SV3D: Novel Multi-view Synthesis and 3D Generation from a Single Image using Latent Video Diffusion

Vikram Voleti*, Chun-Han Yao*, Mark Boss*, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani

Stability AI

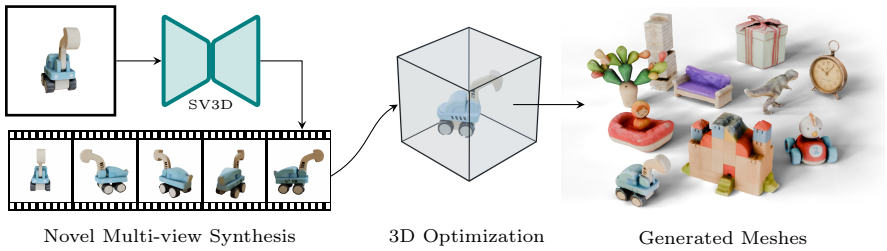

Fig. 1: Stable Video 3D (SV3D). From a single image, SV3D generates consistent novel multi-view images. We then optimize a 3D representation with SV3D-generated views resulting in high-quality 3D meshes.

Abstract. We present Stable Video 3D (SV3D) — a latent video diffusion model for high-resolution, image-to-multi-view generation of orbital videos around a 3D object. Recent works propose to adapt 2D generative models for novel view synthesis (NVS) and 3D optimization. However, these methods have several disadvantages due to limited views or inconsistent NVS, affecting the performance of 3D object generation. In this work, we propose SV3D that adapts image-to-video diffusion model for novel multi-view synthesis and 3D generation, thereby leveraging the generalization and multi-view consistency of the video models, while further adding explicit camera control for NVS. We also propose improved 3D optimization techniques for image-to-3D generation using SV3D and its NVS outputs. Extensive experiments on multiple datasets with 2D and 3D metrics and user study demonstrate SV3D's state-of-the-art performance on NVS as well as 3D reconstruction compared to prior works.

Keywords: 3D synthesis · Video generation · Novel view synthesis

1 Introduction

Single-image 3D object reconstruction is a long-standing problem in computer vision with a wide range of applications in game design, AR/VR, e-commerce, robotics, etc. It is a highly challenging and ill-posed problem as it requires lifting 2D pixels to 3D space while also reasoning about the unseen portions of the object in 3D.

Despite being a long-standing vision problem, it is only recently that practically useful results are being produced by leveraging advances in the generative AI. This is mainly made possible by the large-scale pretraining of generative models which enables sufficient generalization to various domains. A typical strategy is to use image-based 2D generative models (*e.g.*, Imagen [42], Stable Diffusion (SD) [40]) to provide a 3D optimization loss function for unseen novel views of a given object [23,31,37]. In addition, several works repurpose these 2D generative models to perform novel view synthesis (NVS) from a single image [26,27,30,54], and then use the synthesized novel views for 3D generation. Conceptually, these works mimic a typical photogrammetry-based 3D object capture pipeline, *i.e.*, first photographing multi-view images of an object, followed by 3D optimization; except that the explicit multi-view capture is replaced by novel-view synthesis using a generative model, either via text prompt or camera pose control.

A key issue in these generation-based reconstruction methods is the lack of multi-view consistency in the underlying generative model, resulting in inconsistent novel views. Some works try to address the this by jointly reasoning a 3D representation during NVS [6, 15, 27], but at the cost of high compute and data, often still resulting in inconsistent geometric and texture details. In this work, we tackle this issue by adapting a high-resolution, image-conditioned video diffusion model for NVS followed by 3D generation.

Novel Multi-view Synthesis. We adapt a latent video diffusion model (Stable Video Diffusion - SVD [2]) to generate multiple novel views of a given object with *explicit camera pose conditioning*. SVD demonstrates excellent *multi-view consistency* for video generation, and we repurpose it for NVS. In addition, SVD also has good *generalization* capabilities as it is trained on large-scale image and video data that are more readily available than large-scale 3D data. In short, we adapt the video diffusion model for NVS from a single image with three useful properties for 3D object generation: *pose-controllable, multi-view consistent, and generalizable*. We call our resulting NVS network 'SV3D'. To our knowledge, this is the first work that adapts a video diffusion model for explicit pose-controlled view synthesis. Some contemporary works such as [2, 29] demonstrate this, but only generate orbital videos without any explicit camera control.

3D Generation. We then utilize our SV3D model for 3D object generation by optimizing a NeRF and DMTet mesh in a coarse-to-fine manner. Benefiting from the multi-view consistency in SV3D, we are able to produce high-quality 3D meshes directly from the SV3D novel view images. We also design a masked score distillation sampling (SDS) [37] loss to further enhance 3D quality in the regions that are not visible in the SV3D-predicted novel views. In addition, we propose to jointly optimize a disentangled illumination model along with 3D shape and texture, effectively reducing the issue of baked-in lighting.

We perform extensive comparisons of both our NVS and 3D generation results with respective state-of-the-art methods, demonstrating considerably better outputs with SV3D. For NVS, SV3D shows high-level of multi-view consistency and generalization to real-world images while being pose controllable. Our resulting 3D meshes are able to capture intricate geometric and texture details.

2 Background

2.1 Novel View Synthesis

We organize the related works along three crucial aspects of novel view synthesis (NVS): generalization, controllability, and multi-view (3D) consistency.

Generalization. Diffusion models [14, 48] have recently shown to be powerful generative models that can generate a wide variety of images [3, 40, 41] and videos [2, 12, 53] by iteratively denoising a noise sample. Among these models, the publicly available Stable Diffusion (SD) [40] and Stable Video Diffusion (SVD) [2] demonstrate strong generalization ability by being trained on extremely large datasets like LAION [43] and LVD [2]. Hence, they are commonly used as foundation models for various generation tasks, *e.g.* novel view synthesis.

Controllability. Ideally, a controllable NVS model allows us to generate an image corresponding to any arbitrary viewpoint. For this, Zero123 [26] repurposes an image diffusion model to a novel view synthesizer, conditioned on a single-view image and the pose difference between the input and target views. Follow-up works such as Zero123XL [8] and Stable Zero123 [49] advance the quality of diffusion-based NVS, as well as the trained NeRFs using SDS loss. However, they only generate one novel view at a time, and thus are not designed to be multi-view consistent inherently. Recent works such as EscherNet [21] and Free3D [64] are capable of better multi-view consistency with intelligent camera position embedding design. However, they only use image-based diffusion models, and generate images at 256×256 resolution. We finetune a video diffusion model to generate novel views at 576×576 resolution.

Multi-view Consistency. Multi-view consistency is the most critical requirement for high-quality NVS and 3D generation. To address the inconsistency issue, MVDream [46], SyncDreamer [27], HexGen3D [30], and Zero123++ [45] propose to generate multiple (specific) views of an object simultaneously. However, they are not controllable: given a conditional image, they only generate specific views, not arbitrary viewpoints. Moreover, they were finetuned from image-based diffusion models, *i.e.* multi-view consistency was imposed by adding interaction among the multiple generated views through cross-attention layers. Hence, their output quality is limited to the generalizability of their base image-based model, and their 3D finetuning dataset. Efficient-3DiM finetunes the SD model with a stronger vision transformer DINO v2 [36]. Consistent-1-to-3 [59] and SPAD [18] leverage epipolar geometry. One-2-3-45 [25] and One-2-3-45++ [24] train additional 3D network using the 2D generator's outputs. MVDream [46], Consistent123 [55], and Wonder3D [28] also train multi-view diffusion models, yet still require post-processing for video rendering. SyncDreamer [27] and ConsistNet [57] employ 3D representation into the latent diffusion model.

Exploiting Video Diffusion Models. To achieve better generalization and multi-view consistency, some contemporary works exploit the temporal priors in video diffusion models for NVS. For instance, Vivid-1-to-3 [22] combines a view-conditioned diffusion model and video diffusion model to generate consistent views. SVD-MV [2] and IM-3D [29] finetune a video diffusion model for NVS.

However, they generate $\leq 360^\circ$ views at the same elevation only. Unlike SV3D, none of them are capable of rendering any arbitrary view of the 3D object.

We argue that the existing NVS and 3D generation methods do not fully leverage the superior generalization capability, controllability, and consistency in video diffusion models. In this paper, we fill this important gap and train SV3D, a state-of-the-art novel multi-view synthesis video diffusion model at 576×576 resolution, and leverage it for 3D generation.

2.2 3D Generation

Recent advances in 3D representations and diffusion-based generative models have significantly improved the quality of image-to-3D generation. Here, we briefly summarize the related works in these two categories.

3D Representation. 3D generation has seen great progress since the advent of Neural Radiance Fields (NeRFs) [32] and its subsequent variants [1], which implicitly represents a 3D scene as a volumetric function, typically parameterized by a neural network. Notably, Instant-NGP [33] introduces a hash grid feature encoding that can be used as a NeRF backbone for fast inference and ability to recover complex geometry. On the other hand, several recent works improve from an explicit representation such as DMTet [44], which is capable of generating high-resolution 3D shapes due to its hybrid SDF-Mesh representation and high memory efficiency. Similar to [23, 38], we adopt coarse-to-fine training for 3D generation, by first learning a rough object with Instant-NGP NeRF and then refining it using the DMTet representation.

Diffusion-Based 3D Generation. Several recent works [16, 34] train a 3D diffusion model to to learn these flexible 3D representations, which, however, lack generalizability due to the scarcity of 3D data. To learn 3D generation without ground truth 3D data, image/multi-view diffusion models have been used as guidance. DreamFusion [37] and follow-up works [23,31] leverage a trained image diffusion model as a 'scoring' function and calculate SDS loss for text-to-3D generation. However, they are prone to artifacts like Janus problem [31, 37] and over-saturated texture. Inspired by Zero123 [26], recent works [21, 24, 25, 27, 29, 38, 45, 49, 51, 64] finetune image/video diffusion models to generate novel view images as a stronger guidance. Our method shares the same spirit as this line of work, but produces denser, controllable, and more consistent multi-view images, thus resulting in better 3D generation quality.

