

DreamBooth3D: Subject-Driven Text-to-3D Generation

Amit Raj Srinivas Kaza Ben Poole Michael Niemeyer Nataniel Ruiz Ben Mildenhall
 Shiran Zada Kfir Aberman Michael Rubinstein Jonathan Barron Yuanzhen Li Varun Jampani
Google

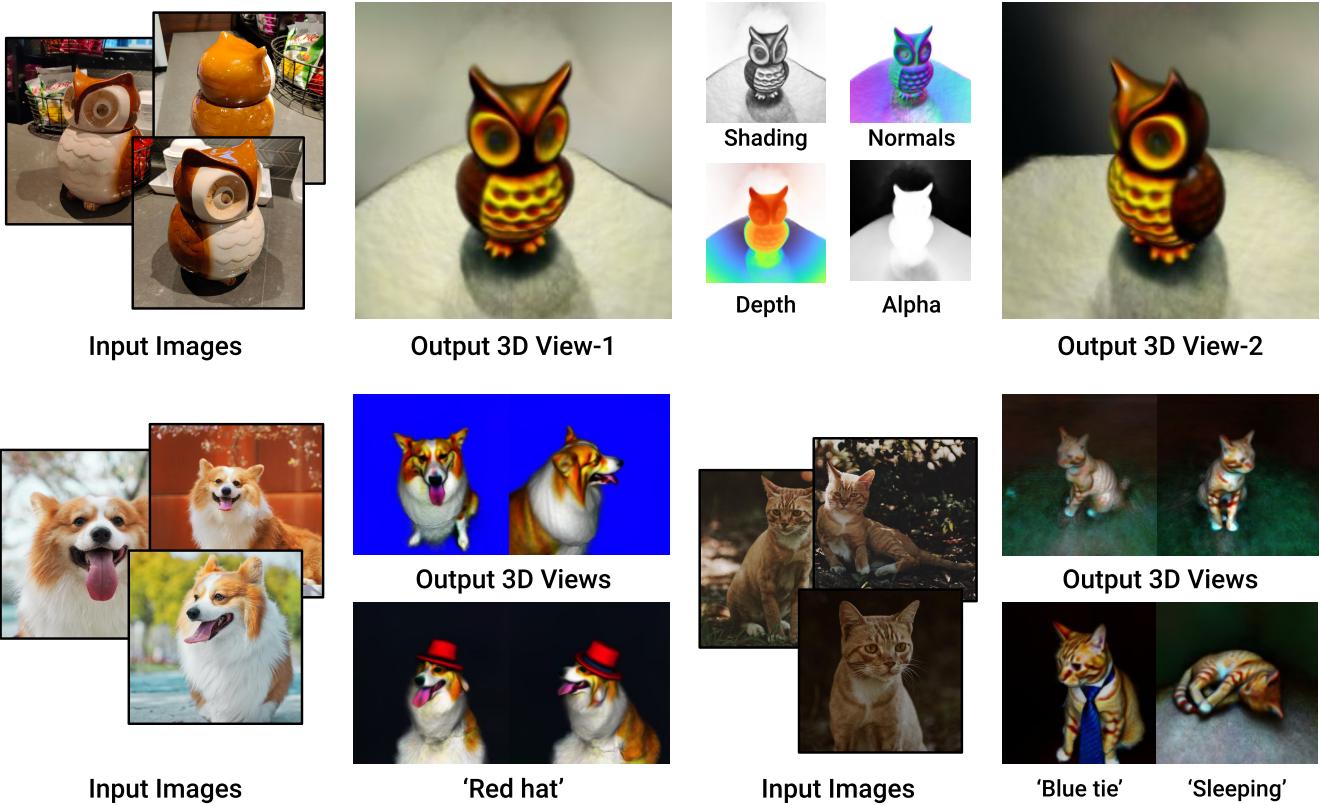


Figure 1: **DreamBooth3D** is a personalized text-to-3D generative model that creates plausible 3D assets of a specific subject from just 3-6 images. Top: 3D output and geometry estimated for an owl object. Bottom: our approach can generate variations of the 3D subject in different contexts (sleeping) or with different accessories (hat or tie) based on a text prompt.

Abstract

We present *DreamBooth3D*, an approach to personalize text-to-3D generative models from as few as 3-6 casually captured images of a subject. Our approach combines recent advances in personalizing text-to-image models (*DreamBooth*) with text-to-3D generation (*DreamFusion*). We find that naively combining these methods fails to yield satisfactory subject-specific 3D assets due to personalized text-to-image models overfitting to the input viewpoints of the subject. We overcome this through a 3-stage optimization strategy where we jointly leverage the 3D consistency of neural radiance fields together with the personalization capability of text-to-image models. Our method

can produce high-quality, subject-specific 3D assets with text-driven modifications such as novel poses, colors and attributes that are not seen in any of the input images of the subject. More results are available at our project page: <https://dreambooth3d.github.io>

1. Introduction

Text-to-Image (T2I) generative models [6, 36, 37, 39] have greatly expanded the ways we can create and edit visual content. Recent works [23, 27, 33, 44] have demonstrated high-quality Text-to-3D generation by optimizing neural radiance fields (NeRFs) [28] using the T2I diffusion models. Such automatic 3D asset creation with input text

prompts alone has applications in a wide range of areas, such as graphics, VR, movies, and gaming.

Although text prompts allow for some degree of control over the generated 3D asset, it is often difficult to precisely control its identity, geometry, and appearance solely with text. In particular, these methods lack the ability to generate 3D assets of a specific subject (e.g., a specific dog instead of a generic dog). Enabling the generation of subject-specific 3D assets would significantly ease the workflow for artists and 3D acquisition. There has been remarkable success [13, 21, 38] in personalizing T2I models for subject-specific 2D image generation. These techniques allow the generation of specific subject images in varying contexts, but they do not generate 3D assets or afford any 3D control, such as viewpoint changes.

In this work, we propose ‘DreamBooth3D’, a method for subject-driven Text-to-3D generation. Given a few (3-6) causal image captures of a subject (without any additional information such as camera pose), we generate subject-specific 3D assets that also adhere to the contextualization provided in the input text prompts. That is, we can generate 3D assets with geometric and appearance identity of a given subject while also respecting the variations (e.g. sleeping or jumping dog) provided by the input text prompt.

