (by selecting a token from the vocabulary that matches the object), or using an automated method.

One automated method that we found to be successful was to use CLIP (which is also the text encoder of the Stable Diffusion model) to infer a starting token to initialize the inversion procedure. For this automated procedure, we begin by considering the set of all tokens in the CLIP text tokenizer which are nouns, according to the WordNet [8] database. We use only nouns because we aim to reconstruct objects, not reproduce styles or visual properties. We then compute text embeddings for captions of the form “An image of a \langle token \rangle ” using each of these tokens. Separately, we compute the image embedding for the input image. Finally, we take the token whose caption is most similar to the image embedding as initialization for our textual inversion procedure.

We use the manual initialization method for the examples in the main paper and we use the automated initialization method for the examples in the supplemental material (i.e. those included below).

B. Method diagram

We provide a diagram illustrating our method in Fig. 2.

C. Additional Qualitative Examples

In Fig. 13, we show additional examples of reconstructions from our model. We see that our method is often able

to reconstruct plausible geometries and object backsides.

D. Additional Comparisons

We provide additional comparisons to recent single-view reconstruction methods on the lego scene from the synthetic NeRF [27] dataset. We compare on the special test set created by SinNeRF [56], which consists of 60 views very close to the reference view. We emphasize that our method is not tailored to this setting, whereas the other methods are designed specifically for it. For example, some other methods work by warping the input image, which only performs well for novel views close to the reference view.

Table 2. **Novel view synthesis comparison.** A comparison of RealFusion against recent single-view reconstruction methods on the task of novel view synthesis on the synthetic lego scene from NeRF [27]. These numbers are computed on the test set rendered by SinNeRF [56], which contains 60 views very close to the reference view. This is a setting highly favorable to methods that use depth supervision, such as DS-NeRF and SinNeRF .

	Depth?	PSNR	SSIM	LPIPS
PixelNeRF [60]		14.3	0.72	0.22
DietNeRF [18]		15.0	0.72	0.20
DS-NeRF [6]	✓	16.6	0.77	0.16
SinNeRF [56]	✓	21.0	0.82	0.09
RealFusion		16.5	0.76	0.25

E. Text-to-Image-to-3D

In this section, we explore the idea of reconstructing a 3D object from a text prompt alone by first using the text prompt to generate an image, and then reconstructing this image using RealFusion.

We show examples of text-to-image-to-3D generation in Fig. 14.

Compared to the one-step procedure of [33] (i.e. text-to-3D), this two-step procedure (i.e. text-to-image-to-3D) has the advantage that it may be easier for users to control. Under our setup, users can first sample a large number of images from a 2D diffusion model such as Stable Diffusion,

select their desired image, and then lift it to 3D using RealFusion. It is possible that this setup could help address the issue of diversity of generation discussed in [33]. Additionally, in this setting, we find that it is usually not necessary to use single-image textual inversion, since the images sampled in the first stage are already extremely well-aligned with their respective prompts.

F. Analysis of Failure Cases

In Fig. 15, we show additional examples of failure cases from our model. Below, we analyzed what we find to be our three most common failure cases. The techniques we apply in RealFusion (single-image textual inversion, normals smoothing, and coarse-to-fine training) make these failure cases less frequent and less severe, but they still occur on various images.

Neural fields lacking well-defined geometry. One failure case of our method consists of the generation of a semi-transparent neural field which does not have a well-defined geometry. These fields tend to look like the input image when seen from the reference viewpoint, but do not resemble plausible objects when seen from other viewpoints. We note that this behavior is extremely common when using CLIP as a prior model, but it occurs occasionally even when using Stable Diffusion and \mathcal{L}_{SDS} .

Floater. Another failure case involves “floaters,” or disconnected parts of the scene which appear close to the camera. These floaters sometimes appear in front of the reference view as to make the corresponding render look like the input image. Without image-specific prompts, these floaters are a very big issue, appearing in the majority of reconstructions. When using image-specific prompts, the issue of floaters is greatly (but not entirely) alleviated.

The Janus Problem. Named after the two-faced Roman god Janus, the “Janus problem” refers to reconstructions which have two or more faces. This problem arises because the loss function tries to make the render of every view look like the input image, at least to a certain extent.

Our use of view-specific prompting partially alleviates this issue. For example, when we render an image of a panda from the back, we optimize using the text prompt “An image of a ⟨object⟩, back view”, where “⟨object⟩” is our image-specific token corresponding to the image of a panda. However, even with view-specific prompting, this problem still occurs. This problem is visible with the panda in Fig. 14 (row 2). We note that this problem is not unique to our method; it can also be seen with [33] (see Figure 9, last row).