3 SV3D: Novel Multi-view Synthesis

Our main idea is to repurpose temporal consistency in a video diffusion model for spatial 3D consistency of an object. Specifically, we finetune SVD [2] to generate an orbital video around a 3D object, conditioned on a single-view image. This orbital video need not be at the same elevation, or at regularly spaced azimuth angles. SVD is well-suited for this task since it is trained to generate smooth and consistent videos on large-scale datasets of real and high-quality videos.

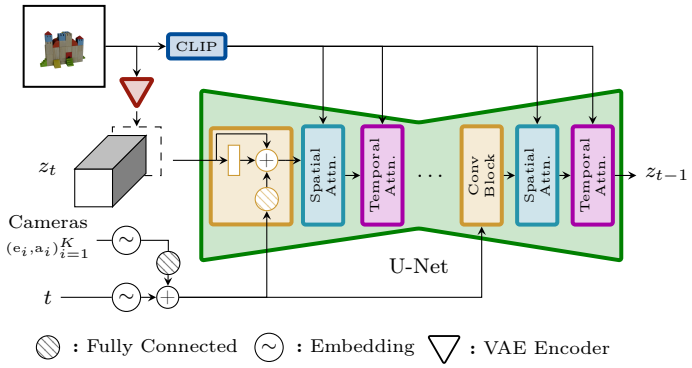

Fig. 2: SV3D Architecture. We add the sinusoidal embedding of the camera orbit elevation and azimuth angles (e, a) to that of the noise step t, and feed the sum to the convolutional blocks in the UNet. We feed the single input image's CLIP embedding to the attention blocks, and concatenate its latent embedding to the noisy state $\mathbf{z}_t$.

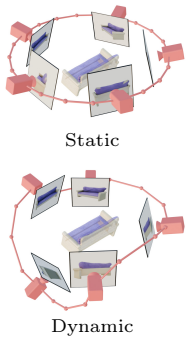

Fig. 3: Static vs. Dynamic Orbits. We use two types of orbits for training the SV3D models.

The exposure to superior data quantity and quality makes it more *generalizable* and *multi-view consistent*, and the flexibility of the SVD architecture makes it amenable to be finetuned for *camera controllability*.

Some prior works attempt to leverage such properties by finetuning image diffusion models, training video diffusion models from scratch, or finetuning video diffusion models to generate pre-defined views at the same elevation (static orbit) around an object [2, 29]. However, we argue that these methods do not fully exploit the potential of video diffusion models. To the best of our knowledge, SV3D is the first video diffusion-based framework for *controllable* multi-view synthesis at 576×576 resolution (and subsequently for 3D generation).

Problem Setting. Formally, given an image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ of an object, our goal is to generate an orbital video $\mathbf{J} \in \mathbb{R}^{K\times 3\times H\times W}$ around the object of $K = 21$ multi-view images along a camera trajectory $\boldsymbol{\pi} \in \mathbb{R}^{K\times 2} = \{(e_i, a_i)\}_{i=1}^K$ as a sequence of K tuples of elevation e and azimuth a angles. We assume the camera always looks at the center of an object (origin of world coordinates), so any viewpoint can be specified by only two parameters: e and a. We generate this orbital video by iteratively denoising samples from a learned conditional distribution $p(\mathbf{J}|\mathbf{I}, \boldsymbol{\pi})$, parameterized by a video diffusion model.

SV3D Architecture. As shown in Fig. 2, the architecture of SV3D builds on that of SVD consisting of a UNet with multiple layers, each layer containing sequences of one residual block with Conv3D layers, and two transformer blocks (spatial and temporal) with attention layers. (i) We remove the vector conditionings of 'fps id' and 'motion bucket id' since they are irrelevant for SV3D. (ii) The conditioning image is concatenated to the noisy latent state input $\mathbf{z}_t$ at noise timestep t to the UNet, after being embedded into latent space by the VAE encoder of SVD. (iii) The CLIP-embedding [39] matrix of the conditioning image is provided to the cross-attention layers of each transformer block as its key and

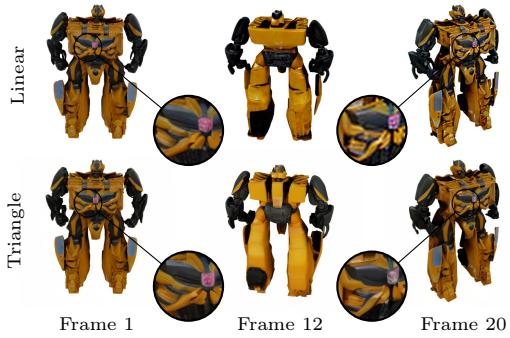

Fig. 4: Linear vs. Triangle CFG Scaling. Notice increased oversharping in the penultimate frame in linear scaling vs. our proposed triangle scaling.

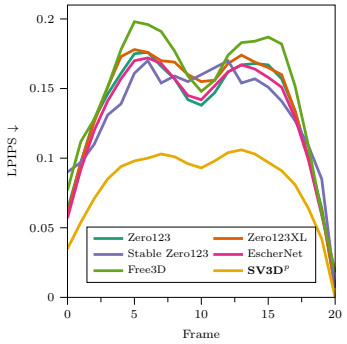

Fig. 5: LPIPS vs. Frame Number. SV3D has the best reconstruction metric per frame.

value, the query being the feature at that layer. (iv) The camera trajectory is fed into the residual blocks along with the diffusion noise timestep. The camera pose angles e_i and a_i and the noise timestep t are first embedded into sinusoidal position embeddings. Then, the camera pose embeddings are concatenated together, linearly transformed, and added to the noise timestep embedding. This is fed to every residual block, where they are added to the block's output feature (after being linearly transformed again to match the feature size).

Static v.s. Dynamic Orbits. As shown in Fig. 3, we design *static* and *dynamic* orbits to study the effects of camera pose conditioning. In a **static** orbit, the camera circles around an object at regularly-spaced azimuths at the same elevation angle as that in the conditioning image. The disadvantage with the static orbit is that we might not get any information about the top or bottom of the object depending on the conditioning elevation angle. In a **dynamic** orbit, the azimuths can be irregularly spaced, and the elevation can vary per view. To build a dynamic orbit, we sample a static orbit, add small random noise to the azimuth angles, and add a random weighted combination of sinusoids with different frequencies to its elevation. This provides temporal smoothness, and ensures that the camera trajectory loops around to end at the same azimuth and elevation as those of the conditioning image.

Thus, with this strategy, we are able to tackle all three aspects of *generalization*, *controllability*, and *consistency* in novel multi-view synthesis by leveraging video diffusion models, additionally conditioning on camera trajectory, and repurposing temporal consistency for spatial 3D object consistency, respectively.

Triangular CFG Scaling. SVD uses a linearly increasing scale for classifier-free guidance (CFG) from 1 to 4 across the generated frames. However, this scaling causes the last few frames in our generated orbits to be over-sharpened, as shown in Fig. 4 Frame 20. Since we generate videos looping back to the front-view image, we propose to use a triangle wave CFG scaling during inference: linearly increase CFG from 1 at the front view to 2.5 at the back view, then linearly decrease it back to 1 at the front view. Fig. 4 also demonstrates that our triangle CFG scaling produces more details in the back view (Frame 12).

Models. We train three image-to-3D-video models finetuned from SVD. First, we train a pose-*u*nconditioned model, SV3D^u, which generates a video of static orbit around an object while only conditioned on a single-view image. Note that unlike SVD-MV [2], we do not provide the elevation angle to the pose-unconditioned model, as we find that the model is able to infer it from the conditioning image. Our second model, the pose-*c*onditioned SV3D^c is conditioned on the input image as well as a sequence of camera elevation and azimuth angles in an orbit, trained on dynamic orbits. Following SVD's [2] intuition to *p*rogressively increase the task difficulty during training, we train our third model, SV3D^p, by first finetuning SVD to produce static orbits unconditionally, then further finetuning on dynamic orbits with camera pose condition.

Training Details. We train SV3D on the Objaverse dataset [9], which contains synthetic 3D objects covering a wide diversity. For each object, we render 21 frames around it on random color background at 576×576 resolution, field-of-view of 33.8 degrees. We choose to finetune the SVD-xt model to output 21 frames. All three models (SV3D^u, SV3D^c, SV3D^p) are trained for 105k iterations in total (SV3D^p is trained unconditionally for 55k iterations and conditionally for 50k iterations), with an effective batch size of 64 on 4 nodes of 8 80GB A100 GPUs for around 6 days. For more training details, please see the appendix.

3.1 Experiments and Results

Datasets. We evaluate the SV3D-generated multi-view images on static and dynamic orbits on the unseen GSO [10] and OmniObject3D [56] datasets. Since many GSO objects are the same items with slightly different colors, we filter 300 objects from GSO to reduce redundancy and maintain diversity. For each object, we render ground truth static and dynamic orbit videos and pick the last frame of each video as the conditioning image. We also conduct a user study on novel view videos from a dataset of 22 real-world images. (More details in appendix.)

Metrics. We use the SV3D models to generate static and dynamic orbit videos corresponding to the ground truth camera trajectories in the evaluation datasets. We compare each generated frame with the corresponding ground truth frames, in terms of Learned Perceptual Similarity (LPIPS [63]), Peak Signal-to-Noise Ratio (PSNR), Structural SIMilarity (SSIM), Mean Squared-Error (MSE), and CLIP-score (CLIP-S). This range of metrics covers both pixel-level as well as semantic aspects. Note that testing on dynamic orbits evaluates the controllability of SV3D models, and all the metrics evaluate multi-view consistency.

