4Diffusion: Multi-view Video Diffusion Model for 4D Generation

Haiyu Zhang[1,2,*] Xinyuan Chen[2], Yaohui Wang[2], Xihui Liu[3], Yunhong Wang[1], Yu Qiao[2,†]

[1]Beihang University   [2]Shanghai AI Laboratory   [3]The University of Hong Kong

[1]{zhyzhy,yhwang}@buaa.edu.cn   [2]{chenxinyuan,wangyaohui,qiaoyu}@pjlab.org.cn

[3]xihuiliu@eee.hku.hk

https://aejion.github.io/4diffusion
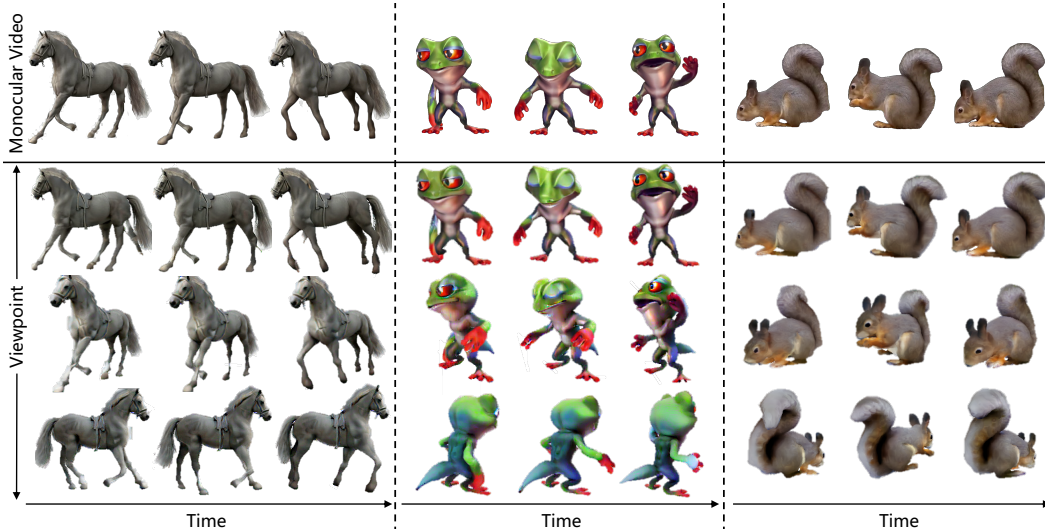
Figure 1: **4Diffusion** generates spatial-temporally consistent 4D contents from monocular videos.

Abstract

Current 4D generation methods have achieved noteworthy efficacy with the aid of advanced diffusion generative models. However, these methods lack multi-view spatial-temporal modeling and encounter challenges in integrating diverse prior knowledge from multiple diffusion models, resulting in inconsistent temporal appearance and flickers. In this paper, we propose a novel 4D generation pipeline, namely **4Diffusion**, aimed at generating spatial-temporally consistent 4D content from a monocular video. We first design a unified diffusion model tailored for multi-view video generation by incorporating a learnable motion module into a frozen 3D-aware diffusion model to capture multi-view spatial-temporal correlations. After training on a curated dataset, our diffusion model acquires reasonable temporal consistency and inherently preserves the generalizability and spatial consistency of the 3D-aware diffusion model. Subsequently, we propose 4D-aware Score Distillation Sampling loss, which is based on our multi-view video diffusion model, to optimize 4D representation parameterized by dynamic NeRF. This aims to eliminate discrepancies arising from multiple diffusion models, allowing for generating spatial-temporally consistent 4D content. Moreover, we devise an anchor loss to enhance the appearance details and facilitate the learning of dynamic

*Work done when Haiyu Zhang interned at Shanghai AI Laboratory.

†Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

NeRF. Extensive qualitative and quantitative experiments demonstrate that our method achieves superior performance compared to previous methods.

1 Introduction

In recent years, diffusion models have significantly impacted the era of image, video, and 3D generation. With the support of large-scale text-to-image diffusion models [43, 1] and 3D-aware diffusion models [44, 29, 52], many works [26, 39, 51, 9, 34, 41, 50, 30, 55, 27] leverage Score Distillation Sampling (SDS) [38] to distill the prior knowledge from diffusion models to optimize a 3D shape parameterized by NeRF [35] or 3DGS [21]. Although they have attained faithful results, they only focus on creating static 3D shapes, neglecting the dynamics of objects in the real world.

Generating 4D content, i.e., dynamic 3D content, holds diverse applications in the virtual realm, including digital human, gaming, media, and AR/VR. The main challenge lies in creating 4D content with vivid motion and high-quality spatial-temporal consistency. The pioneering study MAV3D [47] introduces a two-stage method, which first learns a static 3D shape with a text-to-image diffusion model and then deforms the static 3D shape with a text-to-video diffusion model [46]. However, MAV3D encounters the Janus problem and generates 4D contents with poor appearance and motion [4]. To overcome these issues, the following works [4, 66, 65, 28] employ multiple diffusion models for distinct purposes. Specifically, these methods leverage 3D-aware diffusion models [44, 29] and text-to-image diffusion models [43] to achieve spatial consistency and visually appealing appearance. Akin to MAV3D, they utilize video diffusion models [2, 54, 46] to add motion to create 4D content.