G. Unsuccessful Experiments and Regularization Losses

In the process of developing our method, we experimented with numerous ideas, losses, and regularization terms which were not included in our final method because they either did not improve reconstruction quality or did not improve it enough to justify their complexity. Here, we describe some of these ideas for the benefit of future researchers working on this problem.

Using DM for reconstruction loss. One idea we tried involved using the diffusion model within our reconstruction objective as well as our prior objective. This involved a modified version of \mathcal{L}_{SDS} in which we compared the noise predicted by the diffusion model for our noisy rendered image to the noise predicted by the diffusion model for a noisy version of our input image. We found that with this loss we were able to reconstruct the input image to a certain degree, but that we did not match the exact input image colors or textures.

Normals smoothing in 3D. Our normals smoothing term operates in 2D, using normals rendered via the NeRF equation. We also tried different ways of smoothing normals in 3D. However, possibly due to our grid-based radiance field and/or our finite difference-based normals computation, we found that these regularization terms were all very noisy and harmful to reconstruction quality.

Using monocular depth. We tried incorporating monocular depth predictions into the pipeline, using pre-trained monocular depth networks such as MiDaS [37]. Specifically, we enforced that the depth rendered from the reference view matched the depth predicted by MiDaS for the input image. We found that this additional depth loss in most instances did not noticeably improve reconstruction quality and in some cases was harmful. Nonetheless, these results are not conclusive and future work could pursue other ways of integrating these components.

Using LPIPS and SSIM reconstruction losses. We tried using LPIPS [61] and SSIM losses in place of our L2 reconstruction loss. We found that LPIPS performed similarly to L2, but incurred additional computation and memory usage. We found that SSIM without either L2 and LPIPS resulted in worse reconstruction quality, but that it yielded fine results when combined with them. We did not include it in our final objective for the sake of simplicity.

Rendering at higher resolutions. Since Stable Diffusion operates on images of resolution 512px, it is conceivable that rendering at higher resolution would be beneficial with regard to the prior loss. However, we found no noticeable difference in quality when rendering at higher resolutions than 96px or 128px. For computational purposes, we used resolution 96px for all experiments in the main paper.

Using DINO-based prior losses. Similarly to the CLIP prior loss, one could imagine using other networks to encourage renders from novel views to be semantically similar to the input image. Due to the widespread success of the DINO [2] models in unsupervised learning, we tried using DINO feature losses in addition to the Stable Diffusion prior loss. Specifically, for each image rendered from a novel view, we computed a DINO image embedding and maximized its cosine similarity with the DINO image embedding of the reference image. We found that this did not noticeably improve or degrade performance. For purposes of simplicity, we did not include it.

H. Links to Images for Qualitative Results

For our qualitative results, we primarily use images from datasets such as Co3D. We also use a small number of images sourced directly from the web to show that our method works on uncurated web data. We provide links to all of these images on our [project website](#).