For DreamBooth3D, we draw inspiration from the recent works [33] which propose optimizing a NeRF model using a loss derived from T2I diffusion models. We observe that simply personalizing a T2I model for a given subject and then using that model to optimize a NeRF is prone to several failure modes. A key issue is that the personalized T2I models tend to overfit to the camera viewpoints that are only present in the sparse subject images. As a result, the resulting loss from such personalized T2I models is not sufficient to optimize a coherent 3D NeRF asset from arbitrary continuous viewpoints.

With DreamBooth3D, we propose an effective optimization scheme where we optimize both a NeRF asset and T2I model in conjunction with each other to jointly make them subject-specific. We leverage DreamFusion [33] for NeRF optimization and use DreamBooth [38] for T2I model finetuning. Specifically, we propose a 3-stage optimization framework where in the first stage, we partially finetune a DreamBooth model and then use DreamFusion to optimize a NeRF asset. The partially finetuned DreamBooth model does not overfit to the given subject views, but also do not capture all the subject-specific details. So the resulting NeRF asset is 3D coherent, but is not subject-specific. In the second stage, we fully finetune a DreamBooth model to capture fine subject details and use that model to create multiview pseudo-subject images. That is, we translate multiview renderings from the trained NeRF into subject images using the fully-trained DreamBooth model. In the final stage, we further optimize the DreamBooth model us-

ing both the given subject images along with the pseudo multi-view images; which is then used to optimize our final NeRF 3D volume. In addition, we also use a weak reconstruction loss over the pseudo multi-view dataset to further regularize the final NeRF optimization. The synergistic optimization of the NeRF and T2I models prevents degenerate solutions and avoids overfitting of the DreamBooth model to specific views of the subject, while ensuring that the resulting NeRF model is faithful to the subject’s identity.

For experimental analysis, we use the dataset of 30 subjects proposed in DreamBooth [38] which uses the same input setting of sparse casual subject captures. Results indicate our approach can generate realistic 3D assets with high likeness to a given subject while also respecting the contexts present in the input text prompts. Fig. 1 shows sample results of DreamBooth3D on different subjects and contextualizations. When compared to several baselines, both quantitative and qualitative results demonstrate that DreamBooth3D generations are more 3D coherent and better capture subject details.

2. Related Works

Text-to-Image Generation. Earlier works on generative models are dominated by Generative Adversarial Networks (GANs) which train a generator to synthesis images that are indistinguishable from real images [15, 40]. Other generative approaches include autoregressive models that generate images pixel by pixel or patch by patch [12, 47] and masked image models that iteratively predict the marginal distribution of masked patches in the image [6, 7]. Recently, denoising diffusion models [17] have been proposed for image synthesis, which can generate high-quality images by iteratively denoising a noise image toward a clean image [10, 36, 37, 39]. Diffusion models can also be conditioned on various inputs such as depth-map [48], sketch [43], semantic segmentation [1, 37], text [30, 36, 37, 39] and others [18, 22, 48]. For text conditioning, these models take advantage of pre-trained large language models (LLMs) [34, 35] in order to generate images that are aligned with a natural language text prompt given by the user. Motivated by the success of T2I diffusion models, many works utilize pre-trained T2I models for various tasks such as text-based image manipulation [3, 20, 29].

3D Generation. First works on learning-based 3D content generation performed 3D reconstruction from one or multiple images [8, 11, 14, 26, 45]. While leading to good reconstruction results, they require large-scale datasets of accurate 3D data for training which limits their use in real-world scenarios. Another line of work [4, 5, 16, 31, 41] circumvents the need for accurate 3D data by training 3D-aware generative models from image collections. While achieving impressive results, these methods are sensitive to the

assumed pose distribution and restricted to single object classes. Very recently, text-to-3D methods [19, 23, 27, 33] have been proposed that can generate 3D assets from text prompts by utilizing large pretrained T2I diffusion models. In many applications, however, the conditioning are rather input images optionally with text instead of pure text. As a result, multiple works investigate how input images can be incorporated into the optimization pipeline, *e.g.* by applying a reconstruction loss on the input image and predicted monocular depth [9, 46] or a predicted object mask [24]. This, however, limits their use as it does not exploit the full strength of diffusion models, *e.g.*, the object cannot be recontextualized with additional text input. Instead, we propose to not directly reconstruct the input image, but rather the concept of the provided object. This allows not only for reconstruction, but also for recontextualization and more, and the input images do not need to be taken with the same background, lighting, camera *etc.*

Subject-driven Generation. Recent advances in subject-driven image generation [13, 21, 38] enable users to personalize their image generation for specific subjects and concepts. This has provided T2I models with the ability to capture the visual essence of specific subjects and synthesize novel renditions of them in different contexts. DreamBooth [38] accomplishes this by expanding the language-vision dictionary of the model using rare tokens, model finetuning, and a prior preservation loss for regularization. Textual Inversion [13] accomplishes this by optimizing for a new “word” in the embedding space of a pre-trained text-to-image model that represents the input concept. It’s worth noting that these methods do not generate 3D assets or 3D coherent images. There have also been developments in guiding image generation with grounding inputs [22], editing instructions [3], and task-specific conditions such as edges, depth, and surface normals [48]. However, these techniques do not provide personalization to specific subjects, and do not generate 3D assets.

3. Approach

Problem setup.

The input to our approach forms a set of k casual subject captures, each with n pixels, $\{I_i \in \mathbb{R}^{n \times 3}\}$ ($i \in \{1, \dots, k\}$) and a text prompt T for the contextualization or semantic variation (*e.g.*, sleeping vs. standing dog). Our aim is to generate a 3D asset that captures the identity (geometry and appearance) of the given subject while also being faithful to the text prompt. We optimize 3D assets in the form of Neural Radiance Fields (NeRF) [28], which consists of an MLP network \mathcal{M} that encodes radiance fields in a 3D volume. Note that this problem is considerably more under-constrained and challenging compared to a typical 3D reconstruction setting that requires multi-view image captures. We build our technique on recent advances in

T2I personalization and Text-to-3D optimization. Specifically, we use DreamFusion [33] text-to-3D optimization and DreamBooth [38] personalization in our framework, which we briefly review next.

