IM-3D: Iterative Multiview Diffusion and Reconstruction for High-Quality 3D Generation

Luke Melas-Kyriazi [*1,2] **Iro Laina** [2] **Christian Rupprecht** [2] **Natalia Neverova** [1] **Andrea Vedaldi** [1] **Oran Gafni** [1]
Filippos Kokkinos [*1]

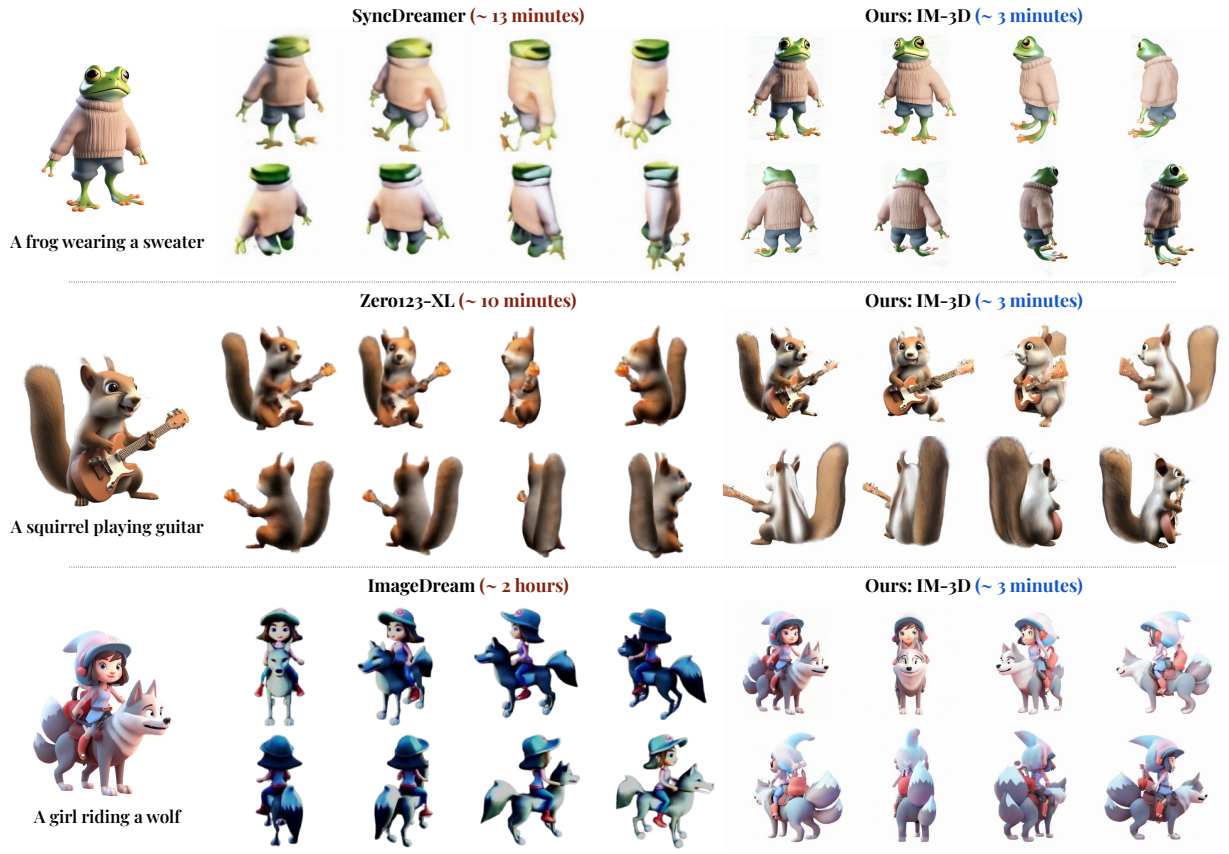

Figure 1: IM-3D generates high-quality and faithful 3D assets from text/image pair. It eschews Score Distillation Sampling (SDS) for robust 3D reconstruction of the output of a video diffusion model, tuned to generate a 360° video of the object.

Abstract

Most text-to-3D generators build upon off-the-shelf text-to-image models trained on billions of images. They use variants of Score Distillation Sampling (SDS), which is slow, somewhat unstable, and prone to artifacts. A mitigation is to fine-tune the 2D generator to be multi-view aware, which can help distillation or can be combined with reconstruction networks to output 3D objects directly. In this paper, we further explore the design space of text-to-3D models. We significantly improve multi-view generation by considering video instead of image generators. Combined with a 3D reconstruction algorithm which, by using Gaussian splatting, can optimize a robust image-based loss, we directly produce high-quality 3D outputs from the generated views. Our new method, IM-3D, reduces the number of evalua-

[*]Equal contribution [1]Meta [2]University of Oxford, Oxford, UK. Correspondence to: Filippos Kokkinos <fkokkinos@meta.com>, Luke Melas-Kyriazi <lukemk@robots.ox.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

tions of the 2D generator network 10-100×, resulting in a much more efficient pipeline, better quality, fewer geometric inconsistencies, and a high yield of usable 3D assets.

1. Introduction

All state-of-the-art open-world text-to-3D generators are built on top of off-the-shelf 2D generators trained on billions of images. This is necessary because there is not enough 3D training data to directly learn generators that can understand language and operate in an open-ended manner. However, the best way of building such models is still debated.

One approach is to perform 3D distillation by adopting Score Distillation Sampling (SDS) (Poole et al., 2023) or one of its variants. These models can work on top of nearly any modern 2D generator, but they require tens of thousands of evaluations of the 2D generator and can take hours to generate a single asset. They are also prone to artifacts and may fail to converge. Mitigating these shortcomings inspired a significant body of research (Wang et al., 2023b).

The fundamental reason for these limitations is that the underlying 2D generator is not 3D aware. SDS slowly makes the different views of the 3D object agree with the 2D model, which characterizes them independently of each other. Several authors (Shi et al., 2023b; Wang & Shi, 2023; Shi et al., 2023a) have shown that fine-tuning the 2D generator to understand the correlation between different views of the object significantly facilitates distillation. More recently, approaches such as (Li et al., 2023) avoid distillation entirely and instead just reconstruct the 3D object from the generated views. However, in order to compensate for defects in multi-view generation, they must incorporate very large 3D reconstruction networks. Ultimately, these approaches are many times faster than distillation, but quality is limited.

In this paper, we explore the benefits of further increasing the quality of multi-view generation and how this might affect the design space of future text-to-3D models. We are inspired by the fact that, in the limit, a 2D generator could output enough consistent views of the object to afford simple multi-view reconstruction, sidestepping distillation and reconstruction networks entirely.

To this end, we introduce IM-3D, a text-to-3D generation approach that leverages **I**terative **M**ultiview diffusion and reconstruction (Figures 1 and 2). IM-3D is based on significantly boosting the quality of the multi-view generation network by switching from a text-to-image to a text-to-video generator network. Specifically, we pick Emu Video (Girdhar et al., 2023), a video generator conditioned both on a reference image and a textual prompt. Our first contribution is to show that Emu Video can be fine-tuned, using a relatively small number of synthetic 3D assets, to generate directly up to 16 high-resolution consistent views (512×512) of the object. While Emu Video is in itself an iterative model based on diffusion, by adopting a fast sampling algorithm, the views can be generated in a few seconds and in a small number of iterations.