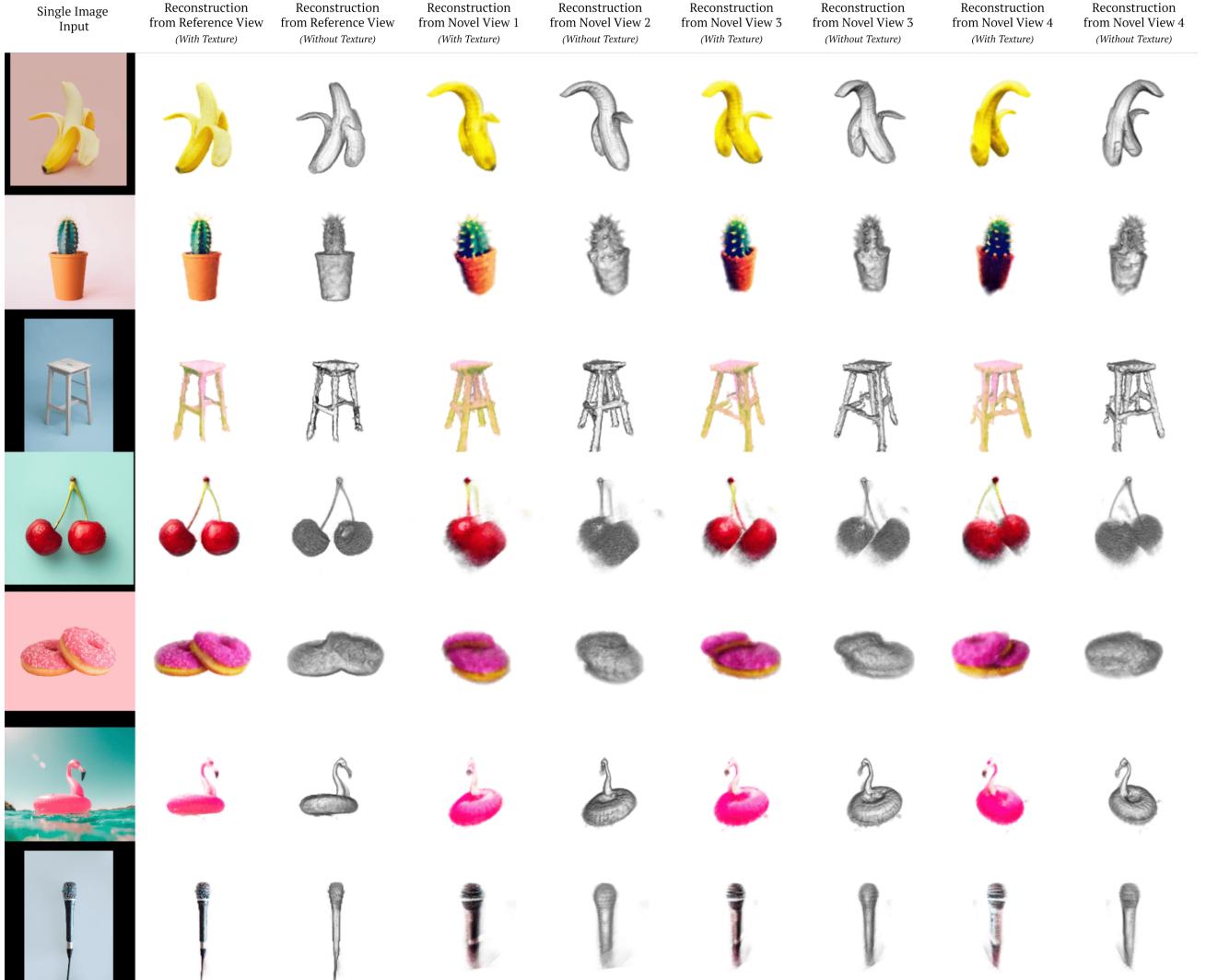


Figure 13. **Additional qualitative examples.** This figure presents additional qualitative examples from our model. The first column shows the input image. The second column shows the reconstruction from the reference viewpoint. The following columns show renders from novel viewpoints, demonstrating that our model is able to reconstruct plausible object shapes.