Baselines. We compare SV3D with several recent NVS methods capable of generating arbitrary views, including Zero123 [26], Zero123-XL [8], SyncDreamer [27], Stable Zero123 [49], Free3D [64], EscherNet [21].

Results. As shown in Tabs. 1 to 4, our SV3D models achieve state-of-the-art performance on novel multi-view synthesis. Tabs. 1 and 3 show results on static orbits, and include all our three models. We see that even our pose-unconditioned model SV3D^u performs better than all prior methods. Tabs. 2 and 4 show results on dynamic orbits, and include our pose-conditioned models SV3D^c and SV3D^p.

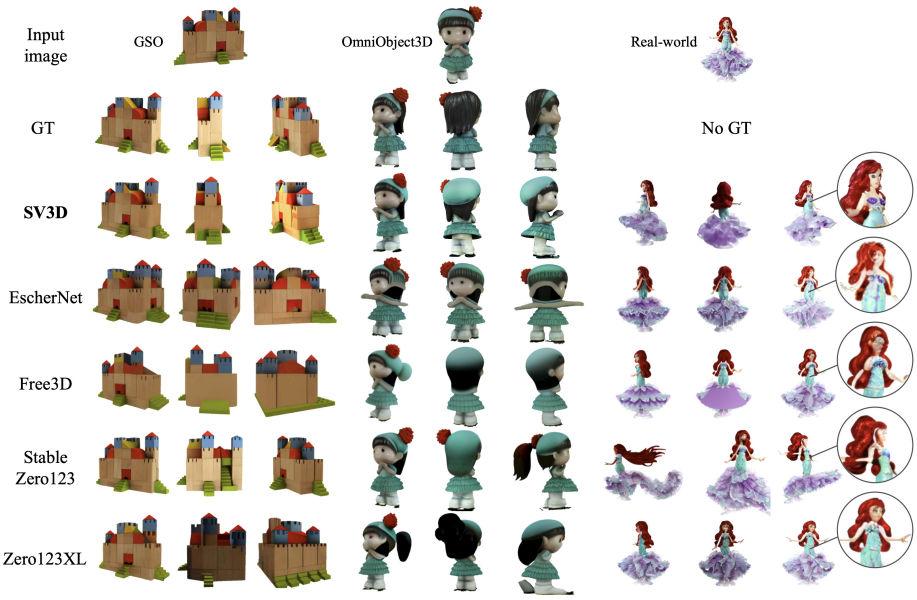

Fig. 6: Visual Comparison of Novel Multi-view Synthesis. SV3D is able to generate novel multi-views that are more detailed, faithful to the conditioning image, and multi-view consistent compared to the prior works.

Ablative Analyses. Interestingly, from Tabs. 1 and 3, we find that both SV3D^c and SV3D^p outperform SV3D^u on generations of static orbits, even though SV3D^u is trained specifically on static orbits. We also observe that SV3D^p achieves better metrics than SV3D^c on both static and dynamic orbits, making it the best performing SV3D model overall. This shows that progressive finetuning from easier (static) to harder (dynamic) tasks is indeed a favorable way to finetune a video diffusion model.

Visual Comparisons in Fig. 6 further demonstrate that SV3D-generated images are more detailed, faithful to the conditioning image, and multi-view consistent compared to the prior works.

Quality Per Frame. We also observe from Fig. 5 that SV3D produces better quality at every frame. We plot the average LPIPS value for each generated frame, across generated GSO static orbit videos. The quality is generally worse around the back view, and better at the beginning and the end (*i.e.* near the conditioning image), as expected.

User Study on Real-World Images. We conducted a user survey to study human preference between static orbital videos generated by SV3D and by other methods. We asked 30 users to pick one between our SV3D-generated static video and other method-generated video as the best orbital video for the corresponding image, using 22 real-world images. We noted that users preferred SV3D-generated videos over Zero123XL, Stable Zero123, EscherNet, and Free3D, 96%, 99%, 96%, and 98% of the time, respectively.

Table 1: Evaluation of novel multi-view synthesis on GSO static orbits

Model	LPIPS↓	PSNR↑	SSIM↑	CLIP-S↑	MSE↓
SyncDreamer [27]	0.17	15.78	0.76	0.87	0.03
Zero123 [26]	0.13	17.29	0.79	0.85	0.04
Zero123XL [8]	0.14	17.11	0.78	0.85	0.04
Stable Zero123 [49]	0.13	18.34	0.78	0.85	0.05
Free3D [64]	0.15	16.18	0.79	0.84	0.04
EscherNet [21]	0.13	16.73	0.79	0.85	0.03
SV3D^u	0.09	21.14	0.87	**0.89**	**0.02**
SV3D^c	0.09	20.56	0.87	0.88	**0.02**
SV3D^p	**0.08**	**21.26**	**0.88**	**0.89**	**0.02**

Table 2: Evaluation of novel multi-view synthesis on GSO dynamic orbits

Model	LPIPS↓	PSNR↑	SSIM↑	CLIP-S↑	MSE↓
Zero123 [26]	0.14	16.99	0.79	0.84	0.04
Zero123XL [8]	0.14	16.73	0.78	0.84	0.04
Stable Zero123 [49]	0.13	18.04	0.78	0.85	0.05
Free3D [64]	0.18	14.93	0.77	0.83	0.05
EscherNet [21]	0.13	16.47	0.79	0.84	0.03
SV3D^c	0.10	19.99	0.86	**0.87**	**0.02**
SV3D^p	**0.09**	**20.38**	**0.87**	**0.87**	**0.02**

Table 3: Evaluation of novel multi-view synthesis on OmniObject3D static orbits

Model	LPIPS↓	PSNR↑	SSIM↑	CLIP-S↑	MSE↓
Zero123 [26]	0.17	15.50	0.76	0.83	0.05
Zero123XL [8]	0.18	15.36	0.75	0.83	0.06
Stable Zero123 [49]	0.15	16.86	0.77	0.84	0.06
Free3D [64]	0.16	15.29	0.78	0.83	0.05
EscherNet [21]	0.17	14.63	0.74	0.83	0.05
SV3D^u	**0.10**	19.68	**0.86**	**0.86**	**0.02**
SV3D^c	**0.10**	19.50	0.85	0.85	**0.02**
SV3D^p	**0.10**	**19.91**	**0.86**	**0.86**	**0.02**

Table 4: Evaluation of novel multi-view synthesis on OmniObject3D dynamic orbits

Model	LPIPS↓	PSNR↑	SSIM↑	CLIP-S↑	MSE↓
Zero123 [26]	0.16	15.78	0.77	0.82	0.05
Zero123XL [8]	0.17	15.49	0.76	0.83	0.05
Stable Zero123 [49]	0.14	16.74	0.77	0.83	0.05
Free3D [64]	0.19	14.28	0.76	0.82	0.06
EscherNet [21]	0.16	15.05	0.76	0.83	0.05
SV3D^c	**0.10**	19.21	**0.85**	0.84	**0.02**
SV3D^p	**0.10**	**19.28**	**0.85**	**0.85**	**0.02**

4 3D Generation from a Single Image Using SV3D

We then generate 3D meshes of objects from a single image by leveraging SV3D. One way is to use the generated static/dynamic orbital samples from SV3D as direct reconstruction targets. Another way is to use SV3D as diffusion guidance with Score Distillation Sampling (SDS) loss [37].

Since SV3D produces more consistent multi-views compared to existing works, we already observe higher-quality 3D reconstructions by only using SV3D outputs for reconstruction when compared to existing works. However, we observe that this naive approach often leads to artifacts like baked-in illumination, rough surfaces, and noisy texture, especially for the unseen regions in the reference orbit. Thus, we further propose several techniques to address these issues.

Coarse-to-Fine Training. We adopt a two-stage, coarse-to-fine training scheme to generate a 3D mesh from input images, similar to [23, 38]. Fig. 7 illustrate an overview of our 3D optimization pipeline. In the coarse stage, we train an Instant-NGP [33] NeRF representation to reconstruct the SV3D-generated images (*i.e.* without SDS loss) at a lower resolution. In the fine stage, we extract a mesh from the trained NeRF using marching cubes [7], and adopt a DMTet [44] representation to finetune the 3D mesh using SDS-based diffusion guidance from SV3D at full-resolution. Finally, we use xatlas [60] to perform the UV unwrapping and export the output object mesh.

Disentangled Illumination Model. Similar to other recent 3D object generation methods [23, 31, 37], our output target is a mesh with a diffuse texture. Typically, such SDS-based optimization techniques use random illuminations at every iteration. However, our SV3D-generated videos are under consistent illumination, *i.e.*, the lighting remains static while the camera circles around an object.

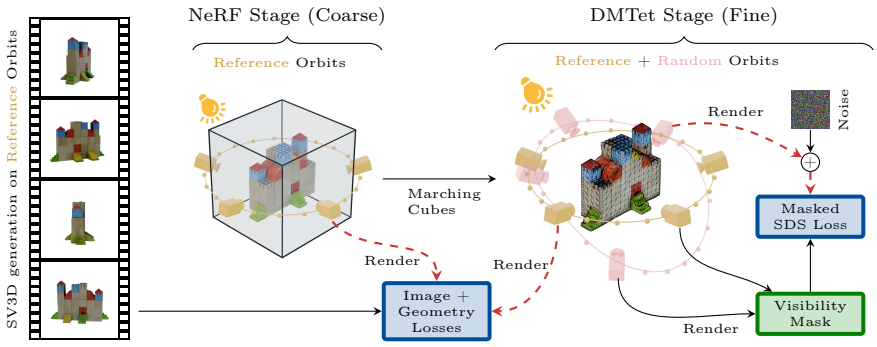

Fig. 7: 3D Optimization Pipeline. We perform a two-stage optimization. In the short first stage, we extract the general shape, texture and illumination from the SV3D generated multi-view images. In the second stage, we extract a mesh with marching cubes and use DMTet to further optimize the shape, texture and illumination. We not only use the SV3D-generated images but a soft-masked SDS loss for unseen areas. Dashed red lines represent backpropagation into the differentiable rendering pipeline.