The aforementioned methods utilize multiple diffusion models for 4D generation. As Fig. 2 illustrates, when diffusing images rendered from a 3D model, the 3D-aware diffusion model [44] generates multi-view images to address the spatial ambiguity. On the other hand, the 2D image diffusion model [43] produces a clean image with subtle details to refine appearance. The 2D video diffusion model [2] generates dynamic frames to ensure temporal consistency within the same viewpoint. However, there is no accurate guidance to ensure multi-view spatial-temporal consistency due to the lack of multi-view spatial-temporal modeling. Moreover, it is challenging to integrate diverse prior knowledge from multiple diffusion models, often leading to inconsistent temporal appearance and flickers as shown in the second row of Fig. 7.
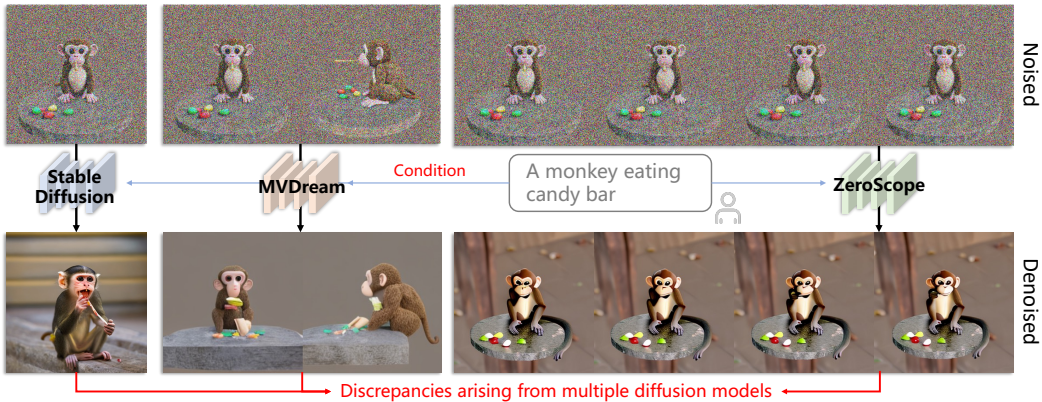
Figure 2: **Challenges.** The denoised images from Stable Diffusion (SD) [43], MVDream [44], and ZeroScope [2]. These diffusion models can not provide multi-view spatial-temporal guidance and exhibit discrepancies, making their integration challenging.

In this paper, we present a novel 4D generation pipeline, namely **4Diffusion**, to create high-quality spatial-temporally consistent 4D content from a monocular video. Specifically, we propose a unified diffusion model, 4DM, to capture multi-view spatial-temporal correlations for multi-view video generation. To achieve this, we construct 4DM based on the powerful pre-trained 3D-aware diffusion model [52], which already ensures high-quality multi-view spatial consistency. We then seamlessly integrate a motion module into the 3D-aware diffusion model to extend the temporal modeling capability. Contrary to previous attempts [16, 14] that typically demand extensive large-scale video datasets for tuning the motion module, 4DM achieves reasonable temporal consistency and

2

captures multi-view spatial-temporal correlations after training on only hundreds of multi-view videos. Importantly, we keep the parameters of the 3D-aware diffusion model unchanged to preserve the generalization ability and spatial consistency. 4DM provides multi-view spatial-temporal guidance for 4D generation. Therefore, we propose 4D-aware SDS loss to distill prior knowledge from 4DM to optimize 4D content parameterized by dynamic NeRF. This approach eliminates discrepancies arising from multiple diffusion models and stabilizes the optimizing process. Moreover, we use 4DM to generate anchor videos conditioned on the input monocular video and devise an anchor loss to enhance the appearance details, facilitating the learning of dynamic NeRF. Finally, we generate 4D content with high-quality spatial-temporal consistency and vibrant motion coherence with the input video as shown in Fig. 1. Qualitative and quantitative experiments demonstrate that our method achieves state-of-the-art performance on multi-view video generation and 4D generation from monocular videos.

To summarize, our contributions are as follows: **1)** We present **4Diffusion**, a novel 4D generation pipeline that generates high-quality spatial-temporal consistent 4D content from a monocular video with a multi-view video diffusion model. **2)** We propose a multi-view video diffusion model, 4DM, which provides multi-view spatial-temporal guidance for 4D generation. It trains on only hundreds of curated high-quality multi-view videos to capture multi-view spatial-temporal correlations. **3)** We combine 4D-aware SDS loss and an anchor loss based on 4DM to optimize dynamic NeRF, which stabilizes the training process and allows for generating high-quality 4D content.

2   Related Work

Recent breakthroughs in multiple research domains have significantly accelerated progress in 4D generation task. Here, we discuss the most relevant fields, including 3D generation, video and 3D-aware diffusion models, and 4D generation.

3D Generation. Recent studies in 3D generation can be classified into three categories: 3D generative methods [53, 17, 45, 36, 60, 3, 58, 12], feed forward methods [18, 49, 23, 67], and diffusion prior-based methods [26, 39, 51, 9, 34, 41, 50, 30, 55, 27]. Inspired by the advancements in 2D content creation, 3D generative methods utilize the robust diffusion [53] or flow-based [60] backbone to generate 3D data represented by Signed Distance Function (SDF) [60], voxel grid [36], triplane [8, 17, 45], or weights of neural network [12]. However, these methods require time-consuming pre-training to fit each 3D data and are limited to creating a single category. Feed forward methods [18, 23] adopt image features extracted from the pre-trained visual encoder DINO [7] to reconstruct 3D representations through a highly scalable and efficient transformer-based decoder. Although they can produce a 3D shape in a few seconds, they demand extensive training on large-scale 3D datasets, which is impractical with limited 4D datasets for 4D generation. Furthermore, diffusion prior-based methods distill prior knowledge from diffusion generative models via SDS [38] to optimize 3D representations, enabling the generation of high-quality 3D shapes with strong generalizability. In contrast to static 3D generation, our method focuses on creating 4D content.

Video and 3D-aware Diffusion Models. With the success of large-scale text-to-image diffusion models [43, 1], recent works attempt to use diffusion models to generate more complex signals, including video and 3D. AnimateDiff [16] inserts a learnable motion module into the frozen text-to-image model for video generation, which preserves the efficacy of the text-to-image model while successfully modeling temporal information. Recent 3D-aware diffusion model Zero-1-to-3 [29] adopts a stable diffusion model conditioned on relative camera pose and a single image for novel view synthesis. However, this method still suffers from the Janus problem and content drafting problem [44] due to the lack of explicit 3D modeling. Approaches like [31, 44, 59, 32, 52] leverage 3D-aware attention block to model the joint probability distribution of multi-view images, leading to spatially consistent generation. However, these approaches are incapable of producing multi-view consistent videos, due to the absence of temporal or spatial modeling.