3.1. Preliminaries

DreamBooth T2I Personalization. T2I diffusion models such as Imagen [39], StableDiffusion [37] and DALL-E 2 [36] generate images from any given text prompt. In particular, a T2I diffusion model $\mathcal{D}_\theta(\epsilon, \mathbf{c})$ takes as input an initial noise $\epsilon \sim \mathcal{N}(0, 1)$ and a text embedding $\mathbf{c} = \Theta(T)$ for a given prompt T with a text encoder Θ and generates an image that follows the description of the prompt. The images generated from these T2I models are usually consistent with the prompt, however, it is difficult to exert fine-grained control in the generated images. To that end, DreamBooth [38] proposes a simple yet effective approach to personalize a T2I diffusion model by finetuning the network on a small set of casual captures $\{I_{if}\}$.

Briefly, DreamBooth uses the following diffusion loss function to finetune the T2I model:

$$\mathcal{L}_d = \mathbb{E}_{\epsilon, t} \left[w_t \left\| \mathcal{D}_\theta(\alpha_t I_i + \sigma_t \epsilon, \mathbf{c}) - I_i \right\|^2 \right], \quad (1)$$

where $t \sim \mathcal{U}[0, 1]$ denotes the time-step in the diffusion process and w_t , α_t and σ_t are the corresponding scheduling parameters. Optionally, DreamBooth uses the class prior-preserving loss for improved diversity and to avoid language drift. Refer to [38] for additional details.

DreamFusion optimizes a volume represented as a NeRF \mathcal{M}_ϕ with parameters ϕ so that random views of the volume match a text prompt T using a T2I diffusion model. The learned implicit network \mathcal{M}_ϕ maps from a 3D location to an albedo and density. The normals computed from the gradient of the density are used to randomly relight the model to improve geometric realism with Lambertian shading. Given a random view v , and random lighting direction, we perform volume rendering to output a shaded image \hat{I}_v . To optimize the parameters of the NeRF ϕ so that these images look like a text prompt T , DreamFusion introduced score distillation sampling (SDS) that pushes noisy versions of the rendered images to lower energy states of the T2I diffusion model:

$$\nabla_\phi \mathcal{L}_{SDS} = \mathbb{E}_{\epsilon, t} \left[w_t \left(\mathcal{D}_\theta(\alpha_t \hat{I}_v + \sigma_t \epsilon, \mathbf{c}) - \hat{I}_v \right) \frac{\partial \hat{I}_v}{\partial \phi} \right]. \quad (2)$$

By randomizing over views and backpropagating through the NeRF, it encourages the renderings to look like an image produced by T2I model \mathcal{D}_θ for a given text prompt. DreamFusion proposes to use coarse view-based prompting to optimize NeRF along multiple views. We follow the exact settings used in [33] for all experiments.

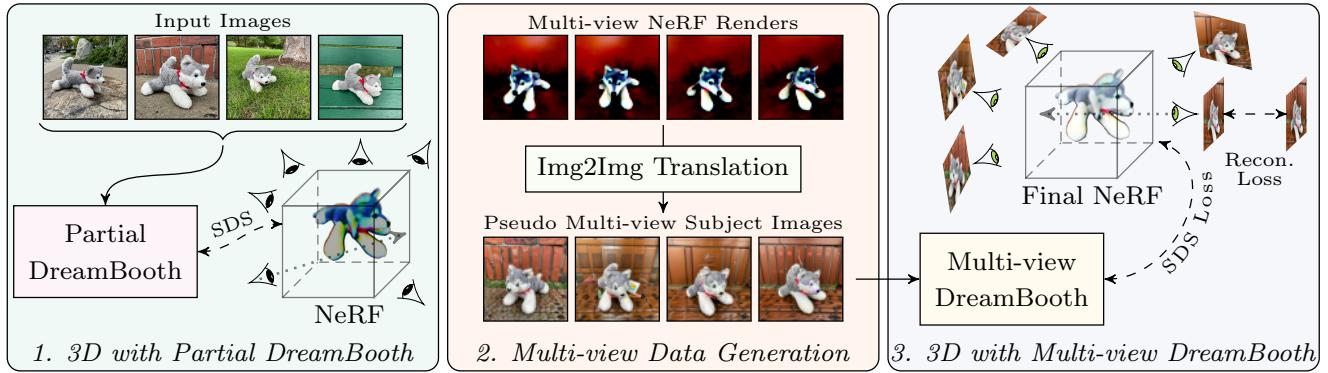


Figure 2: **DreamBooth3D Overview**. In the stage-1 (left), we first partially train a DreamBooth and use the resulting model to optimize the initial NeRF. In stage-2 (middle), we render multi-view images along random viewpoints from the initial NeRF and then translate them into pseudo multi-view subject images using a fully-trained DreamBooth model. In the final stage-3 (right), we further fine-tune the partial DreamBooth using multi-view images and then use the resulting multi-view DreamBooth to optimize the final NeRF 3D asset using the SDS loss along with the multi-view reconstruction loss.

3.2. Failure of Naive Dreambooth+Fusion

A straight-forward approach for subject-driven text-to-3D generation is first personalizing a T2I model and then use the resulting model for Text-to-3D optimization. For instance, doing DreamBooth optimization followed by DreamFusion, which we refer to as DreamBooth+Fusion. Similar baselines are also explored with preliminary experiments in some very recent works such as [23, 27]. However, we find that naive DreamBooth+Fusion technique results in unsatisfactory results as shown in Fig. 3. A key issue we find is that DreamBooth tends to overfit to the subject views that are present in the training views, leading to reduced viewpoint diversity in the image generations. Subject likeness increases with more DreamBooth finetuning steps, while the generated viewpoints get close to that of input exemplar views. As a result, the SDS loss on such a DreamBooth model is not sufficient to obtain a coherent 3D NeRF asset. In general, we observe that the DreamBooth+Fusion NeRF models have same subject views (e.g., face of a dog) imprinted across different viewpoints, a failure mode denoted the “Janus problem” [33].