Our second contribution is to show that we can extract a high-quality 3D object by *directly* fitting a 3D model to the resulting views—without distillation or reconstruction networks—quickly and robustly. To do so, we rely on a 3D reconstruction algorithm based on Gaussian splatting (GS) (Kerbl et al., 2023). The importance of GS is that it affords fast differentiable rendering of the 3D object, which allows the use of image-based losses like LPIPS. The latter is key to bridging the small inconsistencies left by the 2D generator without requiring ad-hoc reconstruction models.

Third, we notice that, while this process results in mostly very good 3D models, some inconsistencies may still remain. We thus propose to close the loop and feed the 3D reconstruction back to the 2D generator. In order to do so, we simply render noised images of the 3D object and restart the video diffusion process from those. This approach is closer in spirit to SDS as it builds consensus progressively, but the feedback loop is closed two or three times per generated asset instead of tens of thousands of times.

There are many advantages to our approach. Compared to SDS, it reduces dramatically the number of evaluations of the 2D generator network. Using a fast sampler, generating the first version of the multi-view images requires only around 40 evaluations. Iterated generations are much shorter (as they start from a partially denoised result), at most doubling the total number of evaluations. This is a 10-100× reduction compared to SDS. The 3D reconstruction is also very fast, taking only a minute for the first version of the asset and a few seconds for the second or third. It also sidesteps typical SDS issues such as artifacts (e.g., saturated colors, Janus problem), lack of diversity (by avoiding mode seeking), and low yield (failure to converge). Compared to methods like (Li et al., 2023), IM-3D is slower but achieves much higher quality and does not require learning large reconstruction networks, offloading most of the work to 2D generation instead.

In a nutshell, our contribution is to show how video generator networks can improve consistent multi-view generators to a point where it is possible to obtain state-of-the-art and efficient text/image-to-3D results without distillation and without training reconstruction networks.

2. Related work

3D Distillation. 3D distillation is the process of extracting a 3D object from a 2D neural network trained to generate images from text or otherwise match them to text. For example, methods like DreamFields (Jain et al., 2022) do so starting

from the CLIP image similarity score. However, most recent methods build on diffusion-based image generators that utilize variants of the Score Distillation Sampling (SDS) loss introduced with DreamFusion (Poole et al., 2023). Fantasia 3D (Chen et al., 2023a) disentangles illumination from materials. Magic3D (Lin et al., 2022) reconstructs high-resolution texture meshes. RealFusion (Melas-Kyriazi et al., 2023) starts from a reference image and fine-tunes the prompt of a 2D generator to match it, distilling a 3D object afterwards. Make-it-3D (Tang et al., 2023b) also starts from a 2D image, combining SDS with a CLIP loss with respect to the reference image and a depth prior. HiFi-123 (Yu et al., 2023) uses DDIM inversion to obtain the code for the reference image. ATT3D (Lorraine et al., 2023) develops an amortized version of SDS, where several variants of the same object are distilled in parallel. HiFA (Zhu & Zhuang, 2023) reformulates the SDS loss and anneals the diffusion noise. DreamTime (Huang et al., 2023) also proposes to optimize the noise schedule. ProlificDreamer (Wang et al., 2023b), SteinDreamer (Wang et al., 2023a), Collaborative SDS (Kim et al., 2023) and Noise-free SDS (Katzir et al., 2023) improve the variance of the SDS gradient estimate. DreamGaussian (Tang et al., 2023a), GaussianDreamer (Yi et al., 2023) and (Chen et al., 2023c) apply Gaussian splatting to the SDS loss.

Methods using multi-view generation. Many methods have proposed to use multi-view generation to improve 3D generation. For multi-view generation, the most common approach is Zero-1-to-3 (Liu et al., 2023b), which fine-tunes the Stable Diffusion (SD) model to generate novel views of an object. Zero123++ (Shi et al., 2023a) further improves on this base model in various ways, including generating directly a grid of several multi-view images. Cascade-Zero123 (Chen et al., 2023b) proposes to apply two such models in sequence: the first to obtain approximate multiple views of the object, and the second to achieve better quality views conditioned on the approximate ones.

Magic123 (Qian et al., 2023) and DreamCraft3D (Sun et al., 2023) combine Zero-1-to-3 and SD. They start from a generated 2D image, extract depth and normals, and apply the RealFusion / DreamBooth technique to fine-tune the 2D diffusion model to generate different views of the object.

MVDream (Shi et al., 2023b) directly generates four fixed viewpoints of an object from a text prompt. Consistent123 (Weng et al., 2023) uses a different form of cross-view attention and generates several views sufficient for direct reconstruction. ConsistNet (Yang et al., 2023) introduces an explicit 3D pooling mechanism to exchange information between views. ImageDream (Wang & Shi, 2023) extends MVDream to start from a given input image and proposes a new variant of image conditioning compared to that of Zero-1-to-3. RichDreamer (Qiu et al., 2023) further learns to generate normals and separation between material and lighting.

Viewset Diffusion (Szymanowicz et al., 2023), Forward Diffusion (Tewari et al., 2023), SyncDreamer (Liu et al., 2023c) and DMV3D (Xu et al., 2023) denoise multiple views of the 3D object simultaneously to improve consistency. 3DGen (Gupta et al., 2023) learns a latent space to encode 3D objects using a VAE-like technique followed by a diffusion model. However, this approach is not very scalable as it requires training the model from scratch using 3D data. HexaGen3D (Mercier et al., 2024) extends 3DGen to use features from an SD model instead, thus increasing the scalability of the approach.

Two concurrent works are Stable Video Diffusion (Blattmann et al., 2023) and ViVid-1-to-3 (Kwak et al., 2023). Both use video models as multiview generators, but they only produce novel views, not 3D assets.

Non-SDS methods. Some text-to-3D methods perform "direct" 3D reconstruction on top of generated views without using SDS. One–2–3–45 (Liu et al., 2023a) compensates for the shortcomings of the multi-view generator by training a reconstruction network. Instant3D (Li et al., 2023) is similar but based on a much larger reconstruction model (Hong et al., 2023). Wonder3D (Long et al., 2023) further learns to generate multiple views of a given input image together with the corresponding normal maps, which are then used to reconstruct the 3D object. AGG (Xu et al., 2024) builds a single-image reconstruction network on top of Gaussian splatting. CAD (Wan et al., 2023) learns a 3D generator network from image samples using a 2D diffusion model, replacing the SDS loss with adversarial training.

Our approach also eschews the SDS loss but shows that it is possible to offload most of the modelling burden to the 2D generator network, utilizing a straightforward and efficient 3D reconstruction algorithm.

3. Method

We first describe our video-based multi-view generator network in Section 3.1 and its training data in Section 3.2, followed by a description of the robust 3D reconstruction module in Section 3.3 and of iterative refinement in Section 3.4. An overview of our method is shown in Figure 2.