4D Generation. Recently, several works have delved into 4D generation from various user-friendly prompts, such as text [47, 4, 66, 28], a single image [65, 66], and a monocular video [42, 20, 62]. The pioneering study MAV3D [47] proposes a two-stage method to optimize 4D representation, i.e., Hexplane [6], with both text-to-image and text-to-video diffusion models in a static-to-dynamic manner. To generate 4D contents with realistic appearance, Dream-in-4D [66] and 4D-fy [4] combine hybrid diffusion models. Specifically, they utilize 3D-aware and 2D diffusion guidance to learn a

3
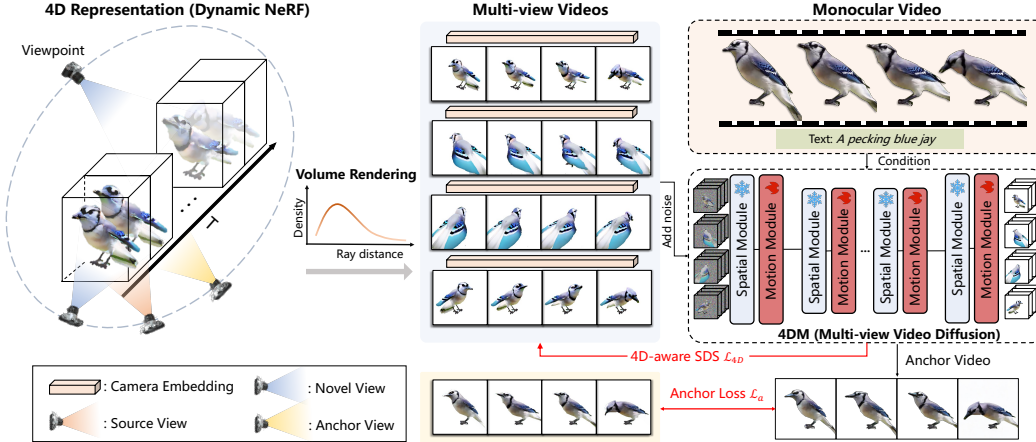
Figure 3: **4Diffusion overview.** Our method first trains a unified diffusion, named 4DM, by inserting a learnable motion module at the end of each frozen spatial module of ImageDream to capture multi-view spatial-temporal correlations. Given a monocular video and text prompt, 4DM can produce consistent multi-view videos. Then, we combine 4D-aware SDS and an anchor loss based on 4DM to optimize 4D content parameterized by Dynamic NeRF.

static 3D representation and incorporate video diffusion guidance to add motion. However, these diffusion models can not offer multi-view spatial-temporally consistent guidance and it is difficult to integrate diverse prior knowledge from multiple diffusion models, resulting in suboptimal results. In contrast to these approaches, we design a unified model to capture multi-view spatial-temporal correlations for 4D generation.

Similar to us, [20, 42, 63, 62, 57] generate 4D content from a monocular video. Consistent4D [20] introduces an interpolation-driven loss between two adjacent frames to enhance spatial-temporal consistency. However, Consistent4D lacks temporal modeling cross frames. DreamGaussian4D [42], 4DGen [62], and SC4D [57] combine 4D Gaussian Splatting [56, 19] into 4D generation pipeline. Although they notably reduce optimization time, they may result in blurred appearance and inaccurate geometry due to the explicit characteristics of Gaussians. STAG4D [63] proposes a training-free strategy to generate sparse anchor multi-view videos for 4D generation. In contrast, we propose a multi-view video diffusion model to provide multi-view spatial-temporal consistency guidance for 4D generation.

3 Method

Given a monocular video $V = \{I_j | j = 1, 2, ..., T\}$ with T frames and an optional textual caption, our goal is to generate a high-quality spatial-temporally consistent 4D content, capable of rendering from any novel viewpoint across the temporal dimension. In Sec. 3.1, we talk about 3D-aware diffusion models, employed as the initialization of our unified diffusion model. In Sec. 3.2, we propose a unified diffusion model 4DM to capture multi-view spatial-temporal correlations for multi-view video generation. Subsequently, we elaborate on distilling prior knowledge from 4DM to optimize 4D content parameterized by dynamic NeRF and devise an anchor loss to enhance the appearance details, as detailed in Sec. 3.3. Fig. 3 shows the overall pipeline of our method.

3.1 Preliminary: 3D-aware Diffusion Models

3D-aware diffusion models learn spatial relationships from multi-view images for 3D generation and can serve as an initialization of our unified diffusion model. Recent works [31, 59, 32] mainly focus on generating multi-view images from predetermined sparse viewpoints and necessitate additional algorithms for 3D reconstruction. Although we can extend these methods to generate multi-view videos and employ 4D reconstruction algorithms, it is challenging to reconstruct high-quality 4D content from a limited number of viewpoints. Therefore, we design our unified diffusion model to generate multi-view videos from arbitrary viewpoints and choose ImageDream [52] as initialization.

4

Given four arbitrary orthogonal viewpoints under canonical coordination and a single image with an optional textual caption, ImageDream can synthesize four multi-view images that align coherently with the input. Specifically, ImageDream utilizes an adapter similar to IP-Adapter [61] to inject image prompts and a 3D self-attention module to capture spatial relationships.

3.2 4DM: Multi-view Video Diffusion Model

To maintain the spatial consistency and mitigate training complexity, we design our multi-view video diffusion model 4DM based on a pre-trained 3D-aware diffusion model (i.e., ImageDream [52]). Given a monocular video with an optional text prompt and four orthogonal novel viewpoints under canonical coordination, 4DM aims to generate four spatial-temporally consistent videos.

Although we can directly use the original ImageDream to generate a set of individual multi-view images to form multi-view videos, the result lacks temporal consistency as ImageDream has no layer for temporal modeling, as shown in Fig. 8. We thus add a zero-initialized motion module at the end of each block of the UViT network of ImageDream. Specifically, each motion module begins with group normalization and a linear projection, followed by two self-attention blocks and one feed-forward block. A final linear projection is then applied, after which the residual hidden feature is added back at the end of each motion module as detailed in Fig. 4. Then, each attention block i of 4DM includes a spatial module and a motion module l_m^i. The spatial module comprises a 3D self-attention module l_s^i and a cross-attention module. We first concatenate the monocular video latent and four multi-view video latents encoded by VAE [22] to obtain a batch B of latents $\mathbf{Z} \in \mathbb{R}^{B \times F \times N \times C \times H \times W}$, where C is the number of channels, H and W are spatial resolutions, $N = 5$ is the number of viewpoints, and F is the number of frames. Subsequently, we reshape the temporal axis into the batch dimension and independently process multi-view video latents through the 3D self-attention module,

$$\mathbf{Z_s} \longleftarrow \mathbf{Reshape}(\mathbf{Z}, B\,F\,N\,C\,H\,W \to (B\,F)\,N\,H\,W\,C), \tag{1}$$

$$\mathbf{Z_s} \longleftarrow l_s^i(\mathbf{Z_s}), \tag{2}$$

$$\mathbf{Z_s} \longleftarrow \mathbf{Reshape}(\mathbf{Z_s}, (B\,F)\,N\,H\,W\,C \to B\,F\,N\,C\,H\,W). \tag{3}$$