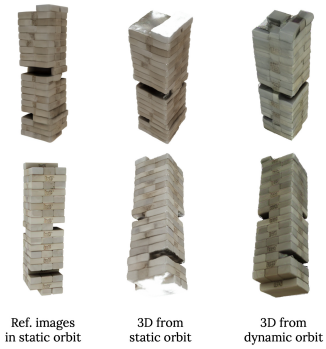

Fig. 8: Influence of Training Orbits. We show that using a dynamic orbit is crucial to 3D generations that are complete from diverse views.

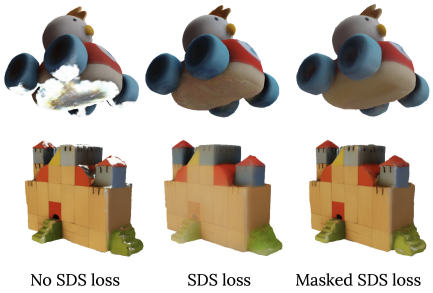

Fig. 9: Influence of SDS. Using our masked SDS loss, we are able to fill in unseen surfaces in the training orbit, producing a cleaner result without oversaturation or blurry artifacts caused by naive SDS.

Hence, to disentangle lighting effects and obtain a cleaner texture, we propose to fit a simple illumination model of 24 Spherical Gaussians (SG) inspired by prior decomposition methods [4, 62].

We model white light and hence only use a scalar amplitude for the SGs. We only consider Lambertian shading, where the cosine shading term is approximated with another SG. We learn the parameters of the illumination SGs using a reconstruction loss between the rendered images and SV3D-generated images.

Inspired by [13, 37] we reduce baked-in illumination with a loss term that replicates the HSV-value component of the input image $\mathbf{I}$ with the rendered illumination L: $\mathcal{L}_{\text{illum}} = |V(\mathbf{I}) - L|^2$, with $V(\boldsymbol{c}) = \max(c_r, c_g, c_b)$. Given these changes, our disentangled illumination model is able to express lighting variation properly and can severely reduce baked-in illumination. Fig. 10 shows sample

reconstructions with and without our illumination modelling. From the results, it is clear that we are able to disentangle the illumination effects from the base color (e.g., dark side of the school bus).

4.1 3D Optimization Strategies and Losses

Reconstruction via Photometric Losses. Intuitively, we can treat the SV3D-generated images as multi-view pseudo-ground truth, and apply 2D reconstruction losses to train the 3D models. In this case, we apply photometric losses on the rendered images from NeRF or DMTet, including the pixel-level MSE loss, mask loss, and a perceptual LPIPS [63] loss. These photometric losses also optimize our illumination model through the differential rendering pipeline.

Training Orbits. For 3D generation, we use SV3D to generate multi-view images following a camera orbit $\boldsymbol{\pi}_{\text{ref}}$, referred to as the *reference orbit* (also see Fig. 7 for the overview). Fig. 8 shows sample reconstruction with using static and dynamic orbital outputs from SV3D. Using a dynamic orbit for training is crucial to high-quality 3D outputs when viewed from various angles, since some top/bottom views are missing in the static orbit (fixed elevation). Hence, for SV3D^c and SV3D^p, we render images on a dynamic orbit whose elevation follows a sine function to ensure that top and bottom views are covered.

Fig. 10: Constant vs. SGs Illumination. Our SGs-based reconstructions do not exhibit darkening on the side of the bus, which enables easier and more convincing relighting for downstream applications.

SV3D-Based SDS Loss. In addition to the reconstruction losses, we can also use SV3D via score-distillation sampling (SDS) [37, 58]. Fig. 9 shows sample reconstructions with and without using SDS losses. As shown in Fig. 9 (left), although training with a dynamic orbit improves overall visibility, we observe that sometimes the output texture is still noisy, perhaps due to partial visibility, self-occlusions, or inconsistent texture/shape between images. Hence, we handle those unseen areas using SDS loss [37] with SV3D as a diffusion guidance.

We sample a random camera orbit $\boldsymbol{\pi}_{\text{rand}}$, and use our 3D NeRF/DMTet parameterized by θ to render the views $\hat{\mathbf{J}}$ of the object along $\boldsymbol{\pi}_{\text{rand}}$. Then, noise ϵ at level t is added to the latent embedding $\mathbf{z}_t$ of $\hat{\mathbf{J}}$, and the following SDS loss (taken expectation over t and ϵ) is backpropagated through the differentiable rendering pipeline: $\mathcal{L}_{\text{sds}} = w(t)\big(\epsilon_\phi(\mathbf{z}_t; \mathbf{I}, \boldsymbol{\pi}_{\text{rand}}, t) - \epsilon\big)\frac{\partial \hat{\mathbf{J}}}{\partial \theta}$, where w is t-dependent weight, ϵ and ϵ_ϕ are the added and predicted noise, and ϕ and θ are the parameters of SV3D and NeRF/DMTet, respectively. See Fig. 7 for an illustration of these loss functions in the overall pipeline.

Masked SDS Loss. As shown in Fig. 9 (middle), in our experiments we found that adding the SDS loss naively can cause unstable training and unfaithful texture to the input images such as oversaturation or blurry artifacts. Therefore, we design a soft masking mechanism to only apply SDS loss on the unseen/occluded areas, allowing it to inpaint the missing details while preserving the texture of

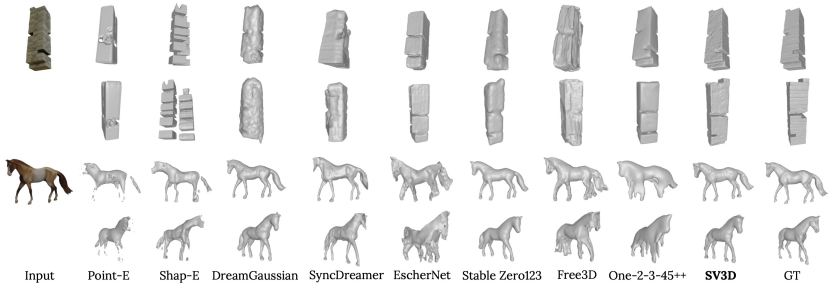

Fig. 11: Visual Comparison of 3D Meshes. For each object, we show the conditioning image (left), and the output meshes rendered in two different views. Our generated meshes are more detailed, faithful to input images, and consistent in 3D. This demonstrates the quality of novel multi-view synthesis by our SV3D model.

clearly-visible surfaces in the training orbit (as seen in Fig. 9 (right)). Also, we only apply the masked SDS loss in the final stage of DMTet optimization, which greatly increased the convergence speed.

We apply SDS loss on only those pixels in the random orbit views that are not likely visible in the reference orbit views. For this, we first render the object from the random camera orbit $\boldsymbol{\pi}_{\text{rand}}$. For each random camera view, we obtain the visible surface points $\boldsymbol{p} \in \mathbb{R}^3$ and their corresponding surface normals $\boldsymbol{n}$. Then, for each reference camera i, we calculate the view directions $\boldsymbol{v}_i$ of the surface $\boldsymbol{p}$ towards its position $\bar{\boldsymbol{\pi}}^i_{\text{ref}} \in \mathbb{R}^3$ (calculated from $\boldsymbol{\pi}^i_{\text{ref}} \in \mathbb{R}^2$) as $\boldsymbol{v}_i = \frac{\bar{\boldsymbol{\pi}}^i_{\text{ref}} - \boldsymbol{p}}{||\bar{\boldsymbol{\pi}}^i_{\text{ref}} - \boldsymbol{p}||}$. We infer the visibility of this surface from the reference camera based on the dot product between $\boldsymbol{v}_i$ and $\boldsymbol{n}$ *i.e.* $\boldsymbol{v}_i \cdot \boldsymbol{n}$. Since higher values roughly indicate more visibility of the surface from the reference camera, we chose that reference camera c that has maximum likelihood of visibility: $c = \max_i (\boldsymbol{v}_i \cdot \boldsymbol{n})$. As we only want to apply SDS loss to unseen or grazing angle areas from c, we use the smoothstep function f_s to smoothly clip to c's visibility range $\boldsymbol{v}_c \cdot \boldsymbol{n}$. In this way, we create a pseudo visibility mask $M = 1 - f_{\text{s}}(\boldsymbol{v}_c \cdot \boldsymbol{n}, 0, 0.5)$, where $f_{\text{s}}(x; f_0, f_1) = \hat{x}^2(3 - 2x)$, with $\hat{x} = \frac{x - f_0}{f_1 - f_0}$. Thus, M is calculated for each random camera render, and the combined visibility mask $\mathbf{M}$ is applied to SDS loss: $\mathcal{L}_{\text{mask-sds}} = \mathbf{M}\mathcal{L}_{\text{sds}}$.

Geometric Priors. Since our rendering-based optimization operates at the image level, we adopt several geometric priors to regularize the output shapes. We add a smooth depth loss from RegNeRF [35] and a bilateral normal smoothness loss [5] to encourage smooth 3D surfaces where the projected image gradients are low. Moreover, we obtain normal estimates from Omnidata [11] and calculate a mono normal loss similar to MonoSDF [61], which can effectively reduce noisy surfaces in the output mesh. Further details about the training losses and optimization process are available in the appendix.