3.1. Multi-view as video generation

Our multi-view generation model is based on fine-tuning an existing text-to-video (T2V) generator network Emu Video (Girdhar et al., 2023). First, it utilizes a text-to-image (T2I) model (Emu (Dai et al., 2023)) to generate an initial image $\mathbf{I}$ corresponding to the given textual prompt $\boldsymbol{p}$. Second, the image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ and the text prompt $\boldsymbol{p}$ are

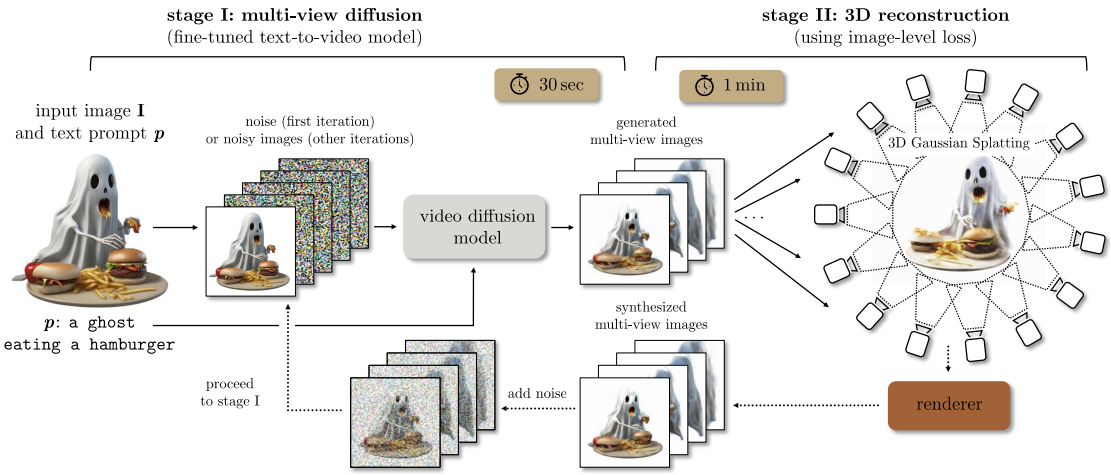

Figure 2: **Overview of IM-3D.** Our model starts from an input image (e.g., generated from a T2I model). It feeds the latter into an image-to-video diffusion model to generate a turn-table like video. The latter is plugged into 3D Gaussian Splatting to *directly* reconstruct the 3D object using image-based losses for robustness. Optionally, renders of the objects are generated and fed back to the video diffusion model, repeating the process for refinement.

fed into a second generator, which produces up to $K = 16$ frames of video $\mathbf{J} \in \mathbb{R}^{K\times 3\times H\times W}$, utilizing $\mathbf{I}$ as guidance for the first frame. Notice that, while the model is trained such that $\mathbf{I} \approx \mathbf{J}_1$, this is not an exact equality. Instead, the model draws a sample $\mathbf{J}$ from a learned conditional distribution $p(\mathbf{J}|\mathbf{I}, \boldsymbol{p})$, which allows it to slightly deviate from the input image to better fit in the generated video. An advantage of Emu Video compared to other video generators is that the video frames $\mathbf{J}$ are already high-quality and high-resolution without requiring sophisticated coarse-to-fine sampling schemes. It is architectured as a fine-tuned version of the original T2I Emu network with some modifications to account for the temporal dependencies between frames.

Starting from the pre-trained Emu Video model, we then fine-tune it to generate a particular kind of video, where the camera moves around a given 3D object, effectively generating *simultaneously* several views of it, in a turn-table-like fashion. In order to do so, we consider an internal dataset of synthetic 3D objects, further described in Section 3.2. This dataset provides us with training videos $\mathcal{J} = \{(\mathbf{J}_n, \mathbf{I}_n, \boldsymbol{p}_n)\}_{n=1}^{N}$, each containing $K = 16$ views of the object taken at fixed angular interval and a random but fixed elevation, the initial image $\mathbf{I}_n = [\mathbf{J}_n]_1$, and the textual prompt $\boldsymbol{p}_n$. The camera distance is fixed across all renders.

Differently from many prior multi-view generation networks, we *do not* pass the camera parameters to the model; instead, we use a fixed camera distance and orientation, randomizing only the elevation. The model simply learns to produce a set of views that follow this distribution.

Like most image and video generators, Emu Video is based on *diffusion* and implements a denoising neural network $\hat{\epsilon}(\mathbf{J}_t, t, \mathbf{I}, \boldsymbol{p})$ that takes as input a noised video $\mathbf{J}_t = \sqrt{1-\sigma_t^2}\mathbf{J} + \sigma_t\epsilon$, where $\epsilon \sim N(0, I)$ is Gaussian noise and $\sigma_t \in [0, 1]$ is the noise level, and tries to estimate the noise ϵ from it. The training uses the standard diffusion loss $\mathcal{L}_{\text{diff}}(\hat{\epsilon}|\mathbf{J}, \mathbf{I}, \boldsymbol{p}, t, \epsilon) = w_t^{\text{diff}} \cdot \|\hat{\epsilon}(\mathbf{J}_t, t, \mathbf{I}, \boldsymbol{p}) - \epsilon\|^2$ where $(\mathbf{J}, \mathbf{I}, \boldsymbol{p}) \in \mathcal{J}$ is a training video, ϵ is a Gaussian sample, t is a time step, also randomly sampled, and w_t is a corresponding weighing factor. To finetune Emu Video, we use $\mathcal{L}_{\text{diff}}$, but freeze all parameters except for the temporal convolutional and attention layers.

3.2. Data

The dataset $\mathcal{J}$ used to train our model consists of turn-table-like videos of synthetic 3D objects. Several related papers in multi-view generation also use synthetic data, taking Objaverse (Deitke et al., 2022) or Objaverse-XL (Deitke et al., 2023) as a source. Here, we utilize an in-house collection of 3D assets of comparable quality, for which we generate textual descriptions using an image captioning network.

Similar to prior works (Li et al., 2023), we use a subset of 100k assets selected for quality, as determined by the CLIP (Radford et al., 2021) alignment between rendered images and textual descriptions. Each video $\mathbf{J} \in \mathcal{J}$ is obtained by sampling one of the 100k assets, choosing a random elevation in $[0, \pi/4]$, and then placing the camera around the object at uniform ($2\pi/K$ degree) intervals.

3.3. Fast and robust reconstruction

To generate a 3D asset from a prompt $\boldsymbol{p}$, we first sample an image $\mathbf{I} \sim p(\mathbf{I}|\boldsymbol{p})$ from the Emu image model, followed by sampling a multi-view video $\mathbf{J} \sim p(\mathbf{J}|\mathbf{I}, \boldsymbol{p})$ from the fine-

tuned Emu Video model. Given the video $\mathbf{J}$, we then *directly* fit a 3D model G. While there are many possible choices for this model, here we use Gaussian splatting (Kerbl et al., 2023), a radiance field that uses a large number of 3D Gaussians to approximate the 3D opacity and color functions.