Then, we use the adapter in ImageDream to individually process the input video frames and output the video features to perform cross-attention operations. Here, we also reshape the temporal axis into the batch dimension to prevent dimensional confusion. Furthermore, for the motion module, we perform self-attention exclusively along the temporal axis by reshaping the spatial dimensions and the viewpoint dimension into the batch dimension,

$$\mathbf{Z}' \longleftarrow \mathbf{Reshape}(\mathbf{Z}, B\,F\,N\,C\,H\,W \to (B\,N\,H\,W)\,F\,C), \tag{4}$$

$$\mathbf{Z}' \longleftarrow l_m^i(\mathbf{Z}'), \tag{5}$$

$$\mathbf{Z}' \longleftarrow \mathbf{Reshape}(\mathbf{Z}', (B\,N\,H\,W)\,F\,C \to B\,F\,N\,C\,H\,W). \tag{6}$$

We utilize Objaverse dataset [11] to train 4DM. Although Objaverse provides nearly 44K animated 3D shapes, rendering multi-view videos and training a diffusion model are time- and computation-consuming using the entire dataset. Moreover, it is worth noting that the Objaverse dataset contains a significant amount of flawed data. Consequently, we manually select a curated subset of 926 high-quality animated 3D shapes from Objaverse dataset [11]. We render multi-view videos from those animated 3D shapes to tune our motion module while holding the parameters of the origin ImageDream frozen. Surprisingly, 4DM successfully learns reasonable temporal dynamics and preserves the characteristics of the origin ImageDream model, including generalization ability, spatial consistency, and image understanding ability, even when trained on a small curated dataset. As Fig. 8 illustrates, 4DM generates multi-view spatial-temporal consistent videos, surpassing the performance of ImageDream. For more details on our dataset, please refer to supplementary material.

Training Objectives. For each animated 3D shape from our dataset, we render a monocular video V_m with a random viewpoint and four videos V_o with orthogonal viewpoints $\mathbf{c}_{mv}^v$ and select $F = 8$ frames from each video at a stride of 4 to create our multi-view video dataset $\mathcal{X}_{mv}^v = \{\mathbf{x}_{mv}^v, y, \mathbf{x}_r^v, \mathbf{c}_{mv}^v\}$. Here, $\mathbf{x}_r^v$ and $\mathbf{x}_{mv}^v$ represent the video clips from V_m and V_o. y is the text prompt captioned by Cap3D [33]. Then, we use $\mathcal{X}_{mv}^v$ following the diffusion loss to train 4DM,

$$\mathcal{L}_{MV}(\theta, \mathcal{X}_{mv}) = \mathbb{E}_{\mathbf{x}, y, \mathbf{x}_r, \mathbf{c}, t, \epsilon} \left[\|\epsilon - \epsilon_\theta(\mathbf{x}^p; y, \mathbf{x}_r^p, \mathbf{c}^p, t)\|_2^2 \right],$$

$$\text{where, } (\mathbf{x}^p, \mathbf{x}_r^p, \mathbf{c}^p) = \begin{cases} (\mathbf{x}_{mv}, \mathbf{0}, \mathbf{0}), & \text{with probability } p \\ (\mathbf{x}_{mv}, \mathbf{x}_r^v, \mathbf{c}_{mv}^v), & \text{with probability } 1 - p \end{cases} \tag{7}$$

5
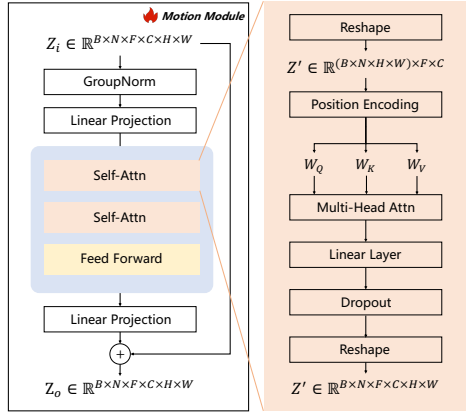
Figure 4: The detailed overview of the architecture of motion module.
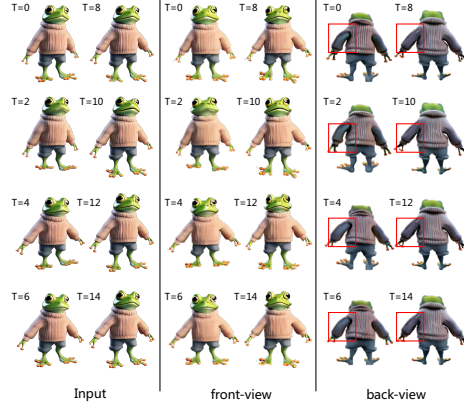
Figure 5: The illustration of multi-view video generation when input video exceeds 8 frames.
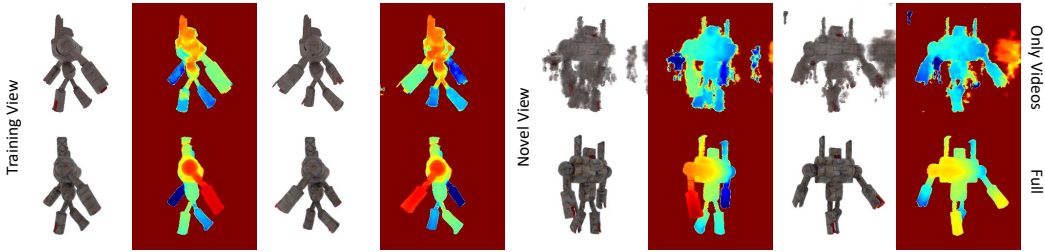
Figure 6: Illustration of directly optimizing on the generated multi-view videos.

here, $\mathbf{x}_{mv}$ represent the noisy video latents derived from $\mathbf{x}_{mv}^v$. These latents are initially encoded by VAE and subsequently noised by random noise ϵ at a diffusion timestep t. For more details about the noising process, please refer to [43]. ϵ_θ is 4DM model parametrized by θ. During the training of 4DM, we ensure that our training data does not overlap with the test data used in our experiments.

