

4D-fy: Text-to-4D Generation Using Hybrid Score Distillation Sampling

Sherwin Bahmani^{1,2} Ivan Skorokhodov^{3,4} Victor Rong^{1,2} Gordon Wetzstein⁵ Leonidas Guibas⁵
Peter Wonka³ Sergey Tulyakov⁴ Jeong Joon Park⁶ Andrea Tagliasacchi^{1,7,8} David B. Lindell^{1,2}

¹University of Toronto ²Vector Institute ³KAUST ⁴Snap Inc. ⁵Stanford University ⁶University of Michigan ⁷SFU ⁸Google

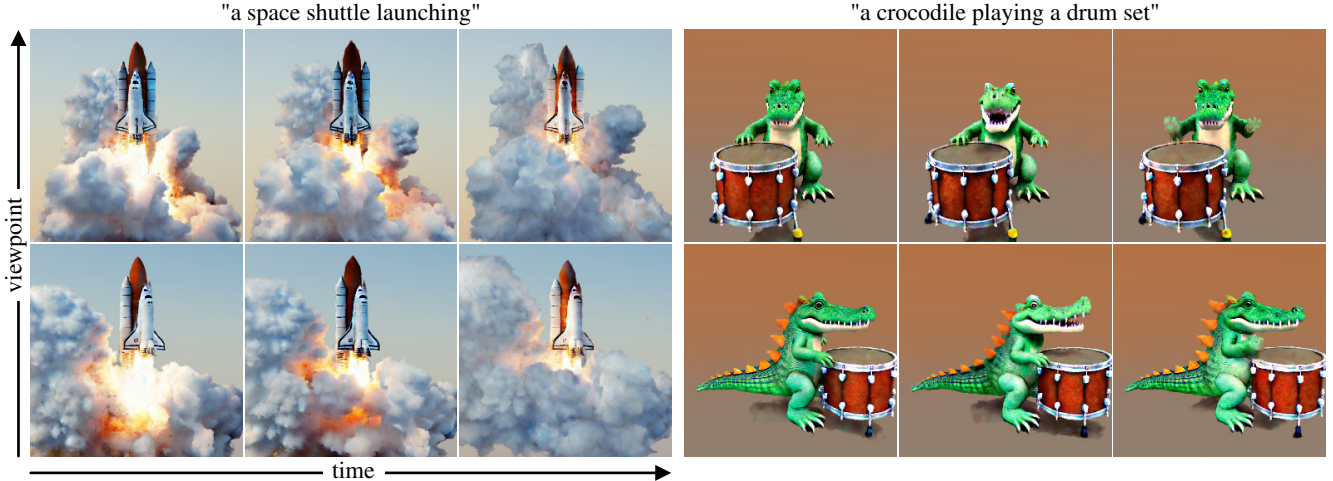


Figure 1. **Text-to-4D Synthesis.** We present 4D-fy, a technique that synthesizes 4D (i.e., dynamic 3D) scenes from a text prompt. We show scenes generated from two text prompts for different viewpoints (vertical dimension) at different time steps (horizontal dimension). Video results can be viewed on our website: <https://sherwinbahmani.github.io/4dfy>

Abstract

Recent breakthroughs in text-to-4D generation rely on pre-trained text-to-image and text-to-video models to generate dynamic 3D scenes. However, current text-to-4D methods face a three-way tradeoff between the quality of scene appearance, 3D structure, and motion. For example, text-to-image models and their 3D-aware variants are trained on internet-scale image datasets and can be used to produce scenes with realistic appearance and 3D structure—but no motion. Text-to-video models are trained on relatively smaller video datasets and can produce scenes with motion, but poorer appearance and 3D structure. While these models have complementary strengths, they also have opposing weaknesses, making it difficult to combine them in a way that alleviates this three-way tradeoff. Here, we introduce hybrid score distillation sampling, an alternating optimization procedure that blends supervision signals from multiple pre-trained diffusion models and incorporates benefits of each for high-fidelity text-to-4D generation. Using hybrid SDS, we demonstrate synthesis of 4D scenes with compelling appearance, 3D structure, and motion.

1. Introduction

The advent of internet-scale image–text datasets [52] and advances in diffusion models [20, 56, 58] have led to new capabilities in stable, high-fidelity image generation from text prompts [6, 49, 50]. Recent methods have also shown that large-scale text-to-image or text-to-video [54] diffusion models learn useful priors for 3D [25, 43] and 4D scene generation [55]. Our work focuses on text-to-4D scene generation (Fig. 1), which promises exciting new capabilities for applications in augmented and virtual reality, computer animation, and industrial design.

Current techniques for generating 3D or 4D scenes from text prompts typically iteratively optimize a representation of the scene using supervisory signals from a diffusion model [43, 65, 69]. Specifically, these methods render an image of a 3D scene, add noise to the rendered image, use a pre-trained diffusion model to denoise the rendered image, and estimate gradients used to update the 3D representation [43, 65]. This procedure, known as score distillation sampling (SDS) [43], underpins most recent methods for text-conditioned scene generation.

Using SDS for text-to-4D generation requires navigating

Table 1. Text-to-4D models face a tradeoff between the quality of appearance, 3D structure, and motion depending on the type of generative model used for score distillation sampling (SDS): text-to-image (T2I), 3D-aware T2I, or text-to-video (T2V).

<i>SDS model</i>	<i>appearance</i>	<i>3D structure</i>	<i>motion</i>
T2I [6, 49, 50, 76]	high	low	N/A
3D-aware T2I [30, 53]	medium	high	N/A
T2V [7, 21, 54, 67, 70]	low	low	high
Our method	medium	high	medium

a three-way tradeoff between the quality of appearance, 3D structure, and motion (see Table 1); existing techniques obtain satisfactory results in just one or two of these categories. For example, while SDS produces images that *appear* realistic when rendering a generated scene from any particular viewpoint, inspecting multiple viewpoints can reveal that the scene has several faces or heads, replicated appendages, or other incorrectly repeated 3D structures—an issue now referred to as the “Janus problem” [53].¹ One way to improve 3D structure is to use SDS with a 3D-aware diffusion model that is trained to generate images from different camera viewpoints [33]. But 3D-aware models sacrifice appearance quality as they require fine-tuning on synthetic datasets of posed images [53]. Incorporating motion into a scene using SDS with a text-to-video model [67] typically degrades the appearance relative to static scenes generated with text-to-image models, which are more realistic (see Fig. 2). While different types of diffusion models thus have complementary qualities, they also have opposing weaknesses (Table 1). Therefore, it is not trivial to combine them in a way that yields text-to-4D generation with high-quality appearance, 3D structure, and motion.

Here, we propose a method for text-to-4D scene generation that alleviates this three-way tradeoff using *hybrid SDS*, an alternating optimization scheme that blends gradient updates from multiple pre-trained diffusion models and synthesizes 4D scenes using the best qualities of each. The method consists of three stages of optimization: (1) we use a 3D-aware text-to-image model [53] to generate an initial static 3D scene (without the Janus problem); (2) we continue the optimization by blending in alternating supervision with variational SDS [69] and a text-to-image model to improve appearance; (3) we blend in alternating supervision using video SDS with a text-to-video model [67] to add motion to the scene. By smoothly incorporating supervisory signals from these three diffusion models throughout the training process, we achieve text-driven 4D scene generation with state-of-the-art quality in terms of appearance, 3D structure, and motion. Overall we provide the following contributions.

- We introduce hybrid SDS, a technique that extracts desir-

¹Referring to the two-faced Roman god of beginnings and endings.