3.3. Dreambooth3D Optimization

To mitigate the aforementioned issues, we propose an effective multi-stage optimization scheme called DreamBooth3D for subject-driven text-to-3D generation. Fig. 2 illustrates the 3 stages in our approach, which we describe in detail next.

Stage-1: 3D with Partial DreamBooth. We first train a personalized DreamBooth model \hat{D}_θ on the input subject images such as those shown in Fig. 2 (left). Our key observation is that the initial checkpoints of DreamBooth (partially finetuned) T2I models do not overfit to the

given subject views. DreamFusion on such partially fine-tuned DreamBooth models can produce a more coherent 3D NeRF. Specifically, we refer to the partially trained DreamBooth model as $\hat{D}_\theta^{partial}$ and use the SDS loss (Eq. 2) to optimize an initial NeRF asset for a given text prompt as illustrated in Fig. 2 (left). However, the partial DreamBooth model as well as the NeRF asset lack complete likeness to the input subject. We can see this *initial* NeRF output in stage-1 to be a 3D model of the subject class that has partial likeness to the given subject while also being faithful to the given text prompt.

Stage-2: Multi-view Data Generation. This stage forms an important part of our approach, where we make use of 3D consistent initial NeRF together with the fully-trained DreamBooth to generate pseudo multi-view subject images. Specifically, we first render multiple images $\{\hat{I}_v \in \mathbb{R}^{n \times 3}\}$ along random viewpoints $\{v\}$ from the initial NeRF asset resulting in the multi-view renders as shown in Fig. 2 (middle). We then add a fixed amount of noise by running the forward diffusion process from each render to t_{pseudo} , and then run the reverse diffusion process to generate samples using the fully-trained DreamBooth model \hat{D}_θ as in [25]. This sampling process is run independently for each view, and results in images that represent the subject well, and cover a wide range of views due to the conditioning on the noisy render of our initial NeRF asset. However, these images are not multi-view consistent as the reverse diffusion process can add different details to different views, so we call this collection of images *pseudo* multi-view images.

Fig. 2 (middle) shows sample resulting images from this image to image (Img2Img) translation. Some prior works such as [25] use such Img2Img translations for image editing applications. In contrast, we use the Img2Img translation in combination with DreamBooth and NeRF 3D asset

to generate pseudo multi-view subject images. A key insight in this stage is that DreamBooth can effectively generate unseen views of the subject given that initial images are close to those unseen views. In addition, DreamBooth can effectively generate output images with more likeness to the given subject compared to input noisy images. Fig. 2 (middle) shows sample outputs of Img2Img translation with the DreamBooth demonstrating more likeness to the subject images while also preserving the viewpoints of the input NeRF renders.

Stage-3: Final NeRF with Multi-view DreamBooth. The previous stage provides pseudo multi-view subject images $\{I_v^{pseudo}\}$ with *near-accurate* camera viewpoints $\{v\}$. Both the viewpoints as well as the subject-likeness are only approximately accurate due to the stochastic nature of DreamBooth and Img2Img translation. We combine the generated multi-view images $\{I_v^{pseudo}\}$ along with the input subject images $\{I_i\}$ to create a combined data $\mathcal{I}^{aug} = \{I_v^{pseudo}\} \cup \{I_i\}$. We then use this data to optimize our final DreamBooth model followed by a final NeRF 3D asset.

More concretely, we further finetune the partially trained DreamBooth $\hat{\mathcal{D}}_\theta^*$ from stage-1 using this augmented data resulting in a DreamBooth we refer to as Multi-view DreamBooth $\hat{\mathcal{D}}_\theta^{multi}$. We then use this $\hat{\mathcal{D}}_\theta^{multi}$ model to optimize NeRF 3D asset using the DreamFusion SDS loss (Eq.2). This results in a NeRF model with considerably better subject-identity as the multi-view DreamBooth has better view generalization and subject preservation compared to the partial DreamBooth from stage-1.

In practice, we observe that the resulting NeRF asset, optimized only using SDS loss, usually has good geometry-likeness to the given subject but has some color saturation artifacts. To account for the color shift we introduce a novel weak reconstruction loss using our psuedo multi-view images $\{I_v^{pseudo}\}$. In particular, since we know the camera parameters $\{P_v\}$ from which these images were generated, we additionally regularize the training of the second NeRF MLP \mathcal{F}_γ , with γ parameters with the reconstruction loss:

$$\mathcal{L}_{recon} = \|\Gamma(\mathcal{F}_\gamma, P_v) - I_v^{pseudo}\|_p, \quad (3)$$

Where $\Gamma(\mathcal{F}_\gamma, P_v)$ is the rendering function that renders an image from the NeRF \mathcal{F}_γ along the camera viewpoint P_v . This loss serves the dual purpose of pulling the color distribution of the generated volume closer to those of the image exemplars and to improve subject likeness in unseen views. Fig. 2 (right) illustrate the optimization of final NeRF with SDS and multi-view reconstruction losses. The final NeRF optimization objective is given as:

$$\mathcal{L} = \lambda_{recon} \mathcal{L}_{recon} + \lambda_{SDS} \mathcal{L}_{SDS} + \lambda_{nerf} \mathcal{L}_{nerf}, \quad (4)$$

where \mathcal{L}_{nerf} denotes the additional NeRF regularizations used in Mip-NeRF360 [2]. See the supplementary material for additional details of the DreamBooth3D optimization.

4. Experiments

Implementation Details. We use the Imagen [39] T2I model in our experiments. The Imagen model uses the T5-XXL [35] language model for text encoding. On the NeRF side, we use DreamFusion [33]. Our model takes around 3 hours per prompt to complete all the 3 stages of the optimization on a 4 core TPUs. We use a fixed 150 iterations to train the partial DreamBooth model $\hat{\mathcal{D}}_\theta^{partial}$. For the full DreamBooth $\hat{\mathcal{D}}_\theta$ training, we use 800 iterations, which we find to be optimal across different subjects. We render 20 images uniformly sampled at a fixed radius from the origin for pseudo multi-view data generation. We finetune the partially trained $\hat{\mathcal{D}}_\theta^*$ for additional 150 iterations in Stage 3. Refer to the supplementary material for more hyperparameter details.