3.3 4D Generation

Dynamic NeRF Representation. Recent methods [25, 13] use neural networks or explicit spatial grids to map a 6D spatial-temporal coordinate $(x, \mathrm{d}, \mathbf{t})$ to density $\tau(x, \mathbf{t}) \in \mathbb{R}_+$ and view-dependent color $c(x, \mathrm{d}, \mathbf{t}) \in \mathbb{R}_+^3$ of dynamic scenes, where $x = o + \ell\mathrm{d}$ ($\ell > 0$) are sampled points along a ray originating at o with direction d and $\mathbf{t}$ denotes timestamp. Then, they leverage volumetric rendering to render images,

$$C = \sum_i \omega_i c_i, \text{ where } \omega_i = e^{-\sum_{j<i} \tau_j (\ell_{j+1} - \ell_j)} \big(1 - e^{\tau_i(\ell_{i+1} - \ell_i)}\big). \tag{8}$$

Following 4D-fy [13] and iNGP [37], we use one multi-resolution spatial grid $\mathbf{P}_{xyz}$ and one spatial-time planes $\mathbf{P}_{xyzt}$ as 4D representation. Here, both $\mathbf{P}_{xyz}$ and $\mathbf{P}_{xyzt}$ use hash tables to store learnable features. Then, we acquire spatial-time features $\mathbf{f}$ through interpolation and hash lookup on $\mathbf{P}_{xyz}$ and $\mathbf{P}_{xyzt}$. Finally, $\mathbf{f}$ are decoded into density and view-independent color using tiny MLPs,

$$\psi : \mathbf{f} \mapsto \tau, \ \ \phi : \mathbf{f} \mapsto c. \tag{9}$$

The entire set of trainable parameters is denoted as θ_{4D}. We can optimize our dynamic NeRF by using the multi-view videos generated from 4DM, however, 4DM can only produce four orthogonal viewpoints at one time. Training with such sparse views often results in overfitting to the training viewpoints, as presented in Fig. 6. To mitigate this, we leverage 4D-aware SDS to optimize the dynamic NeRF, enabling effective rendering from novel viewpoints across the temporal dimension, which is crucial for 4D generation.

6

4D-aware SDS. Once 4DM is trained, we employ 4D-aware SDS loss to guide the optimization of our 4D representation. To be concrete, we utilize Eqn. 8 to render four F frames video V_r with timestamps $\mathbf{t} = \{\mathbf{t_1}, \mathbf{t_2}, ..., \mathbf{t_F}\}$ from four orthogonal viewpoints $\mathbf{c}_{\text{mv}}$. Our 4D-aware SDS injects Gaussian noise ϵ into V_r at a diffusion timestep t and passes to our multi-view video diffusion to provide gradients to update θ_{4D},

$$\nabla_{\theta_{\text{4D}}} \mathcal{L}_{\text{4D-SDS}} \approx \mathbb{E}_{(\mathbf{c}_{\text{mv}}, \mathbf{t}, \epsilon, t)} \left[2(V_r - \hat{V}_0) \frac{\partial V_r}{\partial \theta_{\text{4D}}} \right], \tag{10}$$

where $\hat{V}_0$ denotes the pseudo ground truth denoised from 4DM with the input video V and viewpoints $\mathbf{c}_{\text{mv}}$ as condition. Here, we replace original ϵ-based SDS loss with x_0-reconstruction loss as in [44].

Anchor Loss. Accurately estimating the elevation and azimuth of input monocular video within the canonical coordination is challenging, making it difficult to use the input video directly as supervision signals. Therefore, we utilize 4DM to produce four orthogonal videos conditioned on the input video and select the one with the viewpoint closest to that of the input video as the anchor video. This approach ensures that the anchor video maintains the same quality as the input and improves the results. Moreover, 4DM is currently limited to generating multi-view videos with 8 frames. When the input video exceeds 8 frames, we must apply our multi-view video diffusion model multiple times to generate anchor videos. However, this process may lead to temporally inconsistent results due to the stochasticity of the diffusion model, particularly when the viewpoint is far from the input video as shown in Fig. 5. This inconsistency would degrade the 4D generation performance. Finally, we devise an anchor loss $\mathcal{L}_{\text{a}}$ based on the anchor video to enhance the appearance details and facilitate the learning of dynamic NeRF. Since it is challenging for 4DM to ensure pixel-to-pixel alignment of the anchor video, we follow [10] to use image-level perceptual loss, i.e., LPIPS [64] and SSIM, for dynamic NeRF optimization,

$$\mathcal{L}_{\text{a}} = \lambda_1 \text{LPIPS}(I_r, I_a) + \lambda_2 \text{D-SSIM}(I_r, I_a), \tag{11}$$

where I_r and I_a represent the rendered video and anchor video, λ is the loss weight. Consequently, our total loss function for 4D generation is,

$$\mathcal{L}_{\text{4D}} = \mathcal{L}_{\text{4D-SDS}} + \mathcal{L}_{\text{a}} + \lambda_3 \mathcal{L}_{\text{orient}} + \lambda_4 \mathcal{L}_{\text{opacity}} + \lambda_5 \mathcal{L}_{\text{sparse}}, \tag{12}$$

here $\mathcal{L}_{\text{orient}}$, $\mathcal{L}_{\text{opacity}}$, and $\mathcal{L}_{\text{sparse}}$ are regularization loss in DreamFusion [38].

4 Experiments

Implementation Details. We implement 4DM under the Stable Diffusion framework and initialize it from the checkpoint of ImageDream. We train 4DM with multi-view videos with 256×256 resolutions for 30,000 steps with a batch size of 32, using the AdamW optimizer with a learning rate of 1e-4. The training takes about 2 days with 16 NVIDIA Tesla A100 GPUs. Additionally, for 4D generation experiments, we optimize dynamic NeRF representation in an end-to-end manner, avoiding utilizing multiple stages as in previous works.

Baselines. To evaluate our method, we compare to two video-to-4D approaches, namely Consistent4D [20] and DreamGaussian4D [42], and one text-to-4D approach 4D-fy [4]. We extend 4D-fy to video-prompt 4D generation by using ImageDream as the 3D-aware diffusion model. 4D-fy introduces hybrid SDS to blend gradients from multiple pre-trained diffusion models to create 4D contents. Consistent4D is the first study focusing on the video-to-4D task. They utilize a 3D-aware diffusion model to optimize a cascade dynamic NeRF and propose a consistency loss to address spatial-temporal inconsistency. DreamGaussian4D leverages 4D Gaussian Splatting for faster training.