Figure 2. **Comparing text-to-image and text-to-video models.** Rendered frames from Stable Diffusion version 2.1 (top; text-to-image) [1] and Zeroscope version 2 (bottom; text-to-video) [3] show significant disparity in appearance, with the text-to-image model appearing far more realistic.

able qualities from multiple pre-trained diffusion models and alleviates a tradeoff between appearance, 3D structure, and motion in text-to-4D scene generation.

- We provide a quantitative and qualitative evaluation of the method, and we explore the three-way tradeoff space with ablation studies to facilitate future research.
- We demonstrate text-to-4D generation based on open-source pretrained models and will make all codes and evaluation procedures publicly available.
- We present state-of-the-art results for the task of text-to-4D generation.

2. Related Work

Our method is related to techniques from multiple areas of generative modeling, including text-to-image, text-to-video, and text-to-3D models. For more extensive discussions of related works, we refer readers to a recent state-of-the-art report on diffusion models [42].

Text-to-image generation. Methods for generating images from text prompts are a relatively new innovation, first demonstrated using generative adversarial networks [48, 71, 77]. The problem itself is also related to other methods for text-based image retrieval [34] or image-conditioned text generation [59, 74]. More recently, models trained on text-image datasets with billions of samples [52] have become the state of the art for this task [49].

Diffusion models [20, 57] are a popular architecture for generative modeling on large-scale datasets, and autoregressive models have also shown promising results [46, 75]. Typically, these methods exploit a pretrained text encoder, such as CLIP [45], to encode the text prompt into a feature

vector used to condition the diffusion model [40, 47]. In diffusion models, high-resolution (i.e., megapixel) image generation is achieved by applying repeated upsampling layers [22, 47] or performing diffusion in the lower-resolution latent space of an autoencoder and then decoding the result to recover an image at the nominal resolution [16, 49]. Our work incorporates two open-source text-to-image diffusion models: Stable Diffusion [49] and MVDream [53] (a recent 3D-aware diffusion model) to enable 4D scene generation.

Text-to-video generation. Our work relies on the burgeoning field of video generation via diffusion models, an area that is somewhat constrained by the limited scale of video datasets. To counteract this, methods often utilize a hybrid training approach on both image and video datasets, such as WebVid-10M [5], HD-VG-130M [67], or HD-VILA-100M [72]. Recent approaches in this field typically employ variations of pixel-space upsampling (both in space and time) [21] or latent space upsampling to improve spatial and temporal resolution [17, 18, 68, 78]. Autoregressive models distinguish themselves by their ability to generate videos of varying lengths [63]. Further improvements in video synthesis have been achieved by finetuning pre-trained text-to-image diffusion models on video data [7, 54, 70], or separating the content and motion generation process by using an initial image frame as a starting point [17, 70]. Despite recent advances in text-to-video synthesis, the fidelity of generated videos still lags behind that of static image generation (see Fig. 2) and so they perform poorly when used directly with SDS for text-to-4D generation. Instead, our work leverages an open-source latent space text-to-video diffusion model called Zeroscope [3] (extended from the Modelscope architecture [66]) together with other pre-trained, open-source diffusion models using hybrid SDS.

Text-to-3D generation. Early methods for text-to-3D generation relied on parsers to convert input text to a semantic representation and synthesized scenes from an object database [4, 10, 12]. Later, automated, data-driven methods used multi-modal datasets [11], and pre-trained models, such as CLIP [45], to edit or stylize an input 3D mesh [14, 27] or a radiance field [64]. More recently, CLIP-based supervision enabled synthesis of entire 3D scenes [26, 51], and these techniques evolved into the most recent approaches, which optimize a mesh or radiance field based on SDS supervision [31, 43, 69]. The quality of their 3D structures has been improved by applying diffusion models that consider multiple viewpoints [32, 33, 53]. Alternatively, recent advancements have seen a shift towards using diffusion or transformer models to transform an input 2D image into a 3D representation for novel-view synthesis [9, 15, 35, 44, 60, 61, 73]. Still, these techniques do not yet support generating 4D scenes.

Our work is most closely related to Make-A-

Video3D (MAV3D) [55], a recent method for text-to-4D generation that integrates SDS-based supervision in two separate stages: first with a text-to-image model and subsequently with a text-to-video model. Similar to MAV3D, we aim to generate dynamic 3D scenes; however, our approach uses hybrid SDS, which allows gradient updates from multiple models to be smoothly blended together in an alternating optimization. Our approach generates high-quality dynamic 3D scenes and does not suffer from Janus problems.

3. Method

Our approach for text-to-4D generation builds upon a hash-encoding-based neural representation [39] that implicitly decomposes the scene into static and dynamic feature grids [62]. In this section we overview our representation for 4D neural rendering and describe the optimization procedure based on hybrid SDS (see Fig. 3).

3.1. 4D Neural Rendering

Volumetric neural rendering methods represent a scene using a neural representation to parameterize the attenuation and emission of light at every point in 3D space [36, 38]. We can use such a representation to render an image by casting a ray from the camera center of projection, through each pixel location, and into the scene. For sampled points along the ray $\mu \in \mathbb{R}^3$, we query a neural representation to retrieve a volumetric density $\tau \in \mathbb{R}_+$ and color $\mathbf{c} \in \mathbb{R}_+^3$, which describe attenuation and emission of light, respectively, at a particular point. Then, the resulting density and color samples are alpha-composited to recover the color of a rendered pixel \mathbf{C} as

$$\mathbf{C} = \sum_i w_i \mathbf{c}_i, \quad w_i = \alpha \prod_{j < i} (1 - \alpha_j), \quad (1)$$

where $\alpha_i = 1 - e^{-\tau_i \|\mu_i - \mu_{i+1}\|}$. We query the neural representation using an additional input time variable t , which enables modeling time-varying density and color.

We illustrate the neural representation in Fig. 3; it consists of two multi-resolution hash tables to disentangle static and dynamic scene modeling.

Following Müller et al. [39], the static hash table stores learnable feature vectors that are indexed by a voxel-lookup and hashing operation and decoded into density and color using two small multilayer perceptrons (MLPs). Concretely, we consider the neural representation

$$\mathcal{N}_\theta : \mu, t \rightarrow \tau, \mathbf{c} \quad (2)$$

with $\theta = \{\theta_{\text{static}}, \theta_{\text{dynamic}}, \theta_{\text{MLP}}\}$ denoting all learnable parameters from the static and dynamic hash tables and the MLPs. For a given μ , we query the static hash table by identifying the closest voxel at each scale $1 \leq s \leq S$. Then, we trilinearly

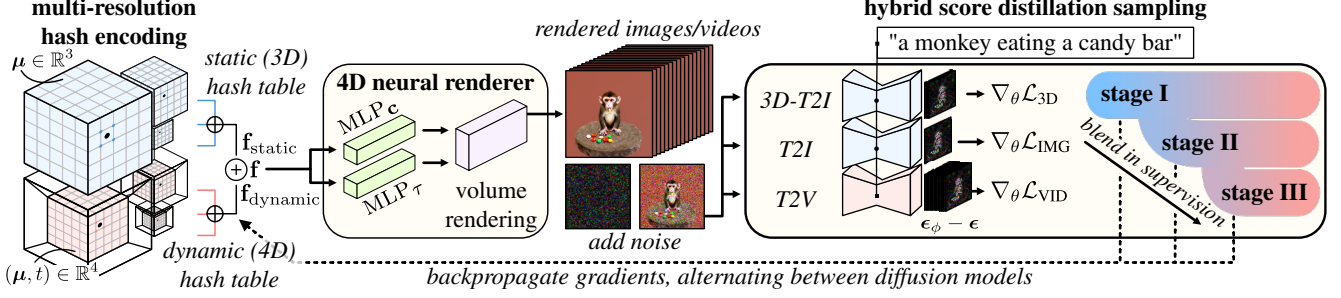


Figure 3. **Overview.** A 4D radiance field is parameterized using a neural representation with a static and dynamic multiscale hash table of features. Images and videos are rendered from the representation using volume rendering, and we supervise the representation using hybrid score distillation sampling—a technique that combines gradients from multiple types of pre-trained diffusion models. In the first stage of training we use gradients $\nabla_{\theta} \mathcal{L}_{3D}$ from a 3D-aware text-to-image model (3D-T2I) to iteratively optimize a representation without the Janus problem. Next, we blend in gradient supervision using variational SDS with a text-to-image model (T2I) to improve the appearance (i.e., we alternate supervision between $\nabla_{\theta} \mathcal{L}_{\text{IMG}}$ and $\nabla_{\theta} \mathcal{L}_{3D}$). In the last stage we incorporate gradients ($\nabla_{\theta} \mathcal{L}_{\text{VID}}$) from a text-to-video model (T2V) to add motion to the scene, and we update the scene using the other models in an alternating fashion.