Datasets. We train our personalized text to 3D models on the image collections released by the authors of [38]. This dataset consists of 30 different image collections with 4-6 casual captures of a wide variety of subjects (dogs, toys, backpack, sunglasses, cartoon etc.). We additionally capture few images of some rare objects (like “owl showpiece” in Fig. 4) to analyze performance on rare objects. Further, we optimize each 3D model on 3–6 prompts to demonstrate 3D contextualizations.

Baselines. We consider two main baselines for comparisons. Latent-NeRF [27] which learns a 3D NeRF model on a latent feature space instead of in RGB pixel space, using an SDS loss in the latent space of Stable Diffusion [37]. As a baseline, we run Latent-NeRF using the fully dreamboothed T2I model and refer to it as “Latent-NeRF” or “L-NeRF” in our experiments. We further compare against a single stage DreamFusion+DreamBooth approach where we first train a DreamBooth diffusion model followed by 3D NeRF optimization using DreamFusion. We refer to our results as “DreamBooth3D” or “DB3D” in the experiments.

Evaluation Metrics. We evaluate our approach with the CLIP R-Precision metric, which measures how accurately we can retrieve a text prompt from an image [32]. Similar to [33], we compute the average CLIP R-Precision over 160 evenly spaces azimuth renders at a fixed elevation of 40 degrees. The CLIP models used for evaluation are the CLIP ViT-B/16, ViT-B/32, and ViT-L-14 models. Since these CLIP metrics can only approximately capture the quality and subject-fidelity of the generated 3D assets, we additionally perform user studies comparing different results.

4.1. Results

Visual Results. Fig. 1 shows sample visual results of our approach along with different semantic variations and contextualizations. Results demonstrate high-quality geometry estimation with DreamBooth3D for even our uncommon owl object. Contextualization examples demonstrate



Figure 3: **Visual Results** on 5 different subjects with two baseline techniques of Latent-NeRF and DreamBooth+Fusion along with those of our technique (DreamBooth3D). Results clearly indicate better 3D consistent results with our approach compared to either of the baseline techniques. See the supplement for additional visualizations and videos.



Figure 4: **Initial vs. Final NeRF Estimates.** Sample multi-view results show that the initial NeRF obtained after stage-1 has only a partial likeness to the given subject whereas the final NeRF from stage-3 of our pipeline has better subject-identity.

that DreamBooth3D faithfully respects the context present in the input text prompt. Fig. 3 shows sample results of our approach in comparison to those of Latent-NeRF and DreamBooth+Fusion baselines. Even though Latent-NeRF works reasonably well in some cases (such as rubber duck in Fig. 3), more often it fails to converge to a coherent 3D model with reasonable shapes. In several cases, Dream-

Booth+Fusion usually produces the 3D assets with Janus problem (same appearance and geometry imprinted across different view angles). DreamBooth3D, on other hand, consistently produces 360° consistent 3D assets while capturing both the geometric and appearance details of the given subject.

Quantitative Comparisons. Table. 1 shows CLIP R-

precision metrics for naive DreamBooth+Fusion (as baseline) and our DreamBooth3D generations. Results clearly demonstrate significantly higher scores for the DreamBooth3D results indicating better 3D consistency and text-prompt alignment of our results.

Initial vs. Final NeRF. Fig. 4 shows sample initial and final NeRF results generated after stages 1 and 3 of our pipeline. As the visual results illustrate, initial-NeRFs only have partial likeness to the given subject, but are consistent in 3D. The final NeRFs from the stage-3 has better likeness to the given subject while retaining the consistent 3D structure. These examples demonstrate the need for the 3-stage optimization in DreamBooth3D.

User Study. We conduct pairwise user studies comparing DreamBooth3D to baselines in order to evaluate our method under three axes: (1) Subject fidelity, where users are asked to answer the question “Which 3D item looks more like the original subject?”; (2) 3D consistency and plausibility where users answer “Which 3D item has a more plausible and consistent geometry?” and (3) Prompt fidelity to the input prompts where users answer “Which video best respects the provided prompt?”. Users can choose either our method or the baseline, or a third option “Cannot determine / both equally”. For the first two user studies on 3D consistency and subject fidelity we compare rotating video results, one for each of the 30 subjects in the dataset and ask 11 users to vote for each pair. For the prompt fidelity study, we generate videos for 54 unique prompt and subject pairs and ask 21 users to respond. We compute final results using majority voting and present them in Figure 5. We find that DreamBooth3D is significantly preferred over the baselines in terms of 3D consistency, subject fidelity as well as prompt fidelity.

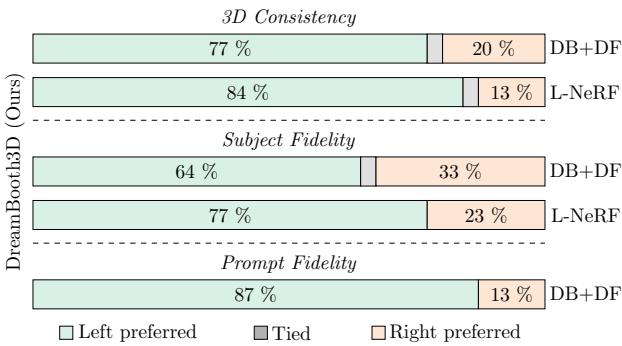


Figure 5: **User Study.** Users show a significant preference for our DreamBooth3D over DB+DF and L-NeRF for 3D consistency, subject fidelity and prompt fidelity.