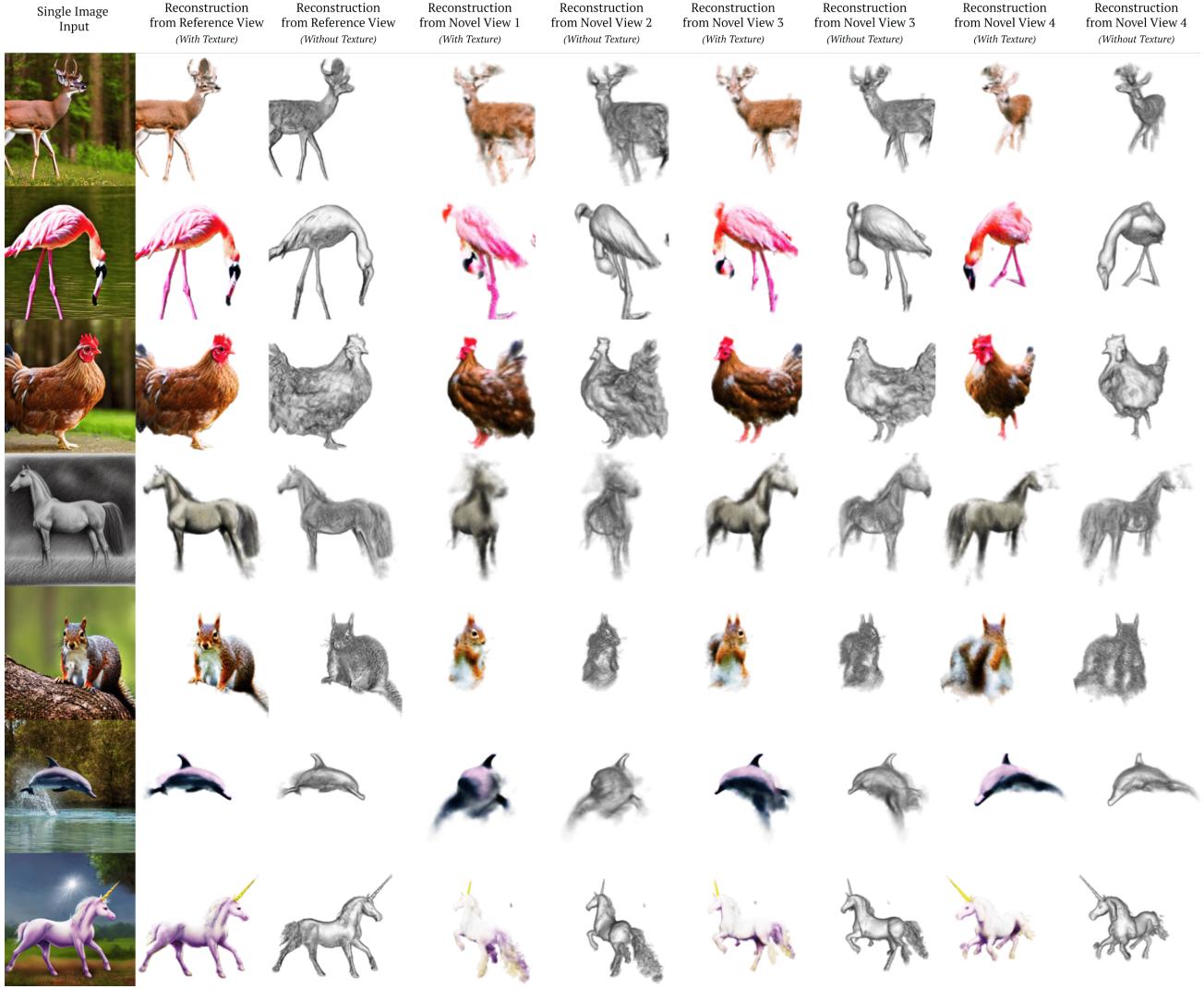


Figure 14. **Text-to-Image-to-3D.** This figure presents examples from our model using images generated directly from text prompts using Stable Diffusion [41]. The images were generated with the prompt “An image of a ___” where the blank space is replaced by “deer”, “flamingo”, “hen”, “pencil drawing of a horse”, “squirrel”, “dolphin”, and “unicorn”, respectively. The results demonstrate that our method is able to reconstruct plausible object shapes even from synthetic images.