interpolate the feature values from the voxel vertices after retrieving them from the hash table. Retrieved features from each scale are concatenated as $f_{\text{static}} = f_{\text{static}}^{(1)} \oplus \dots \oplus f_{\text{static}}^{(S)}$. We follow the same procedure to query the dynamic hash table given (μ, t) , except we use quadrilinear interpolation to interpolate feature values. The resulting features from the static and dynamic hash tables are added as $f = f_{\text{static}} + f_{\text{dynamic}}$. We do not model view-dependent effects in the feature encoding. Finally, we decode density and color as $\text{MLP}_{\tau}(f)$ and $\text{MLP}_c(f)$, respectively.

3.2. Hybrid Score Distillation Sampling

We leverage the 4D representation along with SDS to create dynamic 3D scenes from a text prompt. Our hybrid approach incorporates three different flavors of SDS that are smoothly merged during an alternating optimization procedure to improve the structure and quality of the 4D model:

1. SDS applied to a 3D-aware text-to-image diffusion model to optimize a static scene without the Janus problem.
2. Variational score distillation sampling (VSD; a modified version of SDS [69]) using a standard text-to-image model [49] to improve the appearance of the static scene.
3. Video SDS using a text-to-video model [67], which extends SDS to multiple video frames and adds motion to the scene.

In the following, we describe each type of SDS and how it is used for text-to-4D generation.

3D-aware scene optimization. We first consider optimizing a static scene using SDS with a 3D-aware text-to-image diffusion model [53]. The diffusion model is pre-trained using a stochastic forward process that slowly adds Gaussian noise to multiview images \mathbf{x} over timesteps $0 \leq t_d \leq T_d$. With increasing t_d , the process yields noisy images \mathbf{z}_{t_d} that, at $t_d = T_d$, are close to Gaussian. After training,

the model reverses this process to add structure to the noisy images. It predicts $\hat{\mathbf{x}}_{\phi}(\mathbf{z}_{t_d}; t_d, \mathbf{y}, \mathbf{T})$, which approximates the output of an optimal denoiser at each timestep t_d , conditioned on a text embedding \mathbf{y} [47, 49, 50] and the camera extrinsics \mathbf{T} corresponding to each image. In practice, text-to-image diffusion models typically predict the noise content ϵ_{ϕ} rather than the denoised image $\hat{\mathbf{x}}_{\phi}$. But note that the denoised image can still be obtained as $\hat{\mathbf{x}}_{\phi}(\mathbf{z}_{t_d}; t_d, \mathbf{y}, \mathbf{T}) \propto \mathbf{z}_{t_d} - \epsilon_{\phi}(\mathbf{z}_{t_d}; t_d, \mathbf{y}, \mathbf{T})$, i.e., by subtracting the predicted noise from the noisy image [20]. We implement 3D-aware SDS by rendering multiple images \mathbf{x}_{θ} from the neural representation, adding noise ϵ , and using the 3D-aware diffusion model [53] to predict the noise ϵ_{ϕ} using classifier-free guidance [19]. To update the parameters θ of the neural representation, we use the 3D-aware SDS gradient:

$$\nabla_{\theta} \mathcal{L}_{3D} = \mathbb{E}_{t_d, \epsilon, \mathbf{T}} \left[w(t_d) (\epsilon_{\phi}(\mathbf{z}_{t_d}; t_d, \mathbf{y}, \mathbf{T}) - \epsilon) \frac{\partial \mathbf{x}_{\theta}}{\partial \theta} \right], \quad (3)$$

where $w(t_d)$ is a weighting function that depends on the diffusion timestep, and we add a stop gradient to the output of the diffusion model [53]. Intuitively, the SDS loss queries the diffusion model to see how it adds structure to an image, then this information is used to backpropagate gradients to the scene representation.

Improving appearance using VSD. We incorporate an additional loss term based on VSD [69] to improve the appearance of images rendered from the scene. This term uses a pre-trained text-to-image model [49] along with a finetuning scheme that improves image quality over the 3D-aware text-to-image model alone. We follow Wang et al. [69] and augment the standard SDS gradient with the output of an additional text-to-image diffusion model that is finetuned using a low-rank adaptation [24], during scene optimization.

Specifically, we have

$$\nabla_{\theta} \mathcal{L}_{\text{IMG}} = \mathbb{E}_{t_d, \epsilon, \mathbf{T}} \left[w(t_d) \left(\epsilon_{\phi}(\mathbf{z}_{t_d}; t_d, \mathbf{y}) - \epsilon'_{\phi}(\mathbf{z}_{t_d}; t_d, \mathbf{y}, \mathbf{T}) \right) \frac{\partial \mathbf{x}_{\theta}}{\partial \theta} \right], \quad (4)$$

where ϵ'_{ϕ} is the noise predicted using a finetuned version of the diffusion model that incorporates additional conditioning from the camera extrinsics \mathbf{T} ; here, we let \mathbf{z}_{t_d} represent a noisy version of a single image rendered from \mathcal{N}_{θ} . The model is finetuned using the standard diffusion objective

$$\min_{\theta} \mathbb{E}_{t_d, \epsilon, \mathbf{T}} \left[\|\epsilon'_{\phi}(\mathbf{z}_{t_d}; t_d, \mathbf{y}, \mathbf{T}) - \epsilon\|_2^2 \right]. \quad (5)$$

Note that, different from the original description of VSD [69], we find we can omit the simultaneous optimization over multiple scene samples (i.e. the variational component of [69]), which reduces memory requirements without significantly degrading appearance.

Adding motion with Video SDS. Last, we use supervision from a text-to-video diffusion model [67] to add motion to the generated scene. This procedure extends the original SDS gradient by incorporating structure added by the diffusion model to all noisy video frames [55]. The video SDS gradient is given as

$$\nabla_{\theta} \mathcal{L}_{\text{VID}} = \mathbb{E}_{t_d, \epsilon} \left[w(t_d) \left(\epsilon_{\phi}(\mathbf{z}_{t_d}; t_d, \mathbf{y}) - \epsilon \right) \frac{\partial \mathbf{X}_{\theta}}{\partial \theta} \right]. \quad (6)$$

To simplify notation, we re-use ϵ_{ϕ} and ϵ to here represent the predicted and actual noise for each video frame, and we let \mathbf{X}_{θ} be a collection of V video frames $\mathbf{X}_{\theta} = [\mathbf{x}_{\theta}^{(1)}, \dots, \mathbf{x}_{\theta}^{(V)}]^T$ rendered from the representation.

Optimization procedure – Algorithm 1. We optimize the 4D representation in three stages that smoothly blend supervision in alternating steps from (1) 3D-aware SDS, (2) VSD, and (3) video SDS.