4.2. Sample Applications

DreamBooth3D can faithfully represent the context present in the text prompts while also preserving the subject

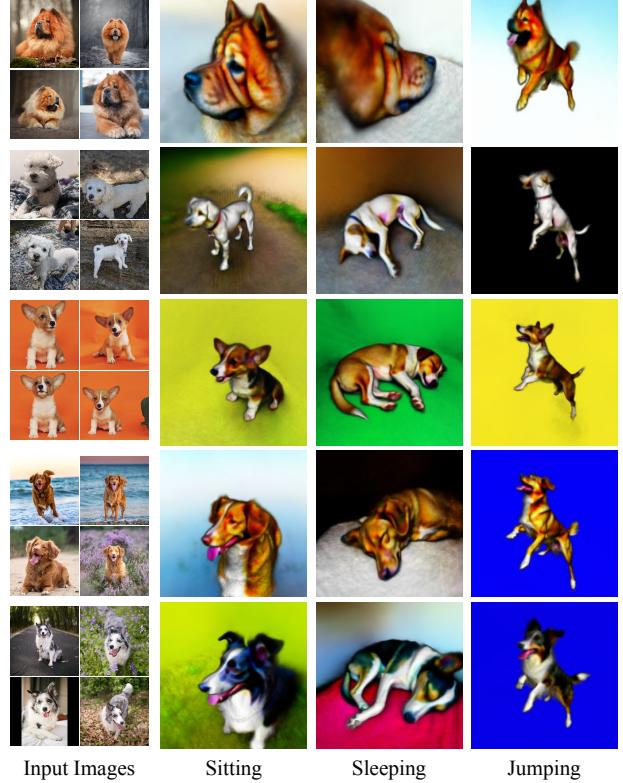


Figure 6: **3D Recontextualizations with DreamBooth3D.** With simple edits in the text prompt, we can generate non-rigid 3D articulations and deformations that correspond to the semantics of the input text. Visuals show consistent contextualization of different dogs in different contexts of sitting, sleeping and jumping. See the supplement for videos.

identity. With simple changes in the text-prompt, DreamBooth3D enables many interesting 3D applications, several of which would otherwise require tedious manual effort to tackle using traditional 3D modeling techniques.

Recontextualization. Fig. 6 shows sample results on different dog subjects, where we recontextualize the 3D dog models with simple prompts of sitting, sleeping and jumping. As the visuals demonstrate, the corresponding 3D models consistently respect the given context in the text prompt across all the subjects. In addition, the 3D articulations and local deformations in the output 3D models are highly realistic even though several of these poses are unseen in the input subject images.

Color/Material Editing. Fig. 7 shows sample color editing results, where a pink backpack can be converted into a blue or green backpack with simple text prompts like ‘a [v] blue backpack’. Similarly, one could also easily edit the material appearance of the 3D asset (for e.g., metal can to wooden can). Refer to the supplementary material for more color

	ViT-B/16↑	ViT-B/32↑	ViT-L-14↑
DreamBooth+Fusion	0.509	0.490	0.506
DreamBooth3D (Ours)	0.783	0.710	0.797

Table 1: **Quantitative comparisons** using CLIP R-precision on DreamBooth+Fusion (baseline) and DreamBooth3D generations indicate that renderings from our 3D model outputs more accurately resemble the text prompts.

and material editing results.

Accessorization. Fig. 7 shows sample accessorization results on a cat subject, where we put on a tie or a suit into the 3D cat model output. Likewise, one can think of other accessorizations like putting on a hat or sunglasses etc.

Stylization. Fig. 7 also shows sample stylization results, where a cream colored shoe is stylized based on color and the addition of frills.

Cartoon-to-3D. A rather striking result we find during our experiments is that DreamBooth3D can even convert non-photorealistic subject images such as 2D flat cartoon images into plausible 3D shapes. Fig. 7 shows a sample result where the resulting 3D model for the red cartoon character is plausible, even though all the images show the cartoon only from the front. Refer to the supplementary material for more qualitative results on different applications.

4.3. Limitations

While our method allows for high-quality 3D asset creation of a given subject and improves over prior work, we observe several limitations. First, the optimized 3D representations are sometimes oversaturated and oversmoothed, which is partially caused by SDS-based optimization with high guidance weighting [33]. This is also a result of being restricted to a relatively low image resolution of 64×64 pixels. Improvements in the efficiency of both diffusion models and neural rendering will potentially allow for scaling to higher resolutions. Furthermore, the optimized 3D representations can sometimes suffer from the Janus problem of appearing to be front-facing from multiple inconsistent viewpoints if the input images do not contain any viewpoint variations. Finally, our model sometimes struggles to reconstruct thin object structures like sunglasses. Fig. 8 shows a couple of failure results.

5. Conclusion

In this paper, we have proposed DreamBooth3D, a method for subject-driven text-to-3D generation. Given a few (3-6) casual image captures of a subject (without any additional information such as camera pose), we generate subject-specific 3D assets that also adhere to the contextualization provided in the input text prompts (e.g. sleep-

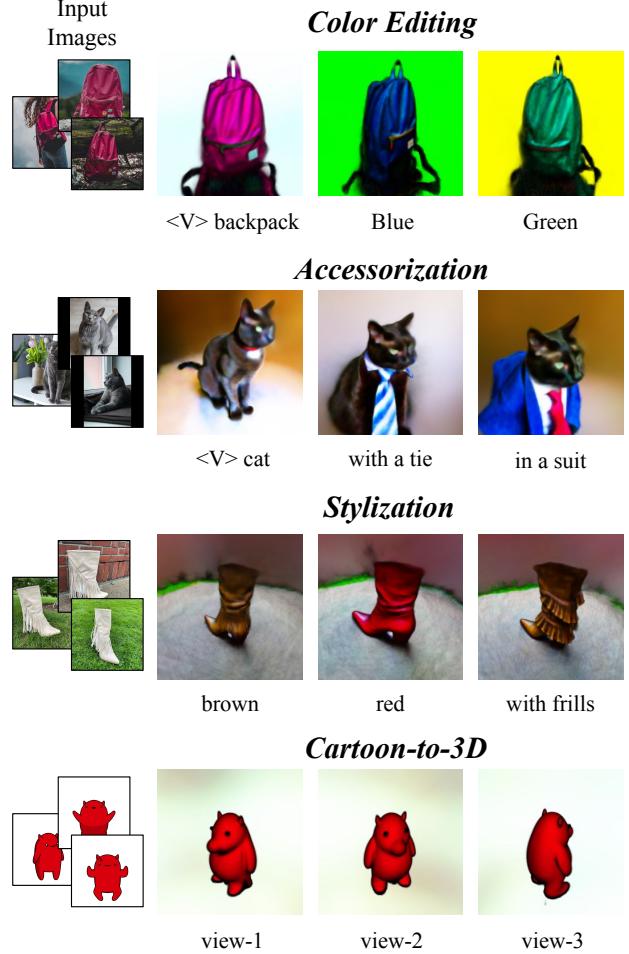


Figure 7: **Sample Applications.** DreamBooth3D’s subject preservation and faithfulness to the text prompt enables several applications such as color/material editing, accessorization, stylization, *etc.* DreamBooth3D can even produce plausible 3D models from unrealistic cartoon images. See the supplemental material for videos.