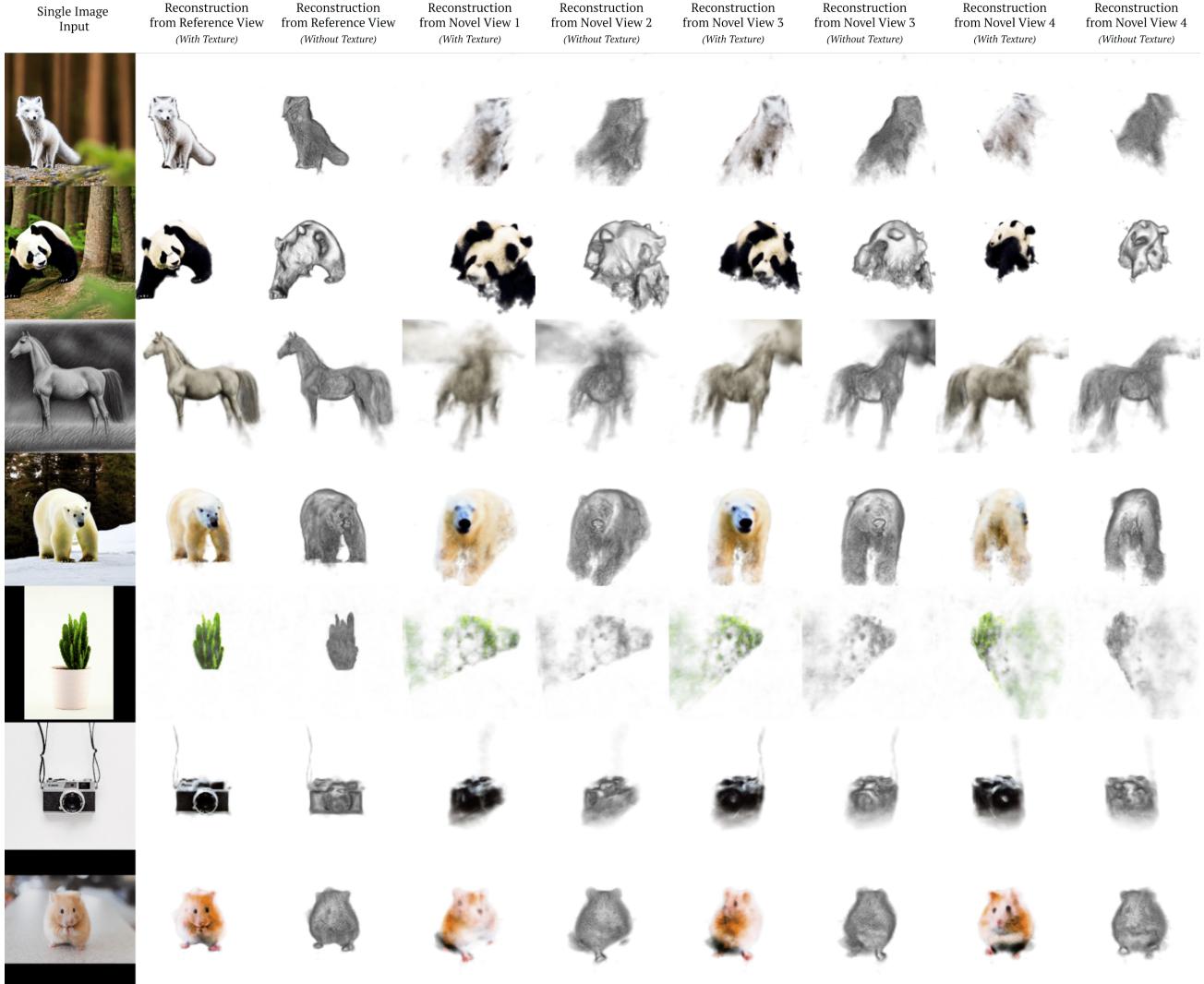


Figure 15. Additional failure cases. This figure presents additional failure cases from our model. The first column shows the input image. The second column shows the reconstruction from the reference viewpoint. The following columns show renders from novel viewpoints, which make clear why these examples are failure cases. Note that some examples (for example, the panda bear in the second row and the hamster in the last row) suffer from the Janus problem.