Stage 1. In the first stage of optimization, we update \mathcal{N}_{θ} using gradients from 3D-aware SDS until convergence. Since this stage focuses on optimizing a static scene, we freeze (i.e. not update) the parameters of the dynamic hash table $\mathbf{f}_{\text{dynamic}}$ and only update the static hash table and decoder MLP. We set the total number of first-stage iterations $N_{\text{stage-1}}$ to match that of Shi et al. [53], which allows the optimization to proceed until there are no distinguishable changes in the rendered scene from one iteration to the next.

Stage 2. Next, we add VSD gradients using an alternating optimization procedure. At each iteration, we randomly select to update the model using $\nabla_{\theta} \mathcal{L}_{3\text{D}}$ or $\nabla_{\theta} \mathcal{L}_{\text{IMG}}$ with probability $P_{3\text{D}}$ and P_{IMG} . We continue this alternating optimization for $N_{\text{stage-2}}$ iterations, until convergence. As we show in the next section, this stage of optimization results in improved appearance compared to using $\nabla_{\theta} \mathcal{L}_{3\text{D}}$ alone while also being free of the Janus problem.

Stage 3. Last, we update the representation using a combination of all gradient updates. Specifically, we randomly

Algorithm 1 Hybrid Score Distillation Sampling

Require:

- \mathcal{N}_{θ} ▷ 4D neural representation
- $N_{\text{stage-1}}, N_{\text{stage-2}}, N_{\text{stage-3}}$ ▷ iterations for each stage
- $P_{3\text{D}}, P_{\text{IMG}}$ ▷ update probabilities
- $\nabla_{\theta} \mathcal{L}_{3\text{D}}, \nabla_{\theta} \mathcal{L}_{\text{IMG}}, \nabla_{\theta} \mathcal{L}_{\text{VID}}$ ▷ SDS grads. (Eqs. 3, 4, 6)

```

1: // Stage 1
2: freeze dynamic hash map ( $\theta_{\text{dynamic}}$ )
3: for iter in  $N_{\text{stage-1}}$  do ▷ 3D update
4:   grad =  $\left\{ \nabla_{\theta} \mathcal{L}_{3\text{D}} \right.$ 
5:   UPDATE(grad)

6: // Stage 2
7: for iter in  $N_{\text{stage-2}}$  do ▷ 3D or IMG update
8:   grad =  $\begin{cases} \nabla_{\theta} \mathcal{L}_{3\text{D}}, & \text{with probability } P_{3\text{D}} \\ \nabla_{\theta} \mathcal{L}_{\text{IMG}}, & \text{otherwise} \end{cases}$ 
9:   UPDATE(grad)

10: // Stage 3
11: decrease learning rate of static hash map ( $\theta_{\text{static}}$ )
12: for iter in  $N_{\text{stage-3}}$  do ▷ 3D, IMG, or VID update
13:   grad =  $\begin{cases} \nabla_{\theta} \mathcal{L}_{3\text{D}}, & \text{with probability } P_{3\text{D}} \\ \nabla_{\theta} \mathcal{L}_{\text{IMG}}, & \text{with probability } P_{3\text{D}} \cdot P_{\text{IMG}} \\ \nabla_{\theta} \mathcal{L}_{\text{VID}}, & \text{otherwise} \end{cases}$ 
14:   if VID, unfreeze  $\theta_{\text{dynamic}}$ 
15:   UPDATE(grad)

16: procedure UPDATE(grad)
17:    $\mathbf{x} \leftarrow \mathcal{N}_{\theta}$  ▷ render images (Eq. 1)
18:   take gradient step on grad ▷ optimize  $\mathcal{N}_{\theta}$ 
19:   if IMG, take finetuning step (Eq. 5)
20: end procedure

```

select to update the model at each iteration using $\nabla_{\theta} \mathcal{L}_{3\text{D}}$, $\nabla_{\theta} \mathcal{L}_{\text{IMG}}$, or $\nabla_{\theta} \mathcal{L}_{\text{VID}}$ with probability $P_{3\text{D}}$, $P_{3\text{D}} \cdot P_{\text{IMG}}$, and $1 - P_{3\text{D}} \cdot P_{\text{IMG}}$, respectively. Since we now aim to incorporate motion into the representation, we unfreeze the parameters of the dynamic hash table during the update with $\nabla_{\theta} \mathcal{L}_{\text{VID}}$ but keep them frozen for updates using the text-to-image models. We also decrease the learning rate of the static hash table to preserve the high-quality appearance from the previous stage. We repeat the alternating optimization in the final stage until convergence, which we find occurs consistently within $N_{\text{stage-3}}$ iterations. Overall, hybrid SDS effectively combines the strengths of each pre-trained diffusion model while avoiding quality degradations that result from naively combining gradients from each model.

3.3. Implementation

We implement hybrid SDS based on the threestudio framework [2], which includes implementations of MV-Dream [53] (for 3D-aware text-to-image diffusion and SDS), ProlificDreamer [69] with Stable Diffusion [49] (text-to-

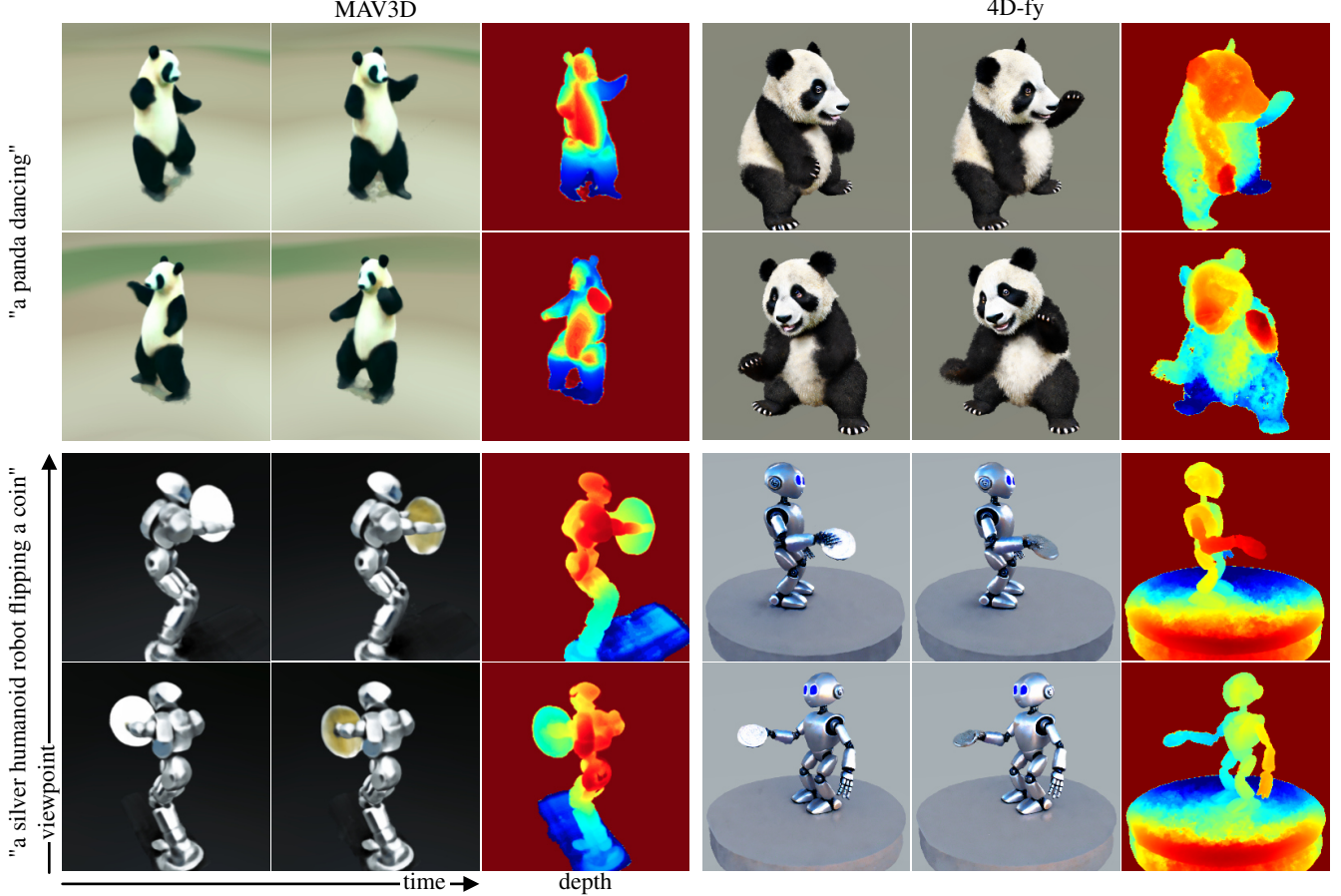


Figure 4. **Text-to-4D Comparison.** We compare against MAV3D [55], and observe our approach obtains significantly higher quality results.

image diffusion and VSD), and we implement the video SDS updates using Zeroscope [3, 67].