4.1 Comparisons on 4D Generation

Qualitative Evaluation. To validate 4Diffusion for 4D generation, we compare it to Consistent4D [20], DreamGaussian4D [42], and 4D-fy [4] on monocular video-to-4D task. Here, we use 3 real-world videos and 3 synthetic videos from the Consistent4D dataset, as well as 3 images from the ImageDream. As discussed in Sec.A.1 of the supplementary materials, for text-image pairs from ImageDream, we utilize SVD to generate input videos. We illustrate the results in Fig. 7. 4D-fy [4] achieves state-of-the-art results on text-to-4D task and can be simply extended to video-prompt

7

Figure 7: 4D generation comparisons with 4D-fy [4], Consistent4D [20], and DreamGaussian4D [42].

4D generation by replacing 3D-aware diffusion model. Here, we utilize ImageDream as the 3D-aware diffusion model in 4D-fy. 4D-fy produces 4D contents with inconsistent temporal appearance, sometimes diverging significantly from the input video, as depicted in the first two columns of Fig. 7. This is primarily because integrating gradients from multiple diffusion models is difficult and they face challenges in multi-view spatial-temporal modeling. Consistent4D is the first work for 4D generation from monocular video. They employ an interpolation loss between two frames to enhance spatial-temporal consistency. However, they lack temporal consistency across frames, leading to poor appearance quality and flickers. DreamGaussian4D generates 4D contents with a blurred appearance and inaccurate geometry because GS struggles to model thin structures and large motions under unconstrained situations. In contrast, 4Diffusion generates high-quality 4D content with 4DM, which captures multi-view spatial-temporal correlations in a unified manner. Overall, our method achieves superior results, demonstrating its effectiveness. For more visualization results, please refer to our supplementary materials.

Quantitative Evaluation. We select 5 test cases from Objaverse, each consisting of a monocular input video and four orthogonal ground truth videos, which are not included in the training data, to evaluate our model. To evaluate image quality, we leverage CLIP-I [40] to measure the similarity.

8

Table 1: Quantitative evaluation on 4D generation.

	Image quality	Tem. Con.	Video quality	Spa. Con.	
	CLIP-I↑	CLIP-C↑	FVD↓	LPIPS↓	PSNR↑
4D-fy[4]	0.8658	0.9487	1042.3	0.2254	14.24
Consistent4D[20]	0.9216	0.9723	706.07	0.1593	16.70
DreamGaussian4D[42]	0.8898	0.9710	760.18	0.1793	15.97
Ours(w/o $\mathcal{L}_{\text{4D-SDS}}$)	0.8195	0.9503	1546.4	0.2356	13.92
Ours(w/o $\mathcal{L}_{\text{a}}$)	0.8823	0.9720	853.57	0.1589	17.20
Ours	**0.9310**	**0.9798**	**417.63**	**0.1199**	**19.07**
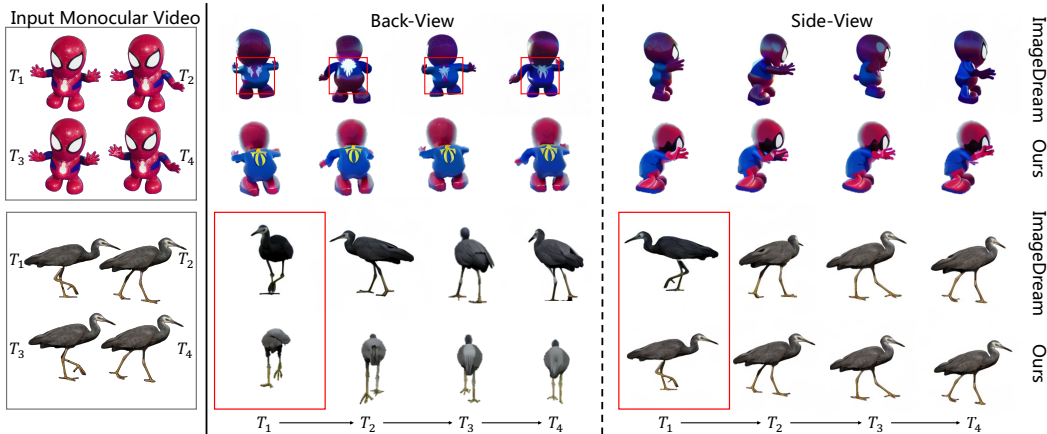
Figure 8: The illustration of synthesized multi-view videos from 4DM and ImageDream [52]. 4DM produces more spatial-temporal consistent results than ImageDream. T denotes the timestep of video clips. All results are generated from DDIM [48] sampler.

We also calculate FVD to evaluate the video quality. We compute LPIPS [64] and PSNR metrics to evaluate the spatial consistency. Here, we use ground truth videos for novel viewpoints to compute the above metrics. Moreover, we compute CLIP-C between frames in each synthetic video to evaluate temporal consistency. Tab. 1 presents the results, clearly demonstrating that 4Diffusion outperforms other methods on all metrics.

4.2 Multi-view Video Generation

Qualitative Evaluation. In this section, we evaluate the multi-view video generation quality produced by 4DM using the same input videos as described in qualitative evaluation in Sec 4.1. We employ ImageDream to synthesize a set of multi-view images as pseudo multi-view video by taking each frame of the input monocular video as an image prompt. Fig. 8 illustrates results with the first two columns corresponding to the input monocular video. Although ImageDream excels at synthesizing spatially consistent images, it struggles to model temporal correlations, leading to inconsistent temporal appearances, such as the icon on the back of Spiderman. Comparatively, 4DM effectively captures reasonable temporal information using the motion module, even when trained on a small curated dataset. Moreover, our model preserves the generalization ability of ImageDream, allowing us to generate high-fidelity multi-view videos, even beyond the distribution of our training dataset. As the last two rows of Fig. 8 show, 4DM produces spatially consistent videos by sharing information across spatial and temporal dimensions to constraint the generation process while ImageDream occasionally fails to generate videos coherent to the viewpoint.

Quantitative Evaluation. We use the same test cases described in quantitative evaluation in Sec 4.1, alongside the test data provided by Consistent4D, to evaluate 4DM. To account for the stochasticity of the diffusion model, we conduct five runs for each test case and report the average metrics. Tab. 2 shows comparative results. Despite the comparable performance in CLIP-I, 4DM excels in

9

Table 2: Quantitative evaluation on multi-view video generation. Here, we employ Consistent4D test dataset to evaluate 4DM and ImageDream. 'Spa. Con.' and 'Tem. Con.' refer to spatial consistency and temporal consistency, respectively.