Given the 3D model G and a camera viewpoint Π, the *differentiable* Gaussian splatting renderer produces an image $\hat{\mathbf{I}} = \mathcal{R}(G, \Pi)$. Compared to other methods such as NeRF (Mildenhall et al., 2020), or even faster versions such as DVGO (Sun et al., 2022) or TensoRF (Chen et al., 2022), the key advantage of Gaussian splatting is the efficiency of the differentiable renderer, both in time and space, which allows rendering a *full* high-resolution image $\mathbf{I}$ at each training iteration instead of just selected pixels as in most prior works. Because of this fact, we can utilize *image-level* losses such as LPIPS (Zhang et al., 2018), i.e., $\mathcal{L}_{\text{LPIPS}}(\hat{\mathbf{I}}, \mathbf{I}) = \sum_{q=1}^{Q} \|w_q \odot (\Phi_q(\hat{\mathbf{I}}) - \Phi_q(\mathbf{I}))\|^2$ where $\Phi_q : \mathbb{R}^{3\times H\times W} \to \mathbb{R}^{C}$ is a family of Q patch-wise feature extractors implemented by the VGG-VD neural network (Simonyan & Zisserman, 2015). We also utilize a second image-based loss $\mathcal{L}_{\text{MS-SSIM}}$, the multi-scale structural similarity index measure (MS-SSIM) (Wang et al., 2003). Finally, we use a mask loss $\mathcal{L}_{\text{Mask}}$ with masks obtained using the method introduced in (Qin et al., 2022). In our ablation studies, we show the significant benefits of using these image-based losses rather than the standard pixel-wise RGB loss $\mathcal{L}_{\text{RGB}}(\hat{\mathbf{I}}, \mathbf{I}) = \|\hat{\mathbf{I}} - \mathbf{I}\|^2$. Our final loss is the weighted loss combination $\mathcal{L} = w_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + w_{\text{SSIM}}\mathcal{L}_{\text{SSIM}} + w_{\text{Mask}}\mathcal{L}_{\text{Mask}}$. The object G is thus reconstructed via direct optimization, i.e., $G^* = \text{argmin}_G \sum_{k=1}^{K} \mathcal{L}(\mathcal{R}(G, \Pi_k), [\mathbf{J}]_k)$ where $[\mathbf{J}]_k$ denotes the k-th image in the video.

3.4. Fast sampling and iterative generation

The SDS loss can be seen as a way to bridge the gap between image generators that are unaware of 3D objects and their 3D reconstructions, absorbing multi-view consistency defects in the generation. Because our model is rather view-consistent from the outset, and because we can use robust reconstruction losses, the SDS loss is unnecessary. Instead, given a prompt $\boldsymbol{p}$, we simply generate an image $\mathbf{I}$, followed by video $\mathbf{J}$, and then fit a 3D object G to the latter.

One main advantage is that this *dramatically* reduces the number of model evaluations compared to using the SDS loss. Optimizing the SDS loss is (approximately) the same as ascending the score, i.e., the gradient $\nabla \log p(\mathbf{J}_t|\mathbf{I}, \boldsymbol{p})$ of the log distribution over noised videos (or images), so optimizing the asset G can be seen as a form of multi-view mode seeking. The score is obtained from the same network $\hat{\epsilon}$. However, despite the conditioning on a specific textual prompt $\boldsymbol{p}$ and input view $\mathbf{I}$, the sampled distribution is rather wide, requiring a very large number (thousands) of iterations to converge to a mode; furthermore, regressing to a mode reduces the diversity and quality of the output.

In our case, the network $\hat{\epsilon}$ is used to generate directly a *single* video $\mathbf{J}$, which is then reconstructed without further invocations to the model. Because the video $\mathbf{J}$ is already sufficiently view-consistent, the 3D reconstruction converges quickly to a good solution. Furthermore, we can adopt fast stochastic ODE solvers such as DPM++ (Lu et al., 2022) to further reduce the number of model evaluations to obtain the video in the first place. Overall, compared to using the SDS loss, the number of model evaluations is reduced by a factor 10-100$\times$ (see the Appendix for additional analysis).

Despite the overall consistency of generated videos $\mathbf{J}$, they are still not perfect. We thus additionally compensate for such inconsistencies during model fitting, but still without resorting to the SDS loss. Instead, we alternate 3D reconstruction and video generation. To do so, once the first video $\mathbf{J}$ and corresponding 3D model G^* are obtained, we use the latter to generate a video $\hat{\mathbf{J}} = \mathcal{R}(G^*, \Pi)$ using the 3D renderer, sample an intermediated noised video $\hat{\mathbf{J}}_t$ by adding noise to it as shown above and then invoking the video generator again to obtain a denoised and updated video $\mathbf{J}'$.

We iterate this process two times. This is vastly faster than using the SDS loss while still being highly robust.

4. Experiments

Our method generates 3D objects from a textual description $\boldsymbol{p}$ and a reference image $\mathbf{I}$. In order to compare to prior work, we consider, in particular, the set of textual prompts from (Shi et al., 2023b), which are often used for evaluation.

Given an input image and prompt $(\mathbf{I}, \boldsymbol{p})$, previous methods either *synthesize* a multi-view image sequence $\mathbf{J}$ (usually by means of a generator network) or output a 3D model, or both. We compare the quality of the produced artifacts visually, utilizing the image sequence $\mathbf{J}$ directly, or corresponding *renders* $\hat{\mathbf{J}}$ of the 3D model. In general, we can expect the quality and faithfulness of $\mathbf{J}$ to be better than that of $\hat{\mathbf{J}}$ because the generated image sequence needs not to be perfectly view-consistent. On the other hand, the renders $\hat{\mathbf{J}}$ from the 3D model are consistent by construction, but may be blurrier than $\mathbf{J}$, or contain other defects.

4.1. Comparison to the state-of-the-art

In this section, we compare IM-3D to relevant state-of-the-art methods in the literature, including MVDream (Shi et al., 2023b), Zero123XL (Deitke et al., 2023), Magic123 (Qian et al., 2023), SyncDreamer (Liu et al., 2023c), ImageDream (Wang & Shi, 2023), LRM (Hong et al., 2023) and One2345++ (Liu et al., 2023a). For LRM, since no public models are available, we utilize the open-source OpenLRM (He & Wang, 2023). For One2345++, which is only available via a web interface, we manually upload each