Hyperparameter values. We initialize the 4D neural representation following [31, 43] and add an offset to the density predicted by the network in the center of the scene to promote object-centric reconstructions. We set the learning rates for the static hash map to 0.01, for the dynamic hash map to 0.01, and for the MLP to 0.001. We drop the learning rate for the static hash map to 0.0001 before the last stage to focus the gradient updates on the dynamic hash map. The values of $N_{\text{stage-1}}$, $N_{\text{stage-2}}$, and $N_{\text{stage-3}}$ are set to 10000, 10000, and 100000, respectively. We set the probabilities for hybrid SDS to $P_{3D} = 0.5$ and $P_{\text{MG}} = 0.5$, which provides a reasonable tradeoff with respect to appearance, 3D structure, and motion.

Rendering. Each of the pretrained diffusion models has a different native resolution, so we render images from \mathcal{N}_θ accordingly. We render four images from different camera positions for the 3D-aware SDS update at the native (256×256 pixel) resolution of the 3D-aware text-to-image model. The VSD update is computed by rendering a 256×256 image

and bilinearly upsampling the image to the native resolution of Stable Diffusion (512×512). Finally, the video SDS update is computed by rendering 16 video frames at 160×288 resolution and upsampling to the native 320×576 resolution of Zeroscope. To render a background for the image, we optimize a second small MLP that takes in a ray direction and returns a background color. We composite the ray color C rendered from \mathcal{N}_θ on top of this background color.

Computation. We optimized the model on an NVIDIA A100 GPU. The entire procedure requires roughly 80 GB of VRAM and the three optimization stages require approximately 2, 2, and 19 hours of compute, respectively.

4. Experiments

4.1. Metrics

We assess our method using CLIP Score [41] and a user study. To promote future research in this field, we will open source the evaluation protocol for the user study. We compare our model against MAV3D for 17 prompts and against our ablations for a subset of 5 prompts. The prompts used for these

Table 2. **Quantitative results.** We compare our method against MAV3D and variations of 4D-fy with different loss terms or backbone architectures (i.e., with HexPlane [8]). The methods are evaluated in terms of CLIP Score (CLIP) and human preference based on appearance quality (AQ), 3D structure quality (SQ), motion quality (MQ), text alignment (TA), and overall preference (Overall). The numbers reported for human preference are the percentages of users who voted for the corresponding method over ours in head-to-head comparisons. We conduct evaluations against MAV3D using 17 text prompts and compare against our ablations using a subset of 5 text prompts. * indicates a statistically not significant preference for 4D-fy compared to the baseline or ablation ($p > 0.05$).

Method	CLIP	Human Preference				Overall
		AQ	SQ	MQ	TA	
MAV3D [55]	34.1	5%	8%	52%*	44%	28%
4D-fy	34.6		—			—
<i>Ablation Study</i>						
4D-fy	35.0		—			—
w/o $\nabla_{\theta}\mathcal{L}_{3D/IMG}$	29.3	0%	0%	22%	14%	6%
w/o $\nabla_{\theta}\mathcal{L}_{3D}$	35.1	12%	11%	5%	8%	9%
w/o $\nabla_{\theta}\mathcal{L}_{IMG}$	34.5	30%	32%	32%	31%	30%
w/o hybrid SDS	33.8	0%	0%	22%	12%	5%
w/ HexPlane	34.5	5%	8%	10%	8%	5%

comparisons are drawn from the 28 text prompts employed by MAV3D. Current text-to-4D models are costly to train, and many researchers in academia do not have access to the scale of resources available to large tech companies. Hence, we only used a subset due to computational limitations.

CLIP Score. CLIP Score [41] evaluates the correlation between a text prompt and an image. Specifically, this corresponds to the cosine similarity between textual CLIP [45] embedding and visual CLIP [45] embedding. The score is bound between 0 and 100, while 100 is best. We calculate the CLIP score for MAV3D using the same procedure we use for our method. Specifically, for each input text prompt, we render a video using the same camera trajectory as MAV3D, i.e., moving around the scene in azimuth with a fixed elevation angle. Subsequently, we score each video frame with CLIP ViT-B/32 and average the scores over all frames and text prompts to derive the final CLIP score.

User study. We conduct qualitative comparisons between our method and the baseline, MAV3D, by surveying 26 human evaluators. We use the same head-to-head comparison model as the user survey conducted by MAV3D. Specifically, we present text prompts alongside the corresponding outputs of our method and the baseline method in random order. Evaluators are requested to specify their overall preference for a video, as well as evaluate four specific properties: appearance quality, 3D structure quality, motion quality, and text alignment. In Table 2, we report the percentage of users who prefer each method overall and based on each of the

four properties. We conduct χ^2 -tests to evaluate statistical significance at the $p < 0.05$ level. Further details on the user study are included in the supplementary.

4.2. Results

We visualize spatio-temporal renderings along with depth maps in comparison to MAV3D in Fig. 4. Although both methods can synthesize 4D scenes, MAV3D noticeably lacks detail. In contrast, our method produces realistic renderings across space and time. We report quantitative metrics in Table 2. In terms of CLIP Score and overall preference in the user study 4D-fy outperforms MAV3D. Users indicated a statistically significant preference towards 4D-fy compared to MAV3D in terms of appearance quality, 3D structure quality, text alignment, and overall preference. They rated the motion quality roughly on par with MAV3D, which used a propriety text-to-video model. For example, overall, 72% of users prefer our method over 28% for MAV3D.

4.3. Ablations

We provide an in-depth analysis motivating our hybrid SDS training scheme by ablating each component and evaluating the use of a 4D neural representation more similar to that of MAV3D. We provide ablations in Table 2 and in Fig. 5.

Image guidance (w/o $\nabla_{\theta}\mathcal{L}_{3D/IMG}$). Technically, learning a dynamic 3D scene solely from a text-to-video model without text-to-image guidance is possible. To demonstrate the drawbacks of this approach, we present results where we skip the first two stages and directly train the model with text-to-video guidance only. This corresponds to setting $P_{3D} = 0$ and $P_{IMG} = 0$. Our experiments reveal that the text-to-video model fails to provide realistic 3D structure and high-quality appearance for generating a dynamic 3D scene.

3D-aware guidance (w/o $\nabla_{\theta}\mathcal{L}_{3D}$). We find that using a 3D-aware diffusion model is crucial for generating realistic 3D structures. If we remove the 3D-aware diffusion model, i.e., by setting $P_{3D} = 0$, we can generate scenes with similar motion and high-quality appearance, but the 3D structure is degraded. This is evident for both scenes in Fig. 5.