	Image quality	Tem. Con.	Video Quality	Spa. Con.	
	CLIP-I↑	CLIP-C↑	FVD↓	LPIPS↓	PSNR↑
ImageDream[52]	0.9165	0.9320	465.94	0.1536	16.57
Ours(w/ whole)	0.8872	0.9478	583.79	0.1763	15.28
Ours(4DM)	**0.9260**	**0.9601**	**427.34**	**0.1346**	**17.88**

generating spatial-temporally consistent multi-view videos, a primary focus of our research. This is evidenced by the superior performance on metrics such as CLIP-C, FVD, LPIPS, and PSNR, which better capture the spatial and temporal fidelity of video content. These metrics demonstrate that our method effectively balances image quality with temporal consistency, making it a robust solution for multi-view video generation.

4.3 Ablation study and analysis

Effectiveness of the Curated Multi-view Video Dataset. To evaluate the importance and effectiveness of the selected high-quality multi-view videos, we use the entire animated 3D shapes from Objaverse and render multi-view videos to fine-tune 4DM (Ours w/ whole). The results are shown in Tab. 2. Given the presence of numerous flawed data within the entire dataset, it compromises the image quality of ImageDream and encounters challenges in precisely capturing spatial-temporal correlations, demonstrating the importance of high-quality datasets for fine-tuning 4DM.

Figure 9: Ablation studies on 4D-aware SDS loss and the anchor loss.

4D-aware SDS Loss. To evaluate the effect of our 4D-aware SDS loss, we substitute the 4DM with ImageDream and use 3D-aware SDS loss based on ImageDream to optimize dynamic NeRF representation. As Fig. 9 depicted, inconsistent temporal textures, such as the leg of the squirrel, emerge due to the lack of temporal modeling of ImageDream, underscoring the significance of capturing spatial-temporal correlations in 4DM. The quantitative results presented in Tab. 1 indicate the significance of our 4D-aware SDS loss.

Anchor Loss. We also assess the impact of the proposed anchor loss. As illustrated in Fig. 9, capturing detailed appearance features, such as the eyes of the squirrel, proves challenging without the anchor loss. Conversely, the anchor images furnish visual clues to facilitate the learning of 4D representation, resulting in high-quality 4D content. The quantitative results Tab. 1 demonstrate the crucial role of our anchor loss.

5 Conclusion

In this paper, we present 4Diffusion for 4D generation from a monocular video. Our method proposes a multi-view video diffusion model 4DM based on a 3D-aware diffusion model for multi-view video generation and provides multi-view spatial-temporal guidance for 4D generation. 4DM captures spatial-temporal correlations and preserves the characteristics of the origin 3D-aware diffusion model even when training on a small curated dataset. Then, we combine 4D-aware SDS loss and an anchor loss based on 4DM to optimize our hash-encoded dynamic NeRF, resulting in spatial-temporally consistent 4D contents coherent with the input monocular video.

10

Acknowlegements

The work is supported by the National Key R&D Program of China (No. 2022ZD0160102), the National Natural Science Foundation of China under Grant No. 62102150, and the Science and Technology Commission of Shanghai Municipality under Grant No. 23QD1400800.

References

[1] Deepfloyd. https://github.com/deep-floyd/IF. 2023.

[2] Zeroscope text-to-video model. https://huggingface.co/cerspense/zeroscope_v2_576w. 2023.

[3] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *CVPR*, pages 12608–12618, 2023.

[4] Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B Lindell. 4d-fy: Text-to-4d generation using hybrid score distillation sampling. In *CVPR*, pages 7996–8006, 2024.

[5] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Rodin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[6] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *CVPR*, pages 130–141, 2023.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021.

[8] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, pages 16123–16133, 2022.

[9] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *ICCV*, pages 22246–22256, 2023.

[10] Zilong Chen, Yikai Wang, Feng Wang, Zhengyi Wang, and Huaping Liu. V3d: Video diffusion models are effective 3d generators. *arXiv preprint arXiv:2403.06738*, 2024.

[11] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, pages 13142–13153, 2023.

[12] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In *ICCV*, pages 14300–14310, 2023.

[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 *CVPR*, pages 12479–12488, 2023.

[14] Xun Guo, Mingwu Zheng, Liang Hou, Yuan Gao, Yufan Deng, Pengfei Wan, Di Zhang, Yufan Liu, Weiming Hu, Zhengjun Zha, Haibin Huang, and Chongyang Ma. I2v-adapter: A general image-to-video adapter for video diffusion models. *arXiv preprint arXiv:2312.16693*, 2023.

[15] Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation. https://github.com/threestudio-project/threestudio, 2023.

[16] 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. In *ICLR*, 2024.

11

[17] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023.

[18] 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. In *ICLR*, 2024.

[19] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *CVPR*, pages 4220–4230, 2024.

[20] Yanqin Jiang, Li Zhang, Jin Gao, Weimin Hu, and Yao Yao. Consistent4d: Consistent 360° dynamic object generation from monocular video. In *ICLR*, 2024.

[21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 42(4), 2023.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[23] 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. In *ICLR*, 2024.

[24] Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: Efficient sampling accelerates nerfs. In *ICCV*, pages 18537–18546, 2023.

[25] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video. In *CVPR*, pages 5521–5531, 2022.

[26] 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. In *CVPR*, pages 300–309, 2023.

[27] 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.

[28] Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. In *CVPR*, pages 8576–8588, 2024.

[29] 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 *ICCV*, pages 9298–9309, 2023.

[30] Xian Liu, Xiaohang Zhan, Jiaxiang Tang, Ying Shan, Gang Zeng, Dahua Lin, Xihui Liu, and Ziwei Liu. Humangaussian: Text-driven 3d human generation with gaussian splatting. In *CVPR*, pages 6646–6657, 2024.

[31] 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. In *ICLR*, 2024.

[32] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion. In *CVPR*, pages 9970–9980, 2024.

[33] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. In *NeurIPS*, 2023.

[34] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360° reconstruction of any object from a single image. In *CVPR*, pages 8446–8455, 2023.

[35] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[36] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *CVPR*, pages 4328–4338, 2023.

[37] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 41(4):1–15, 2022.

12

[38] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023.

[39] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. In *ICLR*, 2024.

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Sutskever Ilya. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

[41] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, Yuanzhen Li, and Varun Jampani. Dreambooth3d: Subject-driven text-to-3d generation. In *ICCV*, 2023.