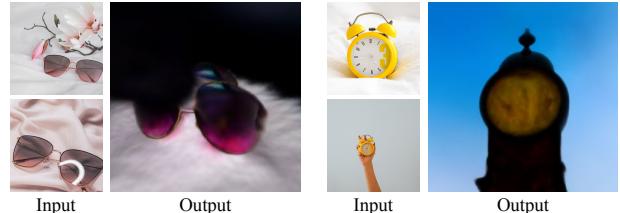


Figure 8: **Sample Failure Cases.** We observe DreamBooth3D often fails to reconstruct thin object structures like sunglasses, and sometimes fails to reconstruct objects with not enough view variation in the input images.

ing, jumping, red, etc.). Our extensive experiments on the DreamBooth dataset [38] have shown that our method can generate realistic 3D assets with high likeness to a given subject while also respecting the contexts present in the input text prompts. Our method outperforms several baselines in both quantitative and qualitative evaluations. In the future, we plan to continue to improve the photorealism and controllability of subject-driven 3D generation.

References

- [1] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation. *arXiv*, 2023. [2](#)
- [2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *CVPR*, 2021. [5](#), [11](#)
- [3] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to Follow Image Editing Instructions. *CVPR*, 2023. [2](#), [3](#)
- [4] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. *CVPR*, 2021. [2](#)
- [5] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. *CVPR*, 2022. [2](#)
- [6] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. *arXiv*, 2023. [1](#), [2](#)
- [7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. *CVPR*, 2022. [2](#)
- [8] Christopher Bongsoo Choy, Danfei Xu, Jun Young Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. *ECCV*, 2016. [2](#)
- [9] Congyue Deng, Chiyu "Max" Jiang, Charles R. Qi, Xincheng Yan, Yin Zhou, Leonidas J. Guibas, and Dragomir Anguelov. NeRDi: Single-View NeRF Synthesis with Language-Guided Diffusion as General Image Priors. *arXiv*, 2022. [3](#)
- [10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. [2](#)
- [11] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *CVPR*, 2017. [2](#)
- [12] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. *ECCV*, 2022. [2](#)
- [13] 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*, 2022. [2](#), [3](#)
- [14] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. *ICCV*, 2019. [2](#)
- [15] 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*, 2020. [2](#)
- [16] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A Style-based 3D Aware Generator for High-Resolution Image Synthesis. *ICLR*, 2022. [2](#)
- [17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. [2](#)
- [18] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv*, 2023. [2](#)
- [19] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-Shot Text-Guided Object Generation with Dream Fields. *CVPR*, 2022. [3](#)
- [20] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv*, 2022. [2](#)
- [21] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. *arXiv*, 2022. [2](#), [3](#)
- [22] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-Set Grounded Text-to-Image Generation. *arXiv*, 2023. [2](#), [3](#)
- [23] 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*, 2022. [1](#), [3](#), [4](#)
- [24] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Realfusion: 360 reconstruction of any object from a single image. *arXiv*, 2023. [3](#)
- [25] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. *ICLR*, 2022. [4](#)
- [26] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. *CVPR*, 2019. [2](#)
- [27] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures. *arXiv*, 2022. [1](#), [3](#), [4](#), [5](#)
- [28] 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. *ECCV*, 2020. [1](#), [3](#), [11](#)
- [29] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv*, 2022. [2](#)
- [30] Alexander Quinn 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. *ICML*, 2022. 2
- [31] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. *CVPR*, 2021. 2
- [32] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. *NeurIPS Datasets and Benchmarks Track*, 2021. 5
- [33] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. *ICLR*, 2023. 1, 2, 3, 4, 5, 8, 11
- [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. *ICML*, 2021. 2
- [35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020. 2, 5
- [36] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv*, 2022. 1, 2, 3
- [37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CVPR*, 2022. 1, 2, 3, 5
- [38] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. *arXiv*, 2022. 2, 3, 5, 9
- [39] 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. *NeurIPS*, 2022. 1, 2, 3, 5
- [40] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. StyleGAN-T: Unlocking the Power of GANs for Fast Large-Scale Text-to-Image Synthesis. *arXiv*, 2023. 2
- [41] Katja Schwarz, Yiyi Liao, and Andreas Geiger. On the frequency bias of generative models. *NeurIPS*, 2021. 2
- [42] 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. 11
- [43] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. *arXiv*, 2022. 2
- [44] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score Jacobian Chaining: Lifting Pretrained 2D Diffusion Models for 3D Generation. *arXiv*, 2022. 1
- [45] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2Mesh++: Multi-View 3D Mesh Generation via Deformation. *ICCV*, 2019. 2
- [46] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. NeuralLift-360: Lifting An In-the-wild 2D Photo to A 3D Object with 360 Views. *arXiv*, 2022. 3
- [47] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunnar Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv*, 2022. 2
- [48] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv*, 2023. 2, 3

A. Summary Video with Visual Results

We summarize our findings in a video, which outlines the three-stage DreamBooth3D method and includes a comparison to the baselines. We also show how our approach compares to other approaches via a user study. Finally, several example applications are shown, including material editing, accessorization, color changes, and pose changes.