VSD guidance (w/o $\nabla_{\theta}\mathcal{L}_{IMG}$). We find that VSD helps provide a realistic scene appearance; if we disable it during scene generation, i.e., $P_{IMG} = 0$, there are some negative effects. For example in Fig. 5, the ice cream cone in the bucket (top row) is more detailed, and the dog’s face (bottom row) is sharper (please zoom in).

Hybrid SDS. To illustrate the impact of our hybrid SDS approach we disable image guidance after the second stage by setting $P_{3D} = 0$ and $P_{IMG} = 0$ for the third stage only. This aligns with the MAV3D training scheme, where a static model is pre-trained with text-to-image and subsequently fine-tuned with text-to-video. Our quantitative and qualita-

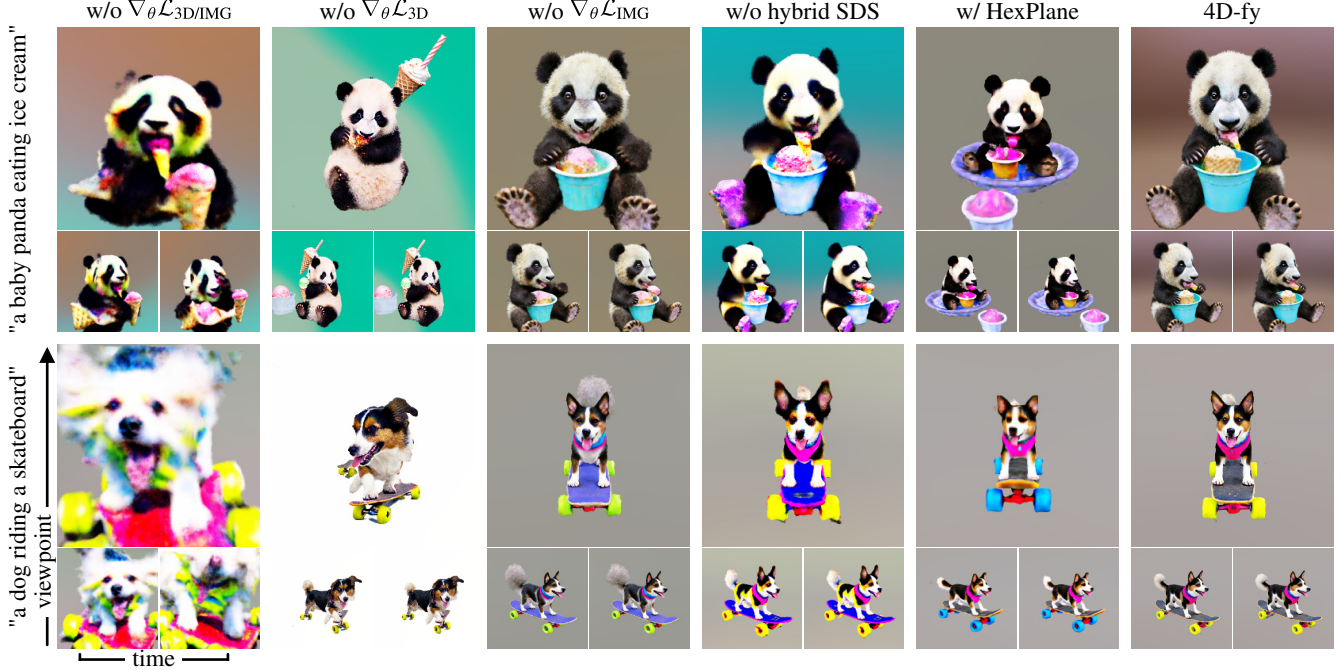


Figure 5. **Ablation study.** We assess the qualitative impact of removing gradient updates from different models during optimization. Our method without image guidance ($\nabla_{\theta} \mathcal{L}_{3D/IMG}$) does not produce realistic appearance and 3D structure. Removing the 3D-aware guidance ($\nabla_{\theta} \mathcal{L}_{3D}$) generates high-quality appearance but low-quality 3D structure. Our approach without VSD ($\nabla_{\theta} \mathcal{L}_{IMG}$) reduces the appearance quality. Hybrid SDS is crucial for appearance and 3D structure, while using HexPlane reduces the appearance quality. Best viewed digitally.

tive analysis shows that this approach results in degraded appearance and 3D structure. We find that incorporating text-to-image, 3D-aware text-to-image, and text-to-video via hybrid SDS in the final optimization stage preserves a realistic appearance and high-quality 3D structure.

Backbone architecture. Finally, we ablate the hash-grid-based 4D representation by replacing it with the HexPlane [8, 13] architecture. This representation similarly disentangles static and dynamic scene components and can be readily integrated into our pipeline. The HexPlane approach fails to match the appearance quality of the hash-grid-based representation. MAV3D uses HexPlane but implements a multi-scale variant with a large 5-layer decoding MLP featuring 128 hidden units. We could not re-implement this approach as the model does not fit on an 80 GB A100 GPU. To allow for a fair comparison, we instead increased the capacity of HexPlane to match the memory consumption of our hash-grid-based representation. We expect that increasing the capacity of HexPlane and longer training times could lead to similar results as our representation.

5. Conclusion

Our method synthesizes high-quality 4D scenes from text prompts using a novel hybrid score distillation sampling procedure. Our work alleviates a three-way tradeoff between

appearance, 3D structure, and motion and is the first to build on open-source models. We will release the code to facilitate future research in text-to-4D generation.

Limitations. Although our method produces compelling dynamic 3D scenes, there are several limitations and avenues for future work. First, the complexity of motion in our scenes is limited to simple movements. We believe that our method will directly benefit from future progress in text-to-video generation, as current text-to-video models suffer from low-quality renderings and unrealistic motion. Another way to improve motion could be exploiting recently proposed dynamic representations, e.g., dynamic 3D Gaussians [37]. Moreover, current metrics in text-to-3D generation are not sufficient, as they mainly rely on image-based metrics and user studies. Designing more sophisticated 3D and 4D metrics is an important direction for future work. Lastly, generating each scene takes a significant amount of time. Concurrent text-to-3D works [23, 28] alleviate this problem by training a large-scale model on 3D data, allowing generation within seconds. Incorporating our hybrid optimization procedure to blend between large-scale pre-training on 2D, 3D, and video data could enable fast text-to-4D generation.

Ethics Statement. We condemn the application of our method for creating realistic fake content intended to harm specific entities or propagate misinformation.

6. Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant program, the Digital Research Alliance of Canada, and by the Advanced Research Computing at Simon Fraser University. It was also supported in part by ARL grant W911NF-21-2-0104, a Vannevar Bush Faculty Fellowship, a gift from the Adobe Corporation, a PECASE by the ARO, NSF award 1839974, Stanford HAI, and a Samsung GRO.