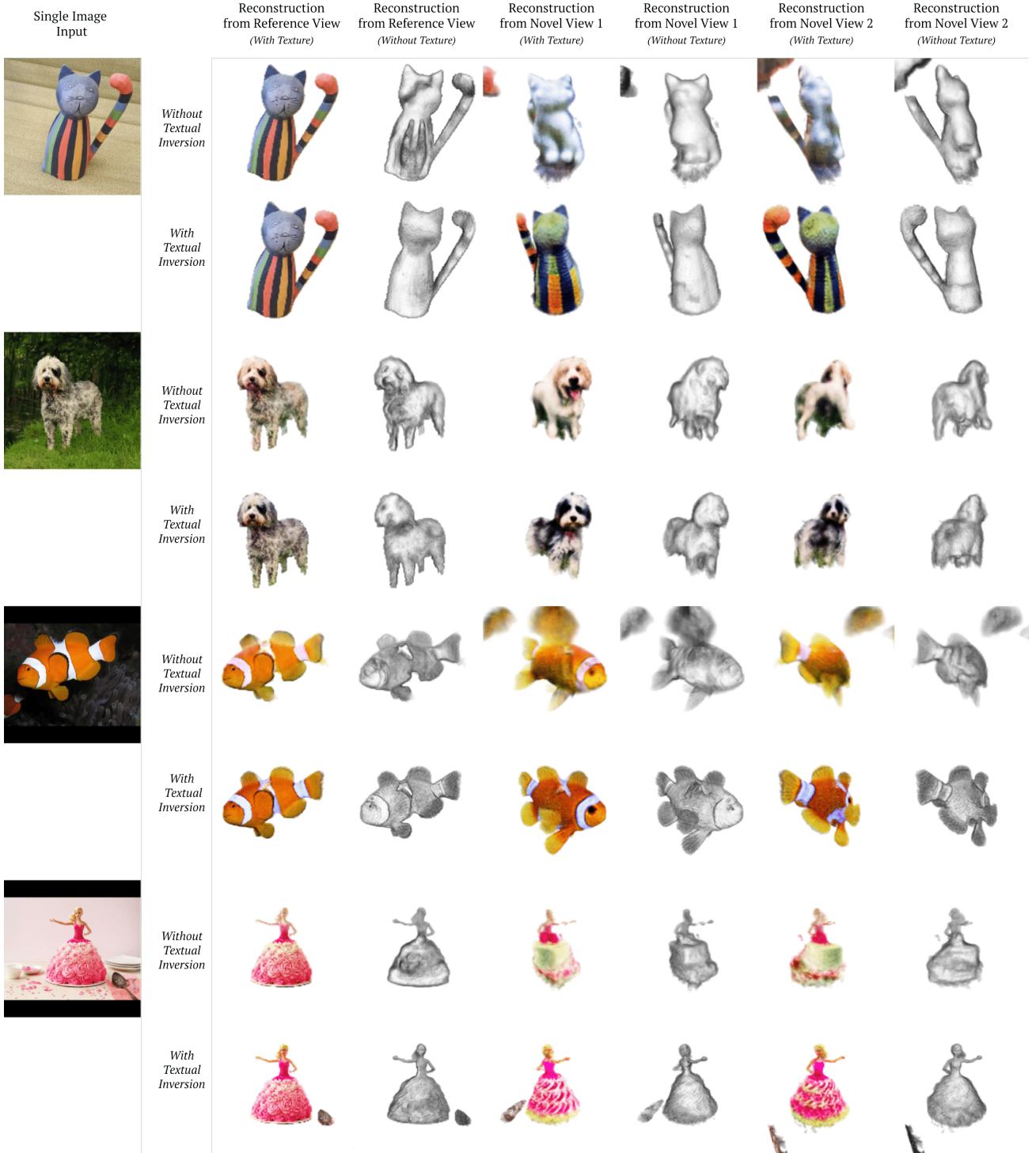


Figure 16. A visualization of the effect of single-image textual inversion on reconstruction quality. An expanded version of Figure 7 in the main paper showing the effect of single-image textual inversion on reconstruction quality. The top row in each pair of rows shows reconstruction results using a standard text prompt, whereas the bottom row shows reconstruction results using single-image textual inversion. The novel views are chosen to show the back side of the object; note how the examples without textual inversion look like highly-generic versions of the objects in the input image.

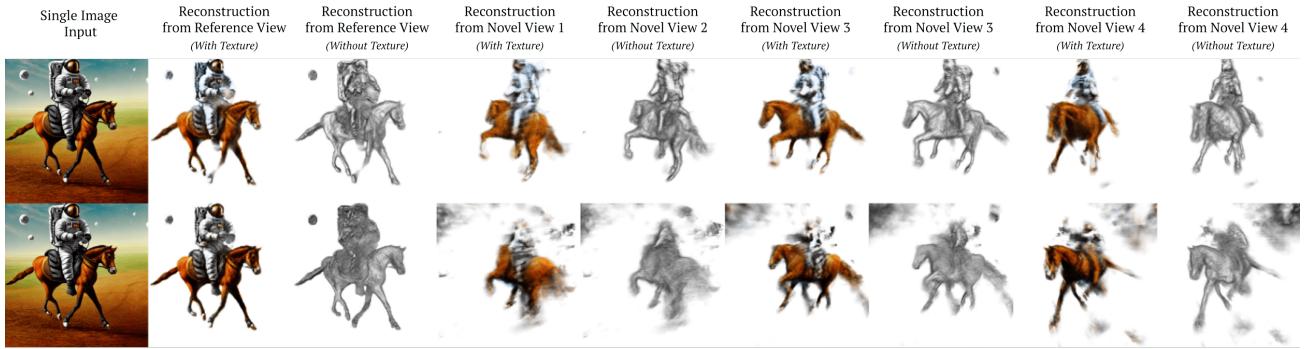


Figure 17. An example of variation across random seeds for a challenging input image. As described in the main paper, our model is able to generate multiple reconstructions for a given input image. For this figure, we apply our method (in a text-to-image-to-3D manner) to a highly challenging image produced by Stable Diffusion from the text prompt “An image of an astronaut riding a horse.” We run reconstruction using two different seeds: one of these (top) yields a reasonable shape, whereas the other is a failure case that does not yield a reasonable shape. This example both highlights the ability of our method to reconstruct highly challenging shapes and also demonstrates how future work could aim to improve reconstruction consistency and quality.