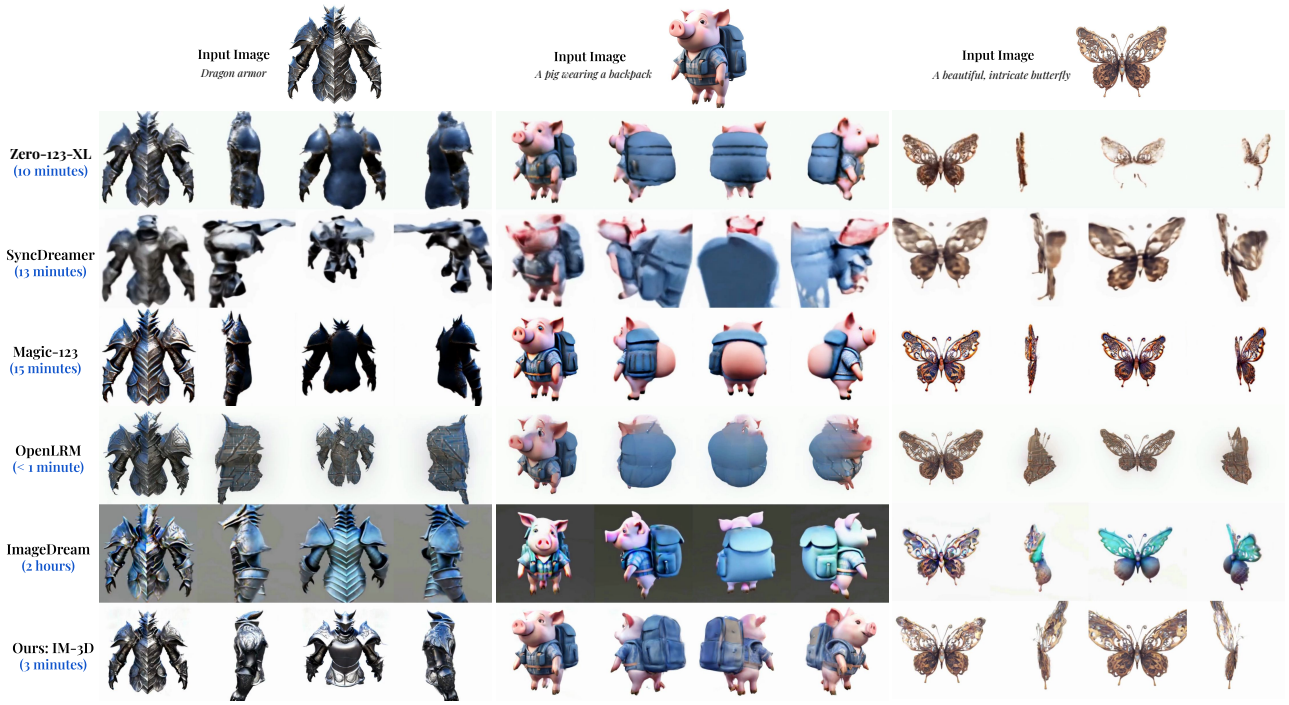

Figure 3: **Qualitative Comparisons.** Our method IM-3D (last row) and others for the same text/image prompt pairs. For IM-3D, we show the final GS reconstruction (which guarantees multi-view consistency). We match the input image faithfully and obtain high-quality, detailed reconstructions in just 3 minutes. Faster methods such as OpenLRM are also much worse.

Figure 4: **A visualization of reconstruction quality over multiple iterations of multiview diffusion and reconstruction.** We compare the initial reconstructions obtained by our model (i.e. the result of training on our initial generated videos) to the result after one iteration of reconstruction and refinement. We see that although the initial reconstructions have reasonable shapes, they lack fine-grained details due to small inconsistencies in the generated multiview images. After one iteration of noising, denoising, and reconstruction, our method resolves these inconsistencies and produces 3D assets with significantly higher levels of detail (as highlighted by the red circles above).

Table 1: **Faithfulness to the textual and visual prompts of image sequences synthesised or rendered by various methods**. Assessed on the prompt list from (Wang & Shi, 2023; Shi et al., 2023b).

		synthesized view		re-rendered view	
model	Time (min)	CLIP (Text)	CLIP (Image)	CLIP (Text)	CLIP (Image)
SDXL (Podell et al., 2023) [upper bound]	*0.03*	*33.33*	*100*	—	—
MVDream (Shi et al., 2023b)	72	31.26 ±2.9	76.44 ±6.5	30.63 ±2.7	76.94 ±5.2
Zero123XL (Deitke et al., 2023)	10	19.58 ±1.3	60.29 ±5.8	29.06 ±3.3	81.33 ±6.9
Magic123 (Qian et al., 2023)	15	—	—	29.51 ±4.7	84.14 ±10.2
SyncDreamer (Liu et al., 2023c)	13	27.76 ±3.0	77.26 ±7.2	26.22 ±3.4	74.95 ±6.6
ImageDream (Wang & Shi, 2023)	120	31.08 ±3.4	85.39 ±5.8	30.73 ±2.3	83.77 ±5.2
OpenLRM (Hong et al., 2023)	**0.17**	—	—	29.75 ±3.2	83.08 ±9.5
One2345++ (Liu et al., 2023a)	0.75	—	—	29.71 ±2.3	83.78 ±6.4
IM-3D (ours)	3	**31.92** ±1.6	**92.38** ±5.1	**31.66** ±1.7	**91.40** ±5.5

Figure 5: **Human evaluation.** We perform human evaluation of IM-3D vs state-of-the-art in Image-to-3D and Text-to-3D. Human raters preferred IM-3D to all competitors with regard to both generation quality and faithfulness, often by a large margin.

image in the evaluation set. We carry out both quantitative and qualitative comparisons using the set of prompts and images from (Shi et al., 2023b; Wang & Shi, 2023).

Quantitative comparison. Table 1 provides a quantitative comparison of our method to others. We adopt the same metrics as (Shi et al., 2023b; Wang & Shi, 2023), which are based on the CLIP (Radford et al., 2021) similarity scores. Specifically, we utilize the ability of CLIP to embed text and images in the same space. We then use the embeddings to compare the textual prompt $\boldsymbol{p}$ and the image prompt $\mathbf{I}$ to the images $\mathbf{J}$ of the object (either synthesized or rendered). A high CLIP similarity score means high faithfulness to the prompt. As an upper bound, we also report the CLIP scores of the prompt images $\mathbf{I}$, which were generated using the SDXL (Podell et al., 2023) model.

The key takeaway from Table 1 is that IM-3D outperforms all others in terms of both textual and visual faithfulness. This is true for both the image sequences $\mathbf{J}$ output by the video generator as well as the renders $\hat{\mathbf{J}}$ from the fitted 3D GS models G. IM-3D is particularly strong when it comes to visual faithfulness, which also means that the images we generate are of a quality comparable to the input image $\mathbf{I}$. Additionally, our method requires significantly less time than most (3 minutes vs hours for some models).

Table 2: **Ablation on the importance of loss terms and 3D representation during the fitting stage.**

Loss / Representation	CLIP (Text)	CLIP (Image)
IM-3D (ours)	**31.66** ±1.7	**91.40** ±5.5
- $\mathcal{L}_{\text{LPIPS}}$	29.38 ±2.1	84.71 ±6.4
- $\mathcal{L}_{\text{RGB}}$ instead of $\mathcal{L}_{\text{LPIPS}}$	29.67 ±2.0	84.99 ±5.9
- $\mathcal{L}_{\text{SSIM}}$	31.53 ±1.8	90.64 ±5.7
- $\mathcal{L}_{\text{Mask}}$	31.43 ±1.9	90.14 ±6.0
w/ NeRF instead of GS	30.42 ±2.1	87.37 ±5.4

Human evaluation. Automated metrics for the evaluation of generative models are not fully representative of the value of the output in applications. Thus, we also conduct a human study. We ask annotators to evaluate our model against a competitor based on (1) Image Alignment and (2) 3D quality. We present to annotators the outputs of two different methods, rendered as 360° videos, and ask them to indicate a preference based on these two criteria. Table 1 shows the win rate when comparing our methods against others. Our method surpasses in performance all other baselines in both studies indicating that the proposed method produces high-quality 3D results that closely align with the image prompt. Further details are provided in the Appendix.