4.2 Experiments and Results

Due to the strong regularization, we only require 600 steps in the coarse stage and 1000 in the fine stage. Overall, the entire mesh extraction process takes ≈8

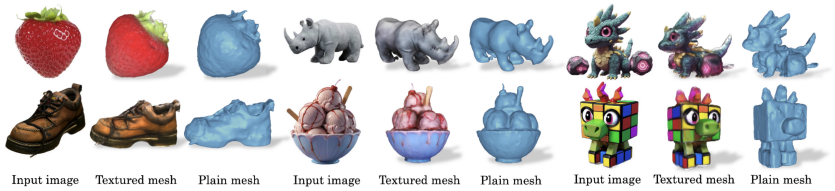

Fig. 12: Real-World 3D Results. Notice the accurate shape and details in our reconstructions even from the diverse images in-the-wild.

minutes without SDS loss, and ≈20 minutes with SDS loss. The coarse stage only takes ≈2 minutes and provides a full 3D representation of the object.

Evaluation. We evaluate our 3D generation framework on 50 randomly sampled objects from the GSO dataset as described in Sec. 3.1. We compute image-based reconstruction metrics (LPIPS, PSNR, SSIM, MSE, and CLIP-S) between the ground truth (GT) GSO images, and rendered images from the trained 3D meshes on the same static/dynamic orbits. In addition, we compute 3D reconstruction metrics of Chamfer distance (CD) and 3D IoU between the GT and predicted meshes. We compare our SV3D-guided 3D generations with several prior methods including Point-E [34], Shap-E [16], One-2-3-45++ [24], DreamGaussian [50], SyncDreamer [27], EscherNet [21], Free3D [64], and Stable Zero123 [49].

Visual Comparison. In Fig. 11, we show visual comparison of our results with those from prior methods. Qualitatively, Point-E [34] and Shap-E [16] often produce incomplete 3D shapes. DreamGaussian [50], SyncDreamer [27], EscherNet [21], and Free3D [64] outputs tend to contain rough surfaces. One-2-3-45++ [24] and Stable Zero123 [49] are able to reconstruct meshes with smooth surface, but lack geometric details. Our mesh outputs are detailed, faithful to input image, and consistent in 3D (see appendix for more examples). We also show 3D mesh renders from real-world images in Fig. 12.

Quantitative Comparison. Tabs. 5 and 6 show the 2D and 3D metric comparisons respectively. All our 3D models achieve better 2D and 3D metrics compared to the prior and concurrent methods, showing the high-fidelity texture and geometry of our output meshes. We render all 3D meshes on the same dynamic orbits and compare them against the GT renders. Our best model, SV3D^p, performs comparably to using GT renders for reconstruction in terms of the 3D metrics, which further demonstrates the 3D consistency of our generated images.

Effects of Photometric and (Masked) SDS Loss. As shown in Tabs. 5 and 6, the 3D outputs using both photometric and Masked SDS losses ('**SV3D^p**') achieves the best results, while training without SDS ('SV3D^p no SDS') leads to marginally lower performance. This demonstrates that the images generated by SV3D are high-quality reconstruction targets, and are often sufficient for 3D generation without the cumbersome SDS-based optimization. Nevertheless, adding SDS generally improves quality, as also shown in Fig. 9.

Effects of SV3D Model and Training Orbit. As shown in Tabs. 5 and 6, SV3D^p achieves the best performance among the three SV3D models, indicating that its synthesized novel views are most faithful to the input image and consis-

Table 5: 2D comparison of our 3D outputs against prior methods on the GSO dataset. Our best performing method uses **SV3D^p** with dynamic (sine elevation) orbit and SDS guidance. Note that all our models achieve better 2D metrics than prior works.

Model	LPIPS↓	PSNR↑	SSIM↑	MSE↓	CLIP-S↑
GT renders	0.078	19.508	0.879	0.014	0.897
EscherNet [21]	0.178	14.438	0.804	0.041	0.835
Free3D [64]	0.197	14.202	0.799	0.043	0.809
Stable Zero123 [49]	0.166	14.635	0.813	0.040	0.805
SV3D^u	0.133	15.957	0.834	0.031	0.871
SV3D^c	0.132	16.373	0.834	0.027	0.870
SV3D^p static orbit	0.125	16.821	0.848	0.025	0.864
SV3D^p no SDS	0.124	16.864	0.841	0.024	0.875
SV3D^p	**0.119**	**17.405**	**0.849**	**0.021**	**0.877**

Table 6: Comparison of 3D metrics on the GSO dataset. Our models perform favorably against prior works.

Model	CD↓	3D IoU↑
GT renders	0.021	0.689
Point-E [34]	0.074	0.162
Shap-E [16]	0.071	0.267
DreamGaussian [50]	0.055	0.411
One-2-3-45++ [24]	0.054	0.406
SyncDreamer [27]	0.053	0.451
EscherNet [21]	0.042	0.466
Free3D [64]	0.047	0.426
Stable Zero123 [49]	0.039	0.550
SV3D^u	0.027	0.589
SV3D^c	0.027	0.584
SV3D^p static orbit	0.028	0.610
SV3D^p no SDS	**0.024**	0.611
SV3D^p	**0.024**	**0.614**

tent in 3D. On the other hand, SV3D^u shares the same disadvantage as several prior works in that it can only generate views at the same elevation, which is insufficient to build a legible 3D object, as shown in Fig. 8. This also leads to the worse performance of 'SV3D^p with static orbit' in Tabs. 5 and 6. Overall, SV3D^p with dynamic orbit and masked SDS loss performs favorably against all other configurations since it can leverage more diverse views of the object.

Limitations. Our SV3D model is by design only capable of handling 2 degrees of freedom: elevation and azimuth; which is usually sufficient for 3D generation from a single image. One may want to tackle more degrees of freedom in cameras for a generalized NVS system, which forms an interesting future work. We also notice that SV3D exhibits some view inconsistency for mirror-like reflective surfaces, which provide a challenge to 3D reconstruction. Lastly, such reflective surfaces are not representable by our Lambertian reflection-based shading model. Conditioning SV3D on the full camera matrix, and extending the shading model are interesting directions for future research.

5 Conclusion

We present SV3D, a latent video diffusion model for novel multi-view synthesis and 3D generation. In addition to leveraging the generalizability and view-consistent prior in SVD, SV3D enables controllability via camera pose conditioning, and generates orbital videos of an object at high resolution on arbitrary camera orbits. We further propose several techniques for improved 3D generation from SV3D, including triangle CFG scaling, disentangled illumination, and masked SDS loss. We conduct extensive experiments to show that SV3D is controllable, multi-view consistent, and generalizable to the real-world, achieving state-of-the-art performance on novel multi-view synthesis and 3D generation. We believe SV3D provides a solid foundation model for further research on 3D object generation. We plan to publicly release SV3D models.

Acknowledgements

We thank Emad Mostaque, Bryce Wilson, Anel Islamovic, Savannah Martin, Ana Guillen, Josephine Parquet, Adam Chen and Ella Irwin for their help in various aspects of the model development and release. We thank Kyle Lacey and Yan Marek for their help with demos. We also thank Eric Courtemanche for his help with visual results.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021)
2. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable Video Diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
3. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. arXiv:2304.08818 (2023)
4. Boss, M., Braun, R., Jampani, V., Barron, J.T., Liu, C., Lensch, H.P.: NeRD: Neural reflectance decomposition from image collections. In: IEEE International Conference on Computer Vision (ICCV) (2021)
5. Boss, M., Engelhardt, A., Kar, A., Li, Y., Sun, D., Barron, J.T., Lensch, H.P., Jampani, V.: SAMURAI: Shape And Material from Unconstrained Real-world Arbitrary Image collections. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
6. Chan, E.R., Nagano, K., Chan, M.A., Bergman, A.W., Park, J.J., Levy, A., Aittala, M., Mello, S.D., Karras, T., Wetzstein, G.: GeNVS: Generative novel view synthesis with 3D-aware diffusion models. International Conference on Computer Vision (2023)
7. Cline, D.B.: Admissibile kernel estimators of a multivariate density. The Annals of Statistics **16**(4), 1421–1427 (1988)
8. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-XL: A universe of 10m+ 3D objects. arXiv preprint arXiv:2307.05663 (2023)
9. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3D objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13142–13153 (2023)
10. Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T.B., Vanhoucke, V.: Google Scanned Objects: A high-quality dataset of 3D scanned household items. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 2553–2560. IEEE (2022)
11. Eftekhar, A., Sax, A., Bachmann, R., Malik, J., Zamir, A.: Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3D scans. In: IEEE International Conference on Computer Vision (ICCV) (2021)