References

- [1] Stable diffusion version 2. <https://github.com/Stability-AI/stablediffusion>. Accessed: 2023-10-31. **2**
- [2] Threestudio github page. <https://github.com/threestudio-project/threestudio>. Accessed: 2023-10-31. **5, 12**
- [3] Zeroscope text-to-video model. https://huggingface.co/cerspense/zeroscope_v2_576w. Accessed: 2023-10-31. **2, 3, 6**
- [4] Giovanni Adorni and Mauro Di Manzo. Natural language input for scene generation. In *Proc. EACL*, 1983. **3**
- [5] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proc. ICCV*, 2021. **3**
- [6] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. **1, 2**
- [7] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proc. CVPR*, 2023. **2, 3**
- [8] Ang Cao and Justin Johnson. HexPlane: a fast representation for dynamic scenes. In *Proc. CVPR*, 2023. **7, 8**
- [9] Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *Proc. ICCV*, 2023. **3**
- [10] Angel Chang, Manolis Savva, and Christopher D Manning. Learning spatial knowledge for text to 3d scene generation. In *Proc. EMNLP*, 2014. **3**
- [11] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Proc. ACCV*, 2018. **3**
- [12] Bob Coyne and Richard Sproat. Wordseye: An automatic text-to-scene conversion system. In *Proc. SIGGRAPH*, 2001. **3**
- [13] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proc. CVPR*, 2023. **8**
- [14] William Gao, Noam Aigerman, Thibault Groueix, Vova Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *Proc. SIGGRAPH*, 2023. **3**
- [15] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In *Proc. ICML*, 2023. **3**
- [16] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proc. CVPR*, 2022. **3**
- [17] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. **3**
- [18] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2022. **3**
- [19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *Proc. NeurIPS Workshop on Deep Generative Models*, 2021. **4**
- [20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. NeurIPS*, 2020. **1, 2, 4**
- [21] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. **2, 3**
- [22] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. **3**
- [23] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. **8**
- [24] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *Proc. ICLR*, 2021. **4**
- [25] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proc. CVPR*, 2022. **1**
- [26] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proc. CVPR*, 2022. **3**
- [27] Nikolay Jetchev. Clipmatrix: Text-controlled creation of 3d textured meshes. *arXiv preprint arXiv:2109.12922*, 2021. **3**
- [28] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023. **8**
- [29] Ruilong Li, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: A general nerf acceleration toolbox. *Proc. ICCV*, 2023. **12**

- [30] Weiyu Li, Rui Chen, Xuelin Chen, and Ping Tan. Sweet-dreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. *arXiv preprint arXiv:2310.02596*, 2023. [2](#)
- [31] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. [3](#), [6](#)
- [32] Yukang Lin, Haonan Han, Chaoqun Gong, Zunnan Xu, Yachao Zhang, and Xiu Li. Consistent123: One image to highly consistent 3d asset using case-aware diffusion priors. *arXiv preprint arXiv:2309.17261*, 2023. [3](#)
- [33] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proc. ICCV*, 2023. [2](#), [3](#)
- [34] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007. [2](#)
- [35] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. [3](#)
- [36] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *TOG*, 38(4):1–14, 2019. [3](#)
- [37] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023. [8](#)
- [38] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. [3](#)
- [39] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 41(4):102:1–102:15, 2022. [3](#), [12](#)
- [40] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proc. ICML*, 2022. [3](#)
- [41] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In *Proc. NeurIPS*, 2021. [6](#), [7](#)
- [42] Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T Barron, Amit H Bermano, Eric Ryan Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, et al. State of the art on diffusion models for visual computing. *arXiv preprint arXiv:2310.07204*, 2023. [2](#)
- [43] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3d using 2d diffusion. In *Proc. ICLR*, 2023. [1](#), [3](#), [6](#)
- [44] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. [3](#)
- [45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. ICML*, 2021. [2](#), [3](#), [7](#)
- [46] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proc. ICML*, 2021. [2](#)
- [47] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. [3](#), [4](#)
- [48] Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proc. ICML*, 2016. [2](#)
- [49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. CVPR*, 2022. [1](#), [2](#), [3](#), [4](#), [5](#)
- [50] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. 2022. [1](#), [2](#), [4](#)
- [51] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshian. Clip-forge: Towards zero-shot text-to-shape generation. In *Proc. CVPR*, 2022. [3](#)
- [52] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. 2022. [1](#), [2](#)
- [53] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. [2](#), [3](#), [4](#), [5](#), [12](#)
- [54] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. [1](#), [2](#), [3](#)
- [55] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. Text-to-4d dynamic scene generation. In *Proc. ICML*, 2023. [1](#), [3](#), [5](#), [6](#), [7](#), [12](#)
- [56] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. ICML*, 2015. [1](#)
- [57] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *Proc. ICLR*, 2021. [2](#)
- [58] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based

- generative modeling through stochastic differential equations. In *Proc. ICLR*, 2021. 1
- [59] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: A survey on deep learning-based image captioning. *PAMI*, 45(1):539–559, 2022. 2
- [60] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. *arXiv preprint arXiv:2303.14184*, 2023. 3
- [61] Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezhikov, Joshua B Tenenbaum, Frédo Durand, William T Freeman, and Vincent Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *arXiv preprint arXiv:2306.11719*, 2023. 3
- [62] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *Proc CVPR*, 2023. 3
- [63] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kilders, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022. 3
- [64] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-NeRF: Text-and-image driven manipulation of neural radiance fields. In *Proc. CVPR*, 2022. 3
- [65] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proc. CVPR*, 2023. 1
- [66] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 3
- [67] Wenjing Wang, Huan Yang, Zixi Tuo, Huiguo He, Junchen Zhu, Jianlong Fu, and Jiaying Liu. Videofactory: Swap attention in spatiotemporal diffusions for text-to-video generation. *arXiv preprint arXiv:2305.10874*, 2023. 2, 3, 4, 5, 6
- [68] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingen Zhou. Videocomposer: Compositional video synthesis with motion controllability. *arXiv preprint arXiv:2306.02018*, 2023. 3
- [69] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 1, 2, 3, 4, 5, 12
- [70] Ruiqi Wu, , Liangyu Chen, Tong Yang, Chunle Guo, Chongyi Li, and Xiangyu Zhang. Lamp: Learn a motion pattern for few-shot-based video generation. *arXiv preprint arXiv:2310.10769*, 2023. 2, 3
- [71] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proc. CVPR*, 2018. 2
- [72] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *Proc. CVPR*, 2022. 3
- [73] Paul Yoo, Jiaxian Guo, Yutaka Matsuo, and Shixiang Shane Gu. Dreamspare: Escaping from plato’s cave with 2d diffusion model given sparse views. *arXiv preprint arXiv:2306.03414*, 2023. 3
- [74] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proc. CVPR*, 2016. 2
- [75] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 2
- [76] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023. 2
- [77] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proc. ICCV*, 2017. 2
- [78] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022. 3

S1. Implementation Details

Here, we provide additional implementation details.

Optimization Following [53, 69], during the first stage of the optimization, we anneal the sampled diffusion timesteps over 5000 iterations from $t_d \in [0.02, 0.98]$ to a range of $t_d \in [0.02, 0.5]$. In subsequent stages, we keep the timestep sampling the same, except for the video SDS updates, where we sample $t_d \in [0.02, 0.98]$. For the $\nabla_{\theta} \mathcal{L}_{\text{VID}}$ updates, we set the total learning rate to 0.1, while $\nabla_{\theta} \mathcal{L}_{\text{3D}}$ and $\nabla_{\theta} \mathcal{L}_{\text{IMG}}$ use a learning rate of 1.0.

Model Our hash-grid-based representation [39] uses default values [2] for both the static and dynamic components: 16 levels, 2 features per level, and a base resolution of 16.

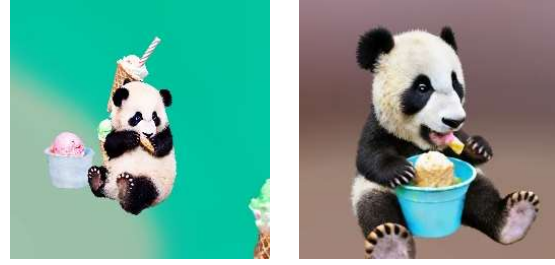
Rendering We use NerfAcc [29] as our rendering pipeline without any changes to the default values in threestudio [2]. During training and inference we use 512 samples per ray. Following the MAV3D [55] evaluation protocol, we sample 64 frames during inference after training our model with 16 frames, which is possible due to the implicit time coordinate. During training, we sample a random time offset with evenly spaced time coordinates over the $[0.0, 1.0]$ time interval. This enables smooth time sampling across the whole range of time coordinates.