4.2. Ablations

Effect of iterative refinements. In Figure 4, we demonstrate the efficacy of our proposed iterative refinement process. Our model's initial reconstructions (derived from training on our initially generated videos) are compared to the outcome following a single iteration of multiview diffusion

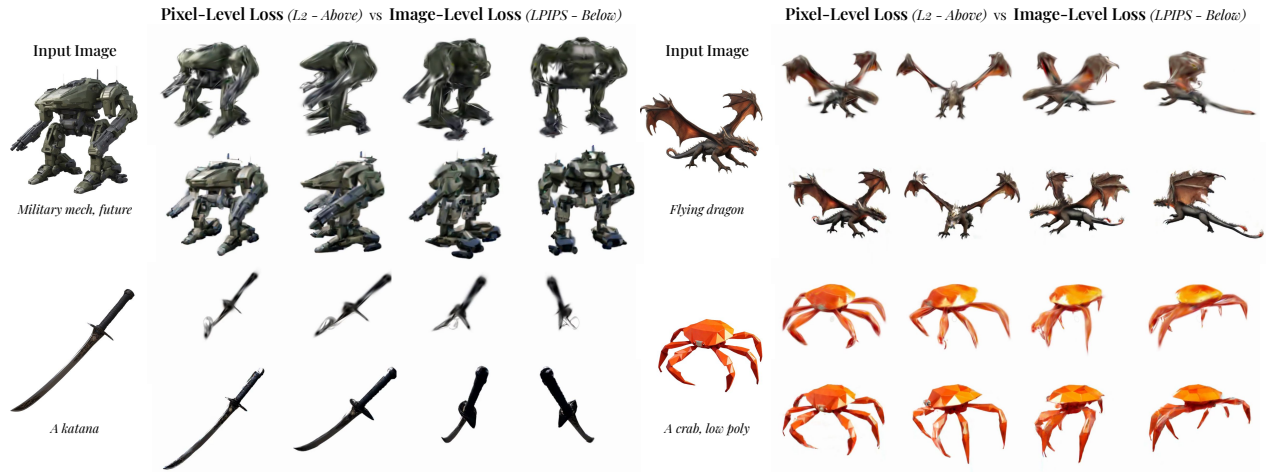

Figure 6: **Reconstruction Quality with Pixel-Level and Image-Level Losses.** We find that image-level losses are crucial to the success of our method. With pixel-level losses such as the L2 loss, small inconsistencies in the generated images are effectively averaged together, resulting in unnatural and blurry-looking reconstructions.

and reconstruction. While the initial reconstructions exhibit satisfactory shapes, they miss out on intricate details due to minor inconsistencies in the initial multiview images. In a few instances, some parts of these initial reconstructions look as if two copies of a shape have been superimposed upon one another as the reconstruction process tries to satisfy two inconsistent views. However, our technique rectifies these discrepancies with one iteration of denoising and reconstruction, significantly enhancing the level of detail.

Image-Level Losses. In Table 2 and Figure 6, we compare the results of optimization with pixel-level and image-level loss functions. We find that image-level losses are central to our method's ability to generate high-quality 3D assets. The use of pixel-level losses such as L2 loss is detrimental, as minor inconsistencies in the multiview images are emphasized by the optimization process and effectively averaged together. This averaging results in a low CLIP score (29.67 vs 31.66 for LPIPS) as well as blurry and unnatural generations.

Comparing 3D Representations The last line of Table 2 provides a comparison of 3D representations, showing the effect of using NeRF as an underlying 3D representation rather than Gaussian splatting (GS). We find that the visual quality of models generated using NeRF is slightly worse than GS. The true benefit of GS is that it is much faster and much more memory-efficient; training with GS takes 3 minutes, whereas training with NeRF takes 40 minutes. Additionally, the memory-efficient nature of Gaussian splatting makes it easy to render at our diffusion model's native resolution of 512px, whereas for NeRF one has to use ray microbatching or optimize at a lower resolution.

Table 3: **Ablation on Using Fewer Frames.** We show quantitative performance when performing our reconstruction and generation using fewer frames.

# Frames	CLIP (Text)	CLIP (Image)
16	**31.66** ±1.7	**91.40** ±5.5
8	31.38 ±1.8	90.06 ±6.3
4	30.06 ±2.6	86.96 ±8.6

Using Fewer Frames Differently from the vast majority of other diffusion-based text-to-3D and image-to-3D approaches, which generate only 1-4 frames, IM-3D generates 16 frames simultaneously. We demonstrate the significance of this in Table 3, finding that our quantitative performance improves as we increase the number of generated frames.

4.3. Limitations

The fine-tuned video generator is generally very view-consistent, but it still has limitations. One interesting failure case is that for highly dynamic subjects (e.g., horses, which are often captured running), the model sometimes renders spurious animations (e.g., walking or galloping) despite our fine-tuning, which is problematic for 3D reconstruction. This occurs more often when the prompt contains verbs describing motion; see the Appendix for an example.

5. Conclusions

In this work, we have shown that starting from a video generator network instead of an image generator can result in better multi-view generation to a point where it can impact the design of future text-to-3D models. In fact, we have shown that the quality is sufficient to eschew distillation losses like SDS as well as large reconstruction networks. Instead, one can simply fit the 3D object to the generated

views using a robust image-based loss. Reconstruction can be further alternated with refining the target video, quickly converging to a better 3D object with minimal impact on efficiency. Compared to works that rely on SDS, our approach significantly reduces the number of evaluations of the 2D generator network, resulting in a faster and more memory-efficient pipeline without compromising on quality.

Impact Statement

Our work uses Generative AI, whose potential impacts are and have been extensively discussed in the academic, business and public spheres. Our work does not change these issues qualitatively. The Emu models (Dai et al., 2023) were explicitly designed with fairness and safety in mind, and fine-tuning them on curated 3D models is likely to further reduce the potential for harm.

References

Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., Jampani, V., and Rombach, R. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv.cs*, abs/2311.15127, 2023.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proc. ICCV*, 2021.

Chen, A., Xu, Z., Geiger, A., Yu, J., and Su, H. TensoRF: Tensorial radiance fields. In *arXiv*, 2022.

Chen, R., Chen, Y., Jiao, N., and Jia, K. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv.cs*, abs/2303.13873, 2023a.

Chen, Y., Fang, J., Huang, Y., Yi, T., Zhang, X., Xie, L., Wang, X., Dai, W., Xiong, H., and Tian, Q. Cascade-Zero123: One image to highly consistent 3D with self-prompted nearby views. *arXiv.cs*, abs/2312.04424, 2023b.

Chen, Z., Wang, F., and Liu, H. Text-to-3D using Gaussian splatting. *arXiv*, (2309.16585), 2023c.

Cline, D. B. H. Admissibile kernel estimators of a multivariate density. *The Annals of Statistics*, 16(4), 1988.

Dai, X., Hou, J., Ma, C., Tsai, S. S., Wang, J., Wang, R., Zhang, P., Vandenhende, S., Wang, X., Dubey, A., Yu, M., Kadian, A., Radenovic, F., Mahajan, D., Li, K., Zhao, Y., Petrovic, V., Singh, M. K., Motwani, S., Wen, Y., Song, Y., Sumbaly, R., Ramanathan, V., He, Z., Vajda, P., and Parikh, D. Emu: Enhancing image generation models using photogenic needles in a haystack. *CoRR*, abs/2309.15807, 2023.

Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., and Farhadi, A. Objaverse: A universe of annotated 3d objects. *arXiv.cs*, abs/2212.08051, 2022.

Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S. Y., VanderBilt, E., Kembhavi, A., Vondrick, C., Gkioxari, G., Ehsani, K., Schmidt, L., and Farhadi, A. Objaverse-XL: A universe of 10M+ 3D objects. *CoRR*, abs/2307.05663, 2023.

Girdhar, R., Singh, M., Brown, A., Duval, Q., Azadi, S., Rambhatla, S. S., Shah, A., Yin, X., Parikh, D., and Misra, I. Emu video: Factorizing text-to-video generation by explicit image conditioning. *CoRR*, abs/2311.10709, 2023.