12. Girdhar, R., Singh, M., Brown, A., Duval, Q., Azadi, S., Rambhatla, S.S., Shah, A., Yin, X., Parikh, D., Misra, I.: EMU VIDEO: Factorizing Text-to-Video Generation by Explicit Image Conditioning (2023), https://emu-video.metademolab.com/assets/emu_video.pdf
13. Hasselgren, J., Hofmann, N., Munkberg, J.: Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (2020)
15. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: LRM: Large reconstruction model for single image to 3D. International Conference on Learning Representations (2024)
16. Jun, H., Nichol, A.: Shap-E: Generating conditional 3D implicit functions (2023)
17. Kanopoulos, N., Vasanthavada, N., Baker, R.L.: Design of an image edge detection filter using the sobel operator. IEEE Journal of solid-state circuits **23**(2), 358–367 (1988)
18. Kant, Y., Wu, Z., Vasilkovsky, M., Qian, G., Ren, J., Guler, R.A., Ghanem, B., Tulyakov, S., Gilitschenski, I., Siarohin, A.: Spad : Spatially aware multiview diffusers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2024)
19. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the Design Space of Diffusion-Based Generative Models. arXiv:2206.00364 (2022)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Kong, X., Liu, S., Lyu, X., Taher, M., Qi, X., Davison, A.J.: Eschernet: A generative model for scalable view synthesis. arXiv preprint arXiv:2402.03908 (2024)
22. Kwak, J.g., Dong, E., Jin, Y., Ko, H., Mahajan, S., Yi, K.M.: Vivid-1-to-3: Novel view synthesis with video diffusion models. IEEE conference on computer vision and pattern recognition (2024)
23. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3D: High-resolution text-to-3D content creation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 300–309 (2023)
24. Liu, M., Shi, R., Chen, L., Zhang, Z., Xu, C., Wei, X., Chen, H., Zeng, C., Gu, J., Su, H.: One-2-3-45++: Fast single image to 3D objects with consistent multi-view generation and 3D diffusion. arXiv preprint arXiv:2311.07885 (2023)
25. Liu, M., Xu, C., Jin, H., Chen, L., Varma T, M., Xu, Z., Su, H.: One-2-3-45: Any single image to 3D mesh in 45 seconds without per-shape optimization. Advances in Neural Information Processing Systems **36** (2023)
26. Liu, R., Wu, R., Hoorick, B.V., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3D object. International Conference on Computer Vision (2023)
27. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: SyncDreamer: Generating multiview-consistent images from a single-view image. arXiv preprint arXiv:2309.03453 (2023)
28. Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., et al.: Wonder3D: Single image to 3D using cross-domain diffusion. arXiv preprint arXiv:2310.15008 (2023)
29. Melas-Kyriazi, L., Laina, I., Rupprecht, C., Neverova, N., Vedaldi, A., Gafni, O., Kokkinos, F.: IM-3D: Iterative multiview diffusion and reconstruction for high-quality 3D generation. arXiv preprint arXiv:2402.08682 (2024)

30. Mercier, A., Nakhli, R., Reddy, M., Yasarla, R., Cai, H., Porikli, F., Berger, G.: HexaGen3D: Stablediffusion is just one step away from fast and diverse Text-to-3D generation. arXiv preprint arXiv:2401.07727 (2024)
31. Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-NeRF for shape-guided generation of 3D shapes and textures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12663–12673 (2023)
32. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision. pp. 405–421 (2020)
33. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. (2022)
34. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-E: A System for Generating 3D Point Clouds from Complex Prompts (2022)
35. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S.M., Geiger, A., Radwan, N.: RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2022)
36. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
37. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3D using 2D diffusion. arXiv (2022)
38. Qian, G., Mai, J., Hamdi, A., Ren, J., Siarohin, A., Li, B., Lee, H.Y., Skorokhodov, I., Wonka, P., Tulyakov, S., et al.: Magic123: One image to high-quality 3D object generation using both 2D and 3D diffusion priors. arXiv preprint arXiv:2306.17843 (2023)
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
40. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
41. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: DreamBooth: Fine tuning text-to-image dissusion models for subject-driven generation. arXiv preprint arXiv:2208.12242 (2022)
42. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Kamyar Seyed Ghasemipour, S., Karagol Ayan, B., Mahdavi, S.S., Gontijo Lopes, R., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. arXiv:2205.11487 (2022)
43. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: LAION-5B: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems **35**, 25278–25294 (2022)
44. Shen, T., Gao, J., Yin, K., Liu, M.Y., Fidler, S.: Deep Marching Tetrahedra: a hybrid representation for high-resolution 3D shape synthesis. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
45. Shi, R., Chen, H., Zhang, Z., Liu, M., Xu, C., Wei, X., Chen, L., Zeng, C., Su, H.: Zero123++: a single image to consistent multi-view diffusion base model. arXiv preprint arXiv:2310.15110 (2023)

46. Shi, Y., Wang, P., Ye, J., Long, M., Li, K., Yang, X.: MVDream: Multi-view diffusion for 3d generation. arXiv preprint arXiv:2308.16512 (2023)
47. Song, Y., Ermon, S.: Improved Techniques for Training Score-Based Generative Models. arXiv:2006.09011 (2020)
48. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
49. StabilityAI: Stable Zero123 (2023), `https://stability.ai/news/stable-zero123-3d-generation`
50. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: DreamGaussian: Generative Gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023)
51. Tang, S., Chen, J., Wang, D., Tang, C., Zhang, F., Fan, Y., Chandra, V., Furukawa, Y., Ranjan, R.: Mvdiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3d object reconstruction. arXiv preprint arXiv:2402.12712 (2024)
52. Tsai, Y.T., Shih, Z.C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. ACM Transactions on Graphics (ToG) (2006)
53. Voleti, V., Jolicoeur-Martineau, A., Pal, C.: MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In: (NeurIPS) Advances in Neural Information Processing Systems (2022)
54. Watson, D., Chan, W., Martin-Brualla, R., Ho, J., Tagliasacchi, A., Norouzi, M.: Novel view synthesis with diffusion models (2022)
55. Weng, H., Yang, T., Wang, J., Li, Y., Zhang, T., Chen, C.L.P., Zhang, L.: Consistent123: Improve consistency for one image to 3D object synthesis (2023)
56. Wu, T., Zhang, J., Fu, X., Wang, Y., Ren, J., Pan, L., Wu, W., Yang, L., Wang, J., Qian, C., et al.: Omniobject3D: Large-vocabulary 3D object dataset for realistic perception, reconstruction and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 803–814 (2023)
57. Yang, J., Cheng, Z., Duan, Y., Ji, P., Li, H.: ConsistNet: Enforcing 3D consistency for multi-view images diffusion. arXiv preprint arXiv:2310.10343 (2023)
58. Yao, C.H., Raj, A., Hung, W.C., Rubinstein, M., Li, Y., Yang, M.H., Jampani, V.: ARTIC3D: Learning robust articulated 3D shapes from noisy web image collections. Advances in Neural Information Processing Systems **36** (2024)
59. Ye, J., Wang, P., Li, K., Shi, Y., Wang, H.: Consistent-1-to-3: Consistent image to 3D view synthesis via geometry-aware diffusion models. In: 3DV (2024)
60. Young, J.: xatlas. `https://github.com/jpcy/xatlas` (2024)
61. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction. Advances in Neural Information Processing Systems (NeurIPS) (2022)
62. Zhang, K., Luan, F., Wang, Q., Bala, K., Snavely, N.: PhySG: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
63. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
64. Zheng, C., Vedaldi, A.: Free3D: Consistent novel view synthesis without 3D representation. arXiv preprint arXiv:2312.04551 (2023)

In this supplemental document, we include the broader impact in Appendix A, implementation details in Appendices B to F, ablative analyses in Appendix G, and additional results in Appendix H. We also attach a teaser video summarizing the SV3D framework (`SV3D_video.mp4`), as well as an HTML page (`SV3D.html`) for more visual comparisons.

A Broader Impact

The advancement of generative models in different media forms is changing how we make and use content. These AI-powered models can create images, videos, 3D objects, and more, in ways we've never seen before. They offer huge potential for innovation in media production. But along with this potential, there are also risks. Before we start using these models widely, it is crucial to make sure we understand the possible downsides and have plans in place to deal with them effectively.

In the case of 3D object generation, the input provided by the user plays a crucial role in constraining the model's output. By supplying a full front view of an object, users limit the model's creative freedom to the visible or unoccluded regions, thus minimizing the potential for generating problematic imagery. Additionally, factors such as predicted depth values and lighting further influence the fidelity and realism of generated content.

Moreover, ensuring the integrity and appropriateness of training data is critical in mitigating risks associated with generative models. Platforms like Sketchfab, which serve as repositories for 3D models used in training, enforce strict content policies to prevent the dissemination of "Unacceptable Content" and disallows it on their platform: https://help.sketchfab.com/hc/en-us/articles/214867883-What-is-Restricted-Content. By adhering to these guidelines and actively monitoring dataset quality, developers can reduce the likelihood of biased or inappropriate outputs.

Sketchfab also has a tag for "Restricted Content" which is deemed to be similar to the PG-13 content rating used in the US (i.e. inappropriate for children under 13). We have confirmed that none of the objects that we use in training have this flag set to true. Thus we go the extra step of excluding even tagged PG-13 content from the training set.

There is a chance that certain content may not have been correctly labeled on Sketchfab. In cases where the uploader fails to tag an object appropriately, Sketchfab provides publicly accessible listings of objects and a mechanism for the community to report any content that may be deemed offensive.

In our regular utilization of Objaverse content, we haven't observed any significant amount of questionable material. Nevertheless, there are occasional instances of doll-like nudity stemming from basic 3D models, which could be crucial for accurately depicting humanoid anatomy. Additionally, the training dataset contains some presence of drugs, drug paraphernalia, as well as a certain level of blood content and weaponry, resembling what might be encountered in a video game context. Should the model be provided with imagery featuring these cate-

gories of content, it possesses the capability to generate corresponding 3D models to some extent.

It is to be noted that SV3D mainly focuses on generating hidden details in the user's input image. If the image is unclear or some parts are hidden, the model guesses what those parts might look like based on its training data. This means it might create details similar to what it has seen before. The training data generally follows the standards of 3D modeling and gaming. However, this could lead to criticisms about the models being too similar to existing trends. But the user's input image limits how creative the model can be and reduces the chance of biases showing up in its creations, especially if the image is clear and straightforward.

B Data Details

Similar to SVD-MV [2], we render views of a subset of the Objaverse dataset [9] consisting of 150K curated CC-licensed 3D objects from the original dataset. Each loaded object is scaled such that the largest world-space XYZ extent of its bounding box is 1. The object is then repositioned such that this bounding box is centered around the origin.

For both the static and dynamic orbits, we use Blender's EEVEE renderer to render an 84-frame RGBA orbit at 576×576 resolution. During training, any 21-frame orbit can be subsampled from this by picking any frame as the first frame, and then choosing every 4th frame after that.