B. NeRF Details

We use Mip-NeRF [2] as our choice of volumetric representation. Particularly, to render the color of a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ cast into the scene, Mip-NeRF divides the ray into intervals and for each interval calculates the mean and variance (μ, Σ) of a conical frustum corresponding to the interval. These values are then used to encode the ray using integrated positional encoding

$$\gamma(\mu, \Sigma) = \left\{ \begin{bmatrix} \sin(2^l \mu) \exp(-2^{(2l-1)} \text{diag}(\sigma)) \\ \sin(2^l \mu) \exp(-2^{(2l-1)} \text{diag}(\sigma)) \end{bmatrix} \right\}_{l=0}^L \quad (5)$$

The learnt volume \mathcal{N}_ϕ is then used to generate albedo \mathbf{c} and opacity σ .

$$\mathbf{c}, \sigma = \mathcal{N}_\phi(\gamma(\mu, \Sigma))$$

The final color is then calculated using numerical quadrature as in [28]. As in [33], we define $\Sigma = \lambda_t^2 I$, where, λ_t is annealed from a high to low value, to gradually introduce higher frequency components during the optimization. The NeRF volume is regularized using the orientation loss introduced in Ref-NeRF [42], to encourage better geometry. Particularly,

$$\mathcal{L}_{ori} = \sum_i \text{stop_grad}(w_i) \max(0, \mathbf{n}_i \cdot \mathbf{v}) \quad (6)$$

Where \mathbf{n}_i is the normal direction at a point, w_i are rendering weights as defined in [28] and \mathbf{v} is the lighting direction. An additional opacity loss is used to encourage foreground/background separation

$$\mathcal{L}_{op} = \sqrt{\left(\sum_i w_i\right)^2 + 0.01} \quad (7)$$

The final NeRF regularization loss is then given by:

$$\mathcal{L}_{nerf} = \mathcal{L}_{op} + \mathcal{L}_{ori} \quad (8)$$

C. Additional results

Fig. 9 provides additional results with associated depths, normals and alpha maps to demonstrate the 3D consistency of our results on a variety of subjects. Fig. 10 shows multiple views of the assets rendered for the same subject with different text prompts.

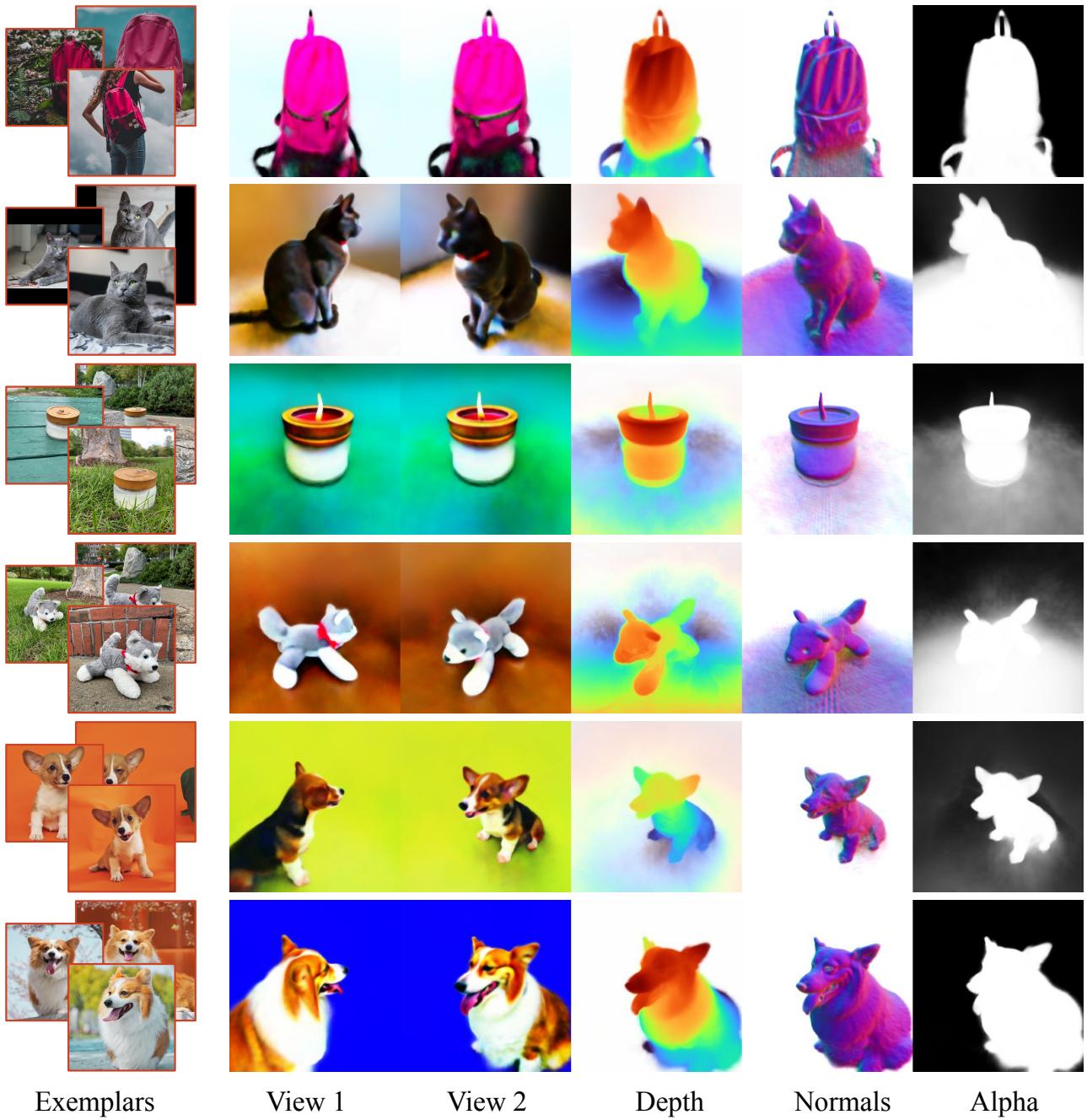


Figure 9: **Additional Results.** Dreambooth3D can produce 3D consistent volumes from text prompts. The figure shows generated assets for the base prompt "A photo of <v>" where <v> is the subject presented in the first column. Column 2 and 3 shows two different views of the rendered volume. Column 3,4 and 5 shows the depth, normals and opacity of the second view respectively.



Figure 10: **Additional Results.** Dreambooth3D is capable of a number of accessorization, composition, material editing tasks through text prompting. An example of this form of prompting is "A photo of <v> wearing a green umbrella". Row 1 shows the rendered geometry and normals of the base subject, and the subsequent rows show material-edited, composited, or accessorized variants. Row 2 demonstrates a material edit to change the dog into a stone statue. Row 3 composites a rainbow carpet into the scene. Row 4 adds a green umbrella.