Gupta, A., Xiong, W., Nie, Y., Jones, I., and Oguz, B. 3DGen: Triplane latent diffusion for textured mesh generation. *corr*, abs/2303.05371, 2023.

He, Z. and Wang, T. Openlrm: Open-source large reconstruction models. https://github.com/3DTopia/OpenLRM, 2023.

Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., and Tan, H. LRM: Large reconstruction model for single image to 3D. *arXiv*, 2023.

Huang, Y., Wang, J., Shi, Y., Qi, X., Zha, Z., and Zhang, L. Dreamtime: An improved optimization strategy for text-to-3D content creation. *CoRR*, abs/2306.12422, 2023.

Jain, A., Mildenhall, B., Barron, J. T., Abbeel, P., and Poole, B. Zero-shot text-guided object generation with dream fields. In *Proc. CVPR*, 2022.

Katzir, O., Patashnik, O., Cohen-Or, D., and Lischinski, D. Noise-free score distillation. *arXiv.cs*, abs/2310.17590, 2023.

Kerbl, B., Kopanas, G., Leimkühler, T., and Drettakis, G. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *Proc. SIGGRAPH*, 42(4), 2023.

Kim, S., Lee, K., Choi, J. S., Jeong, J., Sohn, K., and Shin, J. Collaborative score distillation for consistent visual synthesis. *arXiv.cs*, abs/2307.04787, 2023.

Kwak, J., Dong, E., Jin, Y., Ko, H., Mahajan, S., and Yi, K. M. ViVid-1-to-3: Novel view synthesis with video diffusion models. *arXiv.cs*, abs/2312.01305, 2023.

Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., and Bi, S. Instant3D: Fast text-to-3D with sparse-view generation and large reconstruction model. *arXiv*, 2023. URL https://instant-3d.github.io.

Lin, C., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M., and Lin, T. Magic3D: High-resolution text-to-3d content creation. *arXiv.cs*, abs/2211.10440, 2022.

Liu, M., Xu, C., Jin, H., Chen, L., T, M. V., Xu, Z., and Su, H. One-2-3-45: Any single image to 3D mesh in 45 seconds without per-shape optimization. *arXiv.cs*, abs/2306.16928, 2023a.

Liu, R., Wu, R., Hoorick, B. V., Tokmakov, P., Zakharov, S., and Vondrick, C. Zero-1-to-3: Zero-shot one image to 3d object. *CoRR*, abs/2303.11328, 2023b.

Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., and Wang, W. SyncDreamer: Generating multiview-consistent images from a single-view image. *arXiv*, (2309.03453), 2023c.

Long, X., Guo, Y., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S., Habermann, M., Theobalt, C., and Wang, W. Wonder3D: Single image to 3D using cross-domain diffusion. *arXiv.cs*, abs/2310.15008, 2023.

Lorraine, J., Xie, K., Zeng, X., Lin, C., Takikawa, T., Sharp, N., Lin, T., Liu, M., Fidler, S., and Lucas, J. ATT3D: amortized text-to-3D object synthesis. *arXiv.cs*, abs/2306.07349, 2023.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Proc. NeurIPS*, 2022.

Melas-Kyriazi, L., Rupprecht, C., Laina, I., and Vedaldi, A. RealFusion: 360 reconstruction of any object from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. URL https://lukemelas.github.io/realfusion/.

Mercier, A., Nakhli, R., Reddy, M., and Yasarla, R. HexaGen3D: Stablediffusion is just one step away from fast and diverse text-to-3D generation. *arXiv*, 2024.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. ECCV*, 2020.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. SDXL: improving latent diffusion models for high-resolution image synthesis. *arXiv.cs*, abs/2307.01952, 2023.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. DreamFusion: Text-to-3D using 2D diffusion. In *Proc. ICLR*, 2023.

Qian, G., Mai, J., Hamdi, A., Ren, J., Siarohin, A., Li, B., Lee, H., Skorokhodov, I., Wonka, P., Tulyakov, S., and Ghanem, B. Magic123: One image to high-quality 3D object generation using both 2D and 3D diffusion priors. *arXiv.cs*, abs/2306.17843, 2023.

Qin, X., Dai, H., Hu, X., Fan, D.-P., Shao, L., and Gool, L. V. Highly accurate dichotomous image segmentation. In *ECCV*, 2022.

Qiu, L., Chen, G., Gu, X., Zuo, Q., Xu, M., Wu, Y., Yuan, W., Dong, Z., Bo, L., and Han, X. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. *arXiv.cs*, abs/2311.16918, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *Proc. ICML*, volume 139, pp. 8748–8763, 2021.

Shen, T., Gao, J., Yin, K., Liu, M.-Y., and Fidler, S. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021.

Shi, R., Chen, H., Zhang, Z., Liu, M., Xu, C., Wei, X., Chen, L., Zeng, C., and Su, H. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv.cs*, abs/2310.15110, 2023a.

Shi, Y., Wang, P., Ye, J., Long, M., Li, K., and Yang, X. MVDream: Multi-view diffusion for 3D generation. *arXiv.cs*, abs/2308.16512, 2023b.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015.

Sun, C., Sun, M., and Chen, H. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proc. CVPR*, 2022.

Sun, J., Zhang, B., Shao, R., Wang, L., Liu, W., Xie, Z., and Liu, Y. DreamCraft3D: Hierarchical 3D generation with bootstrapped diffusion prior. *arXiv.cs*, abs/2310.16818, 2023.

Szymanowicz, S., Rupprecht, C., and Vedaldi, A. Viewset diffusion: (0-)image-conditioned 3D generative models from 2D data. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. URL https://szymanowiczs.github.io/viewset-diffusion.

Tang, J., Ren, J., Zhou, H., Liu, Z., and Zeng, G. DreamGaussian: Generative gaussian splatting for efficient 3D content creation. *arXiv*, (2309.16653), 2023a.

Tang, J., Wang, T., Zhang, B., Zhang, T., Yi, R., Ma, L., and Chen, D. Make-it-3d: High-fidelity 3d creation from A single image with diffusion prior. *arXiv.cs*, abs/2303.14184, 2023b.

Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., and Liu, Z. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.

Tewari, A., Yin, T., Cazenavette, G., Rezchikov, S., Tenenbaum, J. B., Durand, F., Freeman, W. T., and Sitzmann, V. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *arXiv.cs*, abs/2306.11719, 2023.

Tochilkin, D., Pankratz, D., Liu, Z., Huang, Z., , Letts, A., Li, Y., Liang, D., Laforte, C., Jampani, V., and Cao, Y.-P. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

Wan, Z., Paschalidou, D., Huang, I., Liu, H., Shen, B., Xiang, X., Liao, J., and Guibas, L. Cad: Photorealistic 3d generation via adversarial distillation. *arXiv*, (2312.06663), 2023.

Wang, P. and Shi, Y. ImageDream: Image-prompt multi-view diffusion for 3D generation. *arXiv.cs*, abs/2312.02201, 2023.

Wang, P., Fan, Z., Xu, D., Wang, D., Mohan, S., Iandola, F., Ranjan, R., Li, Y., Liu, Q., Wang, Z., and Chandra, V. SteinDreamer: Variance reduction for text-to-3d score distillation via stein identity. *arXiv*, 2023a.

Wang, Z., Simoncelli, E., and Bovik, A. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, 2003. doi: 10.1109/ACSSC.2003.1292216.

Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., and Zhu, J. ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. *arXiv.cs*, abs/2305.16213, 2023b.

Weng, H., Yang, T., Wang, J., Li, Y., Zhang, T., Chen, C. L. P., and Zhang, L. Consistent123: Improve consistency for one image to 3d object synthesis. *arXiv*, 2023.

Xu, D., Yuan, Y., Mardani, M., Liu, S., Song, J., Wang, Z., and Vahda, A. AGG: Amortized generative 3d gaussians for single image to 3d. *arXiv.cs*, 2024.

Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzstein, G., Xu, Z., and Zhang, K. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model, 2023.

Yang, J., Cheng, Z., Duan, Y., Ji, P., and Li, H. Consistnet: Enforcing 3d consistency for multi-view images diffusion. *arXiv.cs*, abs/2310.10343, 2023.

Yi, T., Fang, J., Wu, G., Xie, L., Zhang, X., Liu, W., Tian, Q., and Wang, X. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arXiv.cs*, abs/2310.08529, 2023.

Yu, W., Yuan, L., Cao, Y., Gao, X., Li, X., Quan, L., Shan, Y., and Tian, Y. HiFi-123: Towards high-fidelity one image to 3d content generation. *arXiv.cs*, abs/2310.06744, 2023.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. CVPR*, pp. 586–595, 2018.

Zhu, J. and Zhuang, P. HiFA: High-fidelity text-to-3D with advanced diffusion guidance. *CoRR*, abs/2305.18766, 2023.

A. Appendix

A.1. Training details

In line with (Girdhar et al., 2023), we maintain the spatial convolutional and attention layers of Emu Video, fine-tuning only the temporal layers. We minimize the standard diffusion loss over a span of 5 days, employing 80 A100 GPUs with a total batch size of 240 and a learning rate of *1e-5*. Our findings indicate that prolonged training effectively counters the network's inclination to generate 360 videos of deforming objects, given that the initialization is a video generation model. Contrary to MVDream (Shi et al., 2023b) and Instant3D (Li et al., 2023), we observe no degradation in texture quality with extended training. This can be ascribed to the fact that the spatial layers remain static and the network is image-conditioned, necessitating that the generated 360 video retain the high-frequency texture elements of the input.

For Gaussian fitting, we initialize 5000 points at the center of the 3D space, and densify and prune the Gaussians every 50 iterations. We conduct optimization for 1200 iterations and execute Emu Video twice for 10 iterations each using the DPM solver (Lu et al., 2022) during this process, repeating this every 500 iterations. Empirically, we found that setting the weights to $w_{\text{LPIPS}} = 10$, $w_{\text{SSIM}} = 0.2$ and $w_{\text{Mask}} = 1$ yields the best results during the fitting stage.

Prior to fitting, we need to estimate the elevation of the very first generated video. To that end, we trained an elevation estimator on top of DINO (Caron et al., 2021) features using the 100k 3D renderings. The network averages the features of 4 uniformly distributed frames and uses a 2-layer MLP to regress the elevation in radians.

A.2. Human evaluation

In our study, we employed the prompt set delineated in (Shi et al., 2023b) to conduct a human annotation evaluation via Amazon Mechanical Turk (AMT). The task assigned to the annotators involved choosing between two 3D assets, both rendered as 360° videos, with one of the assets being the output of our proposed method. To ensure a robust and unbiased evaluation, we randomized the presentation order of the methods for each question. Each question was assessed by five annotators, and we reported the consensus opinion. Since each question corresponds to a triplet of (competing method, 3D reconstruction, and quality/faithfulness), this is a total of 5 annotators $\times$ (5 methods $\times$ 39 reconstructions $\times$ 2 question types) $= 1950$ annotations. We instructed the annotators to overlook any disparities in background colors, as normalizing all methods to yield the same scale is a non-trivial task.

A.3. Further information on network efficiency

	ProlificDreamer	MVDream	ImageDream	Zero123XL	SyncDreamer	IM-3D
Number of network calls	320000	20000	25000	1200	200	80

Table 4: **Number of diffusion network calls to generate one 3D asset.** The proposed method, IM-3D, requires only a fraction of the model evaluations to compute a 3D asset.

In Table 4, we present the number of diffusion model forward passes used to reconstruct a single 3D object for various 3D generation methods. Whereas some other methods require thousands or tens of thousands of iterations, our method requires less than one hundred.

A.4. Further comparisons with recent methods

In Appendix A.4, we present additional comparisons with recent and concurrently-developed methods (Long et al., 2023; Tochilkin et al., 2024; Tang et al., 2024; Kwak et al., 2023).

A.5. Results on human portraits

Our training data does not contain humans; however, in Figure 7, we show the results of our model on human portrait reconstruction. We find that our model performs well despite the inputs being outside of the training distribution.

Figure 7: **Qualitative examples on generated human portraits.** The input images for the figure are generated by a 2D image generation model.

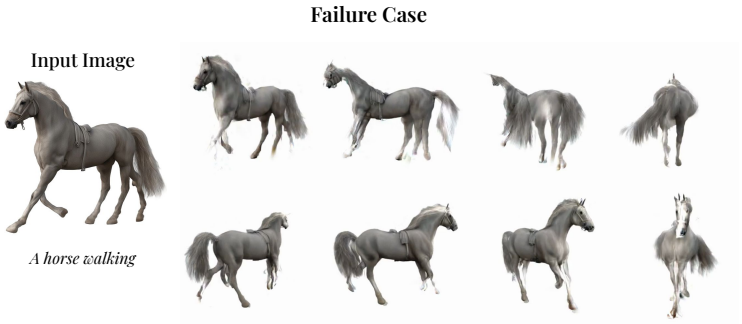

Figure 8: **Visualisation of a failure case.** In this case, our finetuned video network generated an animated video of a horse walking rather than a static video. As a result, the 3D reconstruction process produces erroneous geometry (e.g. the head of the horse is barely visible from some views).

	CLIP-Text	CLIP-Image
Wonder3D (Long et al., 2023)	29.0	79.8
TripoSR (Tochilkin et al., 2024)	29.0	81.5
LGM (Tang et al., 2024)	30.1	87.9
Vivid-1-to-3 (Kwak et al., 2023)	29.5	87.1
Ours	**31.3**	**90.5**

Table 5: **Further comparison results.** We outperform recent and concurrent methods.

Figure 9: **Visualisation of Meshes.** We convert our Gaussian Splatting representation to DMTet (Shen et al., 2021) using marching cubes and optimize the resulting meshes using our iterative multiview diffusion and reconstruction process.

A.6. Failure Cases

As described in the limitations section of the main paper, the video generator does not produce perfect results in all cases. A notable instance of failure is observed with subjects that exhibit high dynamism, such as horses often depicted in motion. In such cases, the model occasionally produces unwarranted animations like walking or galloping, which disrupts the 3D reconstruction process. We show an example in Figure 8.

A.7. Conversion to Meshes

Although Gaussian Splatting is being rapidly adopted by the computer vision and graphics communities, some production applications require that objects be converted to meshes. We show that in these cases, it is straightforward to extract high-quality meshes from our Gaussian Splatting representation: one can run marching cubes (Cline, 1988) and then optionally continue to optimize the resulting mesh using DMTet (Shen et al., 2021). We show visual results of converting our models to meshes in Figure 9.