We apply two background colors to each of these images: random RGB color, and white. This results in a doubling of the number of orbit samples for training. We then encode all of these images into latent space using SVD's VAE, and using CLIP. We then store the latent and CLIP embeddings for all of these images along with the corresponding elevation and azimuth values.

For lighting, we randomly select from a set of 20 curated HDRI envmaps. Each orbit begins with the camera positioned at azimuth 0. Our camera uses a field-of-view of 33.8 degrees. For each object, we adaptively position the camera to a distance sufficient to ensure that the rendered object content makes good and consistent use of the image extents without being clipped in any view.

For static orbits, the camera is positioned at a randomly sampled elevation between [-5, 30] degrees. The azimuth steps by a constant $\frac{360}{84}$ degree delta between each frame. For dynamic orbits, the sequence of camera elevations for each orbit are obtained from a random weighted combination of sinusoids with different frequencies. For each sinusoid, the whole number period is sampled from [1, 5], the amplitude is sampled from [0.5, 10], and a random phase shift is applied. The azimuth angles are sampled regularly, and then a small amount of noise is added to make them irregular. The elevation values are smoothed using a simple convolution kernel and then clamped to a maximum elevation of 89 degrees.

C Training Details

Our approach involves utilizing the widely used EDM [19] framework, incorporating a simplified diffusion loss for fine-tuning, as followed in SVD [2]. We eliminate the conditions of 'fps_id', 'motion_bucket_id', etc. since they are irrelevant for SV3D. Furthermore, we adjust the loss function to assign lower weights to frames closer to the front-view conditioning image, ensuring that challenging back views receive equal training focus as the easier front views. To optimize training efficiency and conserve GPU VRAM, we preprocess and store the precomputed latent and CLIP embeddings for all video frames in advance. During training, these tensors are directly loaded rather than being computed in real-time. We choose to finetune the SVD-xt model to output 21 frames instead of 25 frames. We found that with 21 frames we were able to fit a batch size of 2 on each GPU, instead of 1 with 25 frames at 576×576 resolution. All SV3D models are trained for 105k iterations with an effective batch size of 64 on 4 nodes of 8 80GB A100 GPUs for around 6 days.

D Inference Details

To generate an orbital video during inference, we use 50 steps of the deterministic DDIM sampler [47] with the triangular classifier-free guidance (CFG) scale described in the main paper. This takes ≈40 seconds for the SV3D model.

E Additional Details on Illumination Model

We base our rendering model on Spherical Gaussians (SG) [4,62]. A SG at query location $\boldsymbol{x} \in \mathbb{R}^3$ is defined as $G(\boldsymbol{x}; \boldsymbol{\mu}, c, a) = ae^{s(\boldsymbol{\mu}\cdot\boldsymbol{x}-1)}$, where $\boldsymbol{\mu} \in \mathbb{R}^3$ is the axis, $s \in \mathbb{R}$ the sharpness of the lobe, and $a \in \mathbb{R}$ the amplitude. Here, we point out that we only model white light and hence only use a scalar amplitude. One particularly interesting property of SGs is that the inner product of two SGs is the integral of the product of two SGs. The operation is defined as [52]:

$$\begin{aligned} G_1(\boldsymbol{x}) \cdot G_2(\boldsymbol{x}) = \int_\Omega G_1(\boldsymbol{x}) G_2(\boldsymbol{x}) d\boldsymbol{x} &= \frac{1}{d_m}\left(2\pi a_1 a_2 e^{d_m - \lambda_m}(1.0 - e^{-2d_m})\right) \\ \lambda_m &= \lambda_1 - \lambda_2 \\ d_m &= ||\lambda_1\boldsymbol{\mu}_1 + \lambda_2\boldsymbol{\mu}_2||. \end{aligned} \tag{1}$$

In our illumination model we only consider Lambertian shading. Here, the cosine shading term influences the output the most. This term can be approximated with another SG $G_{\text{cosine}} = (\boldsymbol{x}; \boldsymbol{n}, 2.133, 1.17)$, where $\boldsymbol{n}$ defines the surface normal at $\boldsymbol{x}$. The lighting evaluation using SGs G_i then becomes: $L = \sum_{i=1}^{24} \frac{1}{\pi} \max(G_i \cdot G_{\text{cosine}}, 0)$. As defined previously this results in the full integration of incoming light for each SG and as light is additive evaluating and summing all SGs results in the complete environment illumination. This L is also used in the $\mathcal{L}_{\text{illum}}$ loss described in the main paper. The rendered textured

image is then defined as $\hat{\boldsymbol{I}} = \boldsymbol{c_d}\boldsymbol{L}$, where $\boldsymbol{c_d}$ is the diffuse albedo. We learn $\boldsymbol{\mu}, c, a$ for each SG G_i using reconstruction loss between these rendered images and SV3D-generated images.

F Losses and Optimization for 3D Generation

In addition to the masked SDS loss $\mathcal{L}_{\text{mask-sds}}$ and illumination loss $\mathcal{L}_{\text{illum}}$ detailed in the manuscript, we use several other losses for 3D reconstruction. Our main reconstruction losses are the pixel-level mean squared error $\mathcal{L}_{\text{mse}} = \|\boldsymbol{I} - \hat{\boldsymbol{I}}\|^2$, LPIPS [63] loss $\mathcal{L}_{\text{lpips}}$, and mask loss $\mathcal{L}_{\text{mask}} = \|\boldsymbol{S} - \hat{\boldsymbol{S}}\|^2$, where $\boldsymbol{S}$, $\hat{\boldsymbol{S}}$ are the predicted and ground-truth masks. We further employ a normal loss using the estimated mono normal by Omnidata [11], which is defined as the cosine similarity between the rendered normal $\boldsymbol{n}$ and estimated pseudo ground truths $\bar{\boldsymbol{n}}$: $\mathcal{L}_{\text{normal}} = 1 - (\boldsymbol{n} \cdot \bar{\boldsymbol{n}})$. To regularize the output geometry, we apply a smooth depth loss inspired by RegNeRF [35]: $\mathcal{L}_{\text{depth}}(i,j) = (d(i,j) - d(i,j+1))^2 + (d(i,j) - (d(i+1,j))^2$, where i, j indicate the pixel coordinate. For surface normal we instead rely on a bilateral smoothness loss similar to [5]. We found that this is crucial to getting high-frequency details and avoiding over-smoothed surfaces. For this loss we compute the image gradients of the input image $\nabla \boldsymbol{I}$ with a Sobel filter [17]. We then encourage the gradients of rendered normal $\nabla \boldsymbol{n}$ to be smooth if (and only if) the input image gradients $\nabla \boldsymbol{I}$ are smooth. The loss can be written as $\mathcal{L}_{\text{bilateral}} = e^{-3\nabla \boldsymbol{I}} \sqrt{1 + ||\nabla \boldsymbol{n}||}$. We also found that adding a spatial smoothness regularization on the albedo is beneficial: $\mathcal{L}_{\text{albedo}} = (\boldsymbol{c}_d(\boldsymbol{x}) - \boldsymbol{c}_d(\boldsymbol{x} + \boldsymbol{\epsilon}))^2$, where $\boldsymbol{c}_d$ denotes the albedo, $\boldsymbol{x}$ is a 3D surface point, and $\boldsymbol{\epsilon} \in \mathbb{R}^3$ is a normal distributed offset with a scale of 0.01. The overall objective is then defined as the weighted sum of these losses. All losses are applied in both coarse and fine stages, except that we only apply $\mathcal{L}_{\text{mask-sds}}$ in the last 200 iterations of the fine stage. We use an Adam optimizer [20] with a learning rate of 0.01 for both stages.

G Additional Ablative Analyses

We conduct additional ablative analyses of our 3D generation pipeline in this section.

G.1 SV3D Models

In Tab. 7, we compare the quantitative results using different SV3D models and training losses. Both 2D and 3D evaluation shows that SV3D^p is our best performing model, either for pure photometric reconstruction or SDS-based optimization.

Table 7: Ablative results of different SV3D models and training losses. We show that our SV3D^p model with Photo+SDS losses achieves the best 2D and 3D metrics.

Model	Training losses	LPIPS↓	PSNR↑	SSIM↑	MSE↓	CLIP-S↑	CD↓	3D IoU↑
SV3D^u	Photo	0.132	15.951	0.827	0.032	0.873	0.028	0.583
SV3D^u	Photo+SDS	0.133	15.957	0.834	0.031	0.871	0.027	0.589
SV3D^c	Photo	0.135	15.826	0.832	0.033	0.871	0.029	0.579
SV3D^c	Photo+SDS	0.132	16.373	0.834	0.027	0.870	0.027	0.584
SV3D^p	Photo	0.124	16.864	0.841	0.024	0.875	**0.024**	0.611
SV3D^p	Photo+SDS	**0.119**	**17.405**	**0.849**	**0.021**	**0.877**	**0.024**	**0.614**

G.2 Static v.s. Dynamic Orbits

We also compare the results using different camera orbits for 3D training in Tab. 8. The results show that using a dynamic orbit (sine-30) produces better 3D outputs compared to static orbit since it contains more information of the top and bottom views of the object. However, higher elevation (sine-50) tends to increase inconsistency between multi-view images, and thus resulting in worse 3D reconstruction. In our experiments, we find that setting the elevation within ± 30 degree generally leads to desirable 3D outputs.

Table 8: Ablative results of different reference orbits for 3D generation. We show that using a dynamic orbit (sine elevation) with moderate amplitude performs better than orbits with no or extreme elevation variations.