S2. User Study

The user study was carried out as a single survey consisting of 42 questions, each with 5 subquestions. Each question asked the evaluator to compare two videos: one showing a scene rendered with 4D-fy and another rendered with a separate method using the same text prompt as input. Evaluators filled out five subquestions that asked their preferred video based on appearance quality, 3D structure quality, motion quality, text alignment, and overall preference, as shown in Fig. S6. Evaluators were given the following instructions for each metric.

- **Appearance Quality:** Evaluate the clarity and visual appeal of the scene as it appears from any particular viewpoint (ignoring, e.g., inconsistencies in appearance across different viewpoints). Your assessment should focus on the appearance of the foreground object and ignore the background of the video.
- **3D Structure Quality:** Assess the detail and realism of the shape of the scene across the multiple viewpoints shown in the video.
- **Motion Quality:** Assess the realism of motion, including the amount of motion and how naturally the movements in the video are portrayed.
- **Text Alignment:** Determine how accurately each video reflects the content of the text prompt. Consider whether the key elements of the prompt are represented.

Question 1 of 42



Prompt: Baby panda eating ice cream

Choose the video which you prefer according to each of the following criteria:

Appearance Quality:	<input type="radio"/> Left	<input type="radio"/> Right
3D Structure Quality:	<input type="radio"/> Left	<input type="radio"/> Right
Motion Quality:	<input type="radio"/> Left	<input type="radio"/> Right
Text Alignment:	<input type="radio"/> Left	<input type="radio"/> Right
<hr/>		
Overall Preference:	<input type="radio"/> Left	<input type="radio"/> Right

Figure S6. **Example survey question.** A survey question is shown above. The two videos are rendered using a baseline method (left) and 4D-fy (right). The left–right ordering of videos generated using each method is randomized throughout the survey.

- **Overall Preference:** State your overall preference between the two videos. This is your subjective appraisal of which video, in your view, stands out as better based on appearance quality, 3D structure quality, motion quality, and text alignment, (i.e., overall quality).

Of the 42 questions, 17 were comparisons between 4D-fy and MAV3D. The other 25 questions were between 4D-fy and each of the five ablated methods described in Sec. 4.3 of the main paper, with five questions for each. For each comparison and metric, the results were tested against the null hypothesis that evaluators had no preference between either method; i.e., they would choose either with probability 0.5. We aggregate over prompts and evaluators and use χ^2 analysis to determine the corresponding p -value. We choose $p < 0.05$ as a significant deviation from the null hypothesis and find that—with the exception of comparing motion quality between 4D-fy and MAV3D—all p -values are well below 0.05. The p -value for text alignment and motion quality between 4D-fy and MAV3D are 0.0174 and 0.3919 respectively, while the rest are < 0.0001 . All statistically significant results indicated that users preferred videos rendered using 4D-fy over MAV3D.

S3. Qualitative Results

In Fig. S7, we present additional results generated using our method. We highly encourage readers to view the videos

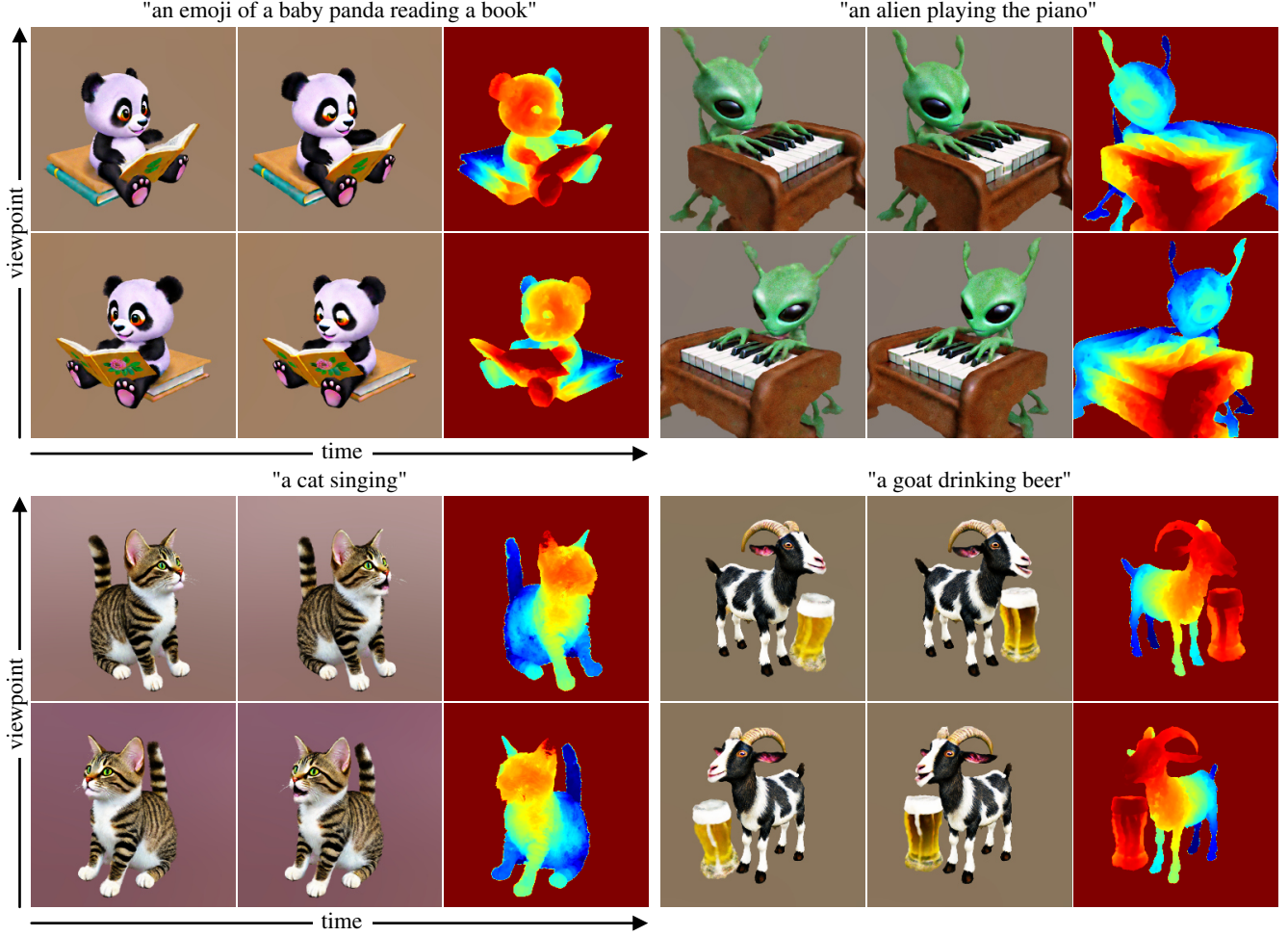


Figure S7. **Text-to-4D Synthesis.** We present additional results using 4D-fy.

included in our supplementary website to gain a better appreciation of our text-to-4D generation results.

S4. Limitations

In addition to limitations outlined in the main paper, we briefly discuss the temporal flickering in renderings generated with 4D-fy. The main goal of our work is to generate high-quality dynamic 3D scenes—and especially to improve image quality compared to previous work where results can appear overly smooth or cartoon-like (e.g., Fig. 4). It may be possible to mitigate flickering artifacts by placing more emphasis on supervision with the video diffusion model; however, this may trade off image quality. Reducing temporal flickering without any penalty to image quality (e.g., avoiding blurry or overly smooth images) is an important direction for future work.

S5. Ethics Statement

In its current form, our method is not able to edit real people. However, it could be extended and misused for generating edited imagery of real people. We condemn the application of our method for creating realistic fake content intended to harm specific entities or propagate misinformation.