Training orbit	LPIPS↓	PSNR↑	SSIM↑	MSE↓	CLIP-S↑	CD↓	3D IoU↑
Static	0.125	16.821	0.848	0.025	0.864	0.028	0.610
Sine-30	**0.119**	**17.405**	0.849	**0.021**	**0.877**	**0.024**	**0.614**
Sine-50	0.123	17.057	**0.854**	0.025	0.873	0.026	0.609

G.3 Masked SDS Loss

Finally, we show the ablative results of our SDS loss in Tab. 9. We compare the results of 1) pure photometric losses, 2) with naive SDS loss (no masking), 3) with hard-masked SDS loss by thresholding the dot product of surface normal and camera viewing angle as visibility masks, and 4) with soft-masked SDS loss as described in the manuscript. Overall, adding SDS guidance from the SV3D model can improve the 2D metrics while maintaining similar 3D metrics. Our novel soft-masked SDS loss generally achieves the best results compared to other baselines.

Table 9: Ablative analyses of Masked SDS loss. Overall, our soft-masked SDS loss leads to higher-quality mesh outputs in terms of most 2D and 3D metrics.

Training losses	LPIPS↓	PSNR↑	SSIM↑	MSE↓	CLIP-S↑	CD↓	3D IoU↑
Photo	0.124	16.864	0.841	0.024	0.875	**0.024**	0.611
Photo+SDS (naive)	0.124	17.007	**0.850**	0.024	0.867	0.025	**0.615**
Photo+SDS (hard masked)	0.124	17.335	0.845	0.022	**0.877**	**0.024**	0.610
Photo+SDS (soft masked)	**0.119**	**17.405**	0.849	**0.021**	**0.877**	**0.024**	0.614

H Additional Visual Results

In this section, we show more results of novel view synthesis and 3D generation.

H.1 Novel View Synthesis

We show the additional NVS results on OmniObject3D [56] and real-world images in Fig. 13 and Fig. 14, respectively. The generated novel multi-view images by SV3D are more detailed and consistent compared to prior state-of-the-arts.

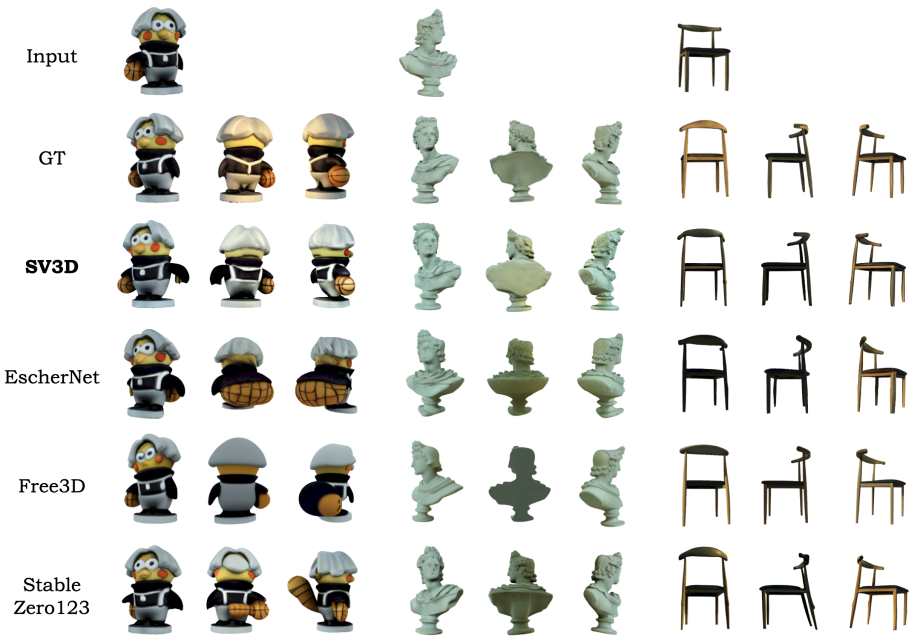

Fig. 13: NVS Results on OmniObject3D [56]. For each object, we show the input image and generated novel multi-view images by different methods. SV3D is able to produce images with consistent pose, color, and geometric details, which are closer the ground-truth renders.

Fig. 14: NVS Results on Real-World Images. Notice the consistent texture, geometry, and pose in SV3D NVS outputs compared to prior works.

H.2 3D Generation

We show the additional 3D generation results on OmniObject3D [56] and real-world images in Fig. 15 and Fig. 16, respectively. Thanks to the consistent multi-view images by SV3D and the novel Masked SDS loss, our 3D generations are detailed, high-fidelity, and generalizable to a wide range of images. Since Free3D [64] does not include a 3D generation method, we run our 3D pipeline on its generated multi-view images for fair 3D comparison.

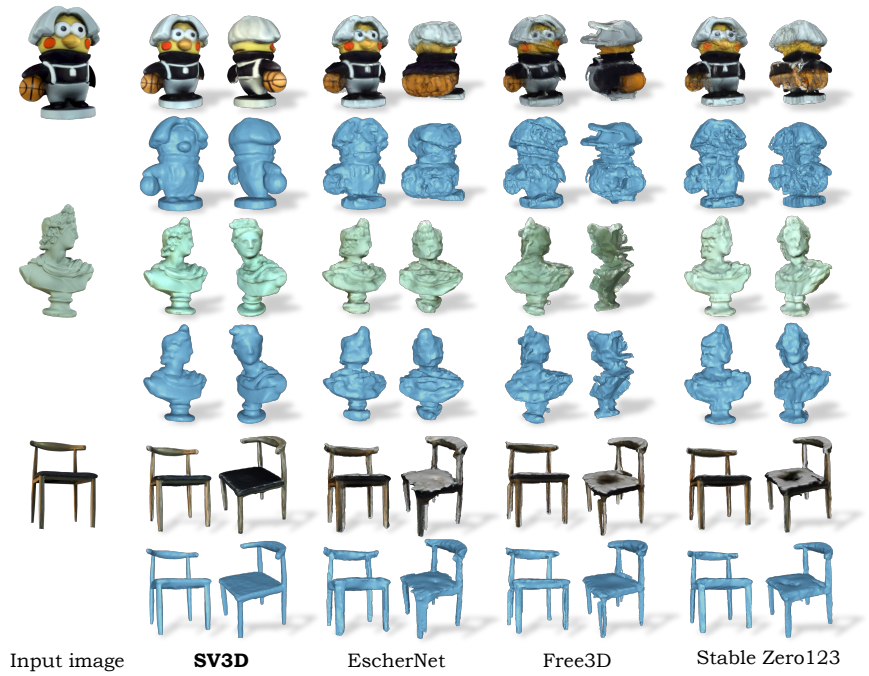

Fig. 15: Mesh Results on OmniObject3D [56]. Thanks to the consistent novel multi-view images generated by SV3D, our 3D meshes contain higher-fidelity details while having smooth surfaces.

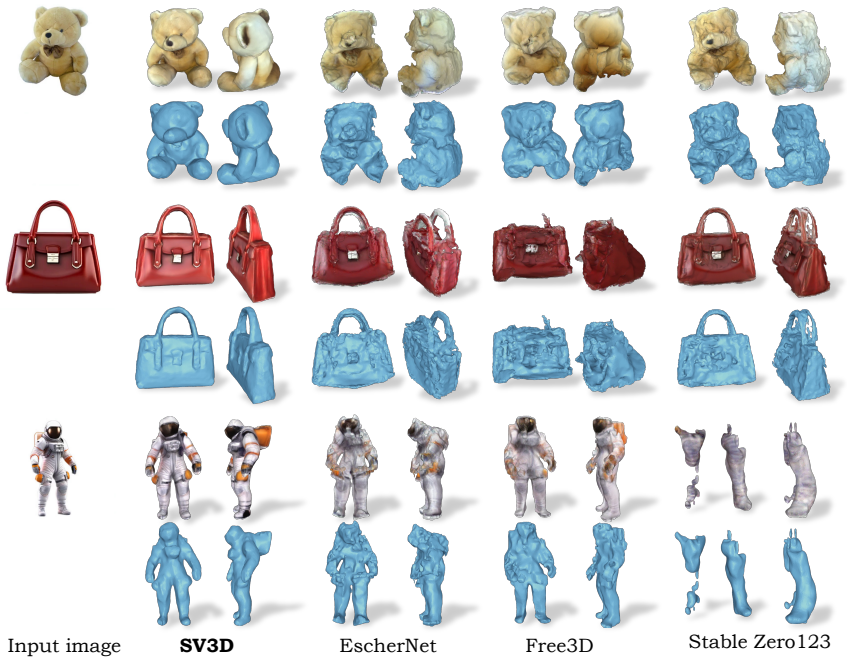

Fig. 16: Mesh Results on Real-World Images. Notice the accurate shape and details in our reconstructions even from the diverse images in-the-wild.

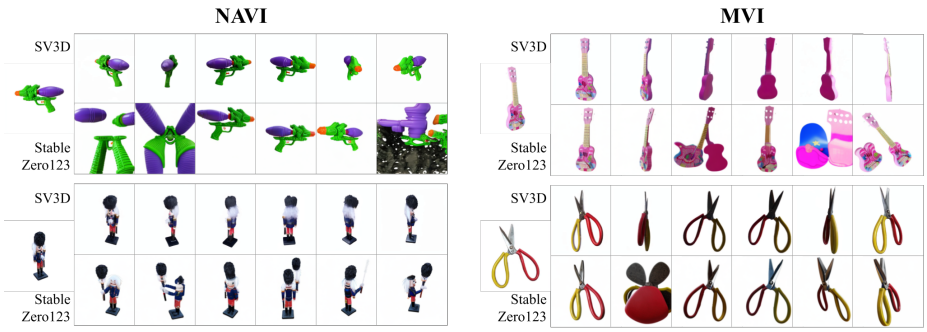

Fig. 17: NVS and 3D Generation Results on Real-World Images. Visual results on real-world images from the MVImgNet (Yu et al. CVPR'23) and NAVI (Jampani et al. NeurIPS'22) datasets. Compared to Stable Zero123, SV3D generates more consistent novel multi-view images, thus resulting in higher-quality 3D meshes with smooth surface and detailed texture.