[42] Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023.

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[44] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *ICLR*, 2024.

[45] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *CVPR*, pages 20875–20886, 2023.

[46] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023.

[47] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, and Yaniv Taigman. Text-to-4d dynamic scene generation. In *ICML*, pages 31915–31929, 2023.

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[49] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *ECCV*, pages 1–18, 2024.

[50] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *ICLR*, 2024.

[51] 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. In *ICCV*, 2023.

[52] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.

[53] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, and Baining Guo. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *CVPR*, pages 4563–4573, 2023.

[54] 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.

[55] 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. In *NeurIPS*, 2023.

[56] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *CVPR*, pages 20310–20320, 2024.

[57] Zijie Wu, Chaohui Yu, Yanqin Jiang, Chenjie Cao, Fan Wang, and Xiang Bai. Sc4d: Sparse-controlled video-to-4d generation and motion transfer. *arXiv preprint arXiv:2404.03736*, 2024.

13

[58] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. In *ICLR*, 2024.

[59] Jiayu Yang, Ziang Cheng, Yunfei Duan, Pan Ji, and Hongdong Li. Consistnet: Enforcing 3d consistency for multi-view images diffusion. In *CVPR*, pages 7079–7088, 2024.

[60] Lior Yariv, Omri Puny, Natalia Neverova, Oran Gafni, and Yaron Lipman. Mosaic-sdf for 3d generative models. In *CVPR*, pages 4630–4639, 2024.

[61] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[62] Yuyang Yin, Dejia Xu, Zhangyang Wang, Yao Zhao, and Yunchao Wei. 4dgen: Grounded 4d content generation with spatial-temporal consistency. *arXiv preprint arXiv:2312.17225*, 2023.

[63] Yifei Zeng, Yanqin Jiang, Siyu Zhu, Yuanxun Lu, Youtian Lin, Hao Zhu, Weiming Hu, Xun Cao, and Yao Yao. Stag4d: Spatial-temporal anchored generative 4d gaussians. *arXiv preprint arXiv:2403.14939*, 2024.

[64] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

[65] Yuyang Zhao, Zhiwen Yan, Enze Xie, Lanqing Hong, Zhenguo Li, and Gim Hee Lee. Animate124: Animating one image to 4d dynamic scene. *arXiv preprint arXiv:2311.14603*, 2023.

[66] Yufeng Zheng, Xueting Li, Koki Nagano, Sifei Liu, Otmar Hilliges, and Shalini De Mello. A unified approach for text-and image-guided 4d scene generation. In *CVPR*, pages 7300–7309, 2024.

[67] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *CVPR*, pages 10324–10335, 20234.

14

A Supplemental material

A.1 More Implementation Details

Table 3: Hash encoding parameters of P_{xyz} and P_{xyzt}

Parameter	Value
Number of levels	16
Hash table size	2^{19}
Number of feature dimensions per level	2
Coarsest resolution	16
Finest resolution	4096
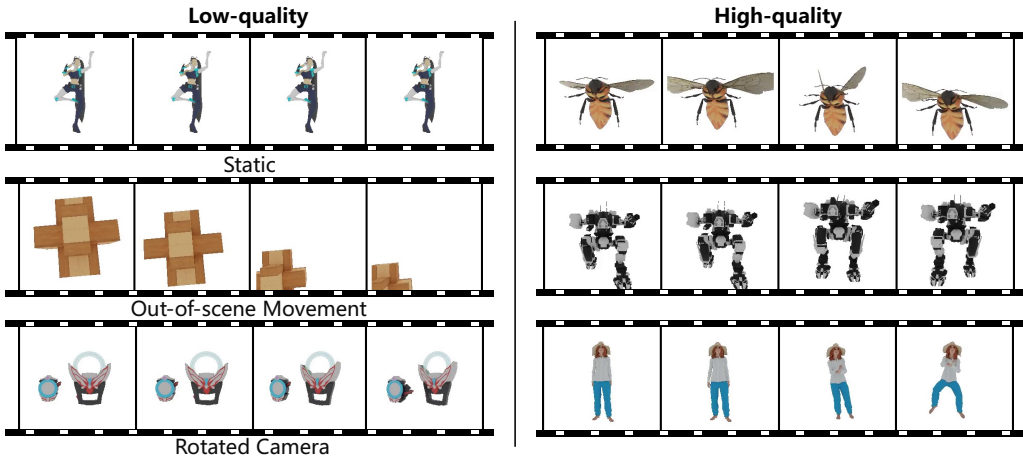
Figure 10: The illustration of our training dataset. We manually filter out animated 3D data with static motion, out-of-scene movement, or rotated camera to curate a dataset with high-quality appearance and realistic motion.

Datasets. We utilize Objaverse dataset [11] to train our multi-view diffusion model, as described in Sec. 3.2. Objaverse dataset comprises a vast collection of 3D shapes with descriptive captions, tags, and animations. We manually filter out animated 3D shapes that contain static objects, out-of-scene movement, rotated cameras, or meaningless objects, resulting in 926 high-quality 3D animated models, as depicted in Fig. 10. We apply Blender to render 32 videos with azimuth angles uniformly ranging from $[-180°, 180°]$ and an elevation angle of $0°$ for each animated 3D model. In our experiments, we use the dataset released by Consistent4D [20], test cases from Objaverse, and text-image pairs from ImageDream [52] project page. Specifically, for text-image pairs, we leverage Stable Video Diffusion V1.1 [5] to produce monocular videos for 4D generation.

4D Generation. We implement our 4D generation model under threestudio framework [15]. Our hash-encoded dynamic NeRF representation utilizes the parameters detailed in Tab. 3. Following [55, 4], we anneal the timesteps of diffusion models from $t \in [0.98, 0.98]$ to $t \in [0.02, 0.25]$ over the initial 5,000 iterations and set the diffusion CFG to 5.0. The loss weights λ_1, λ_2, λ_4 are set to 200, 100, and 100, respectively. Addtionally, λ_3 linearly increases from 10 to 1000 during the first 5,000 iterations and λ_5 is fixed at 100 after the initial 10,000 iterations. The model is trained with AdamW optimizer for 35,000 iterations with a learning rate of 1e-2 except for the decoded MLPs, where the learning rate is adjusted to 1e-3. It takes around 12 hours to train the model on one NVIDIA Tesla A100 GPU.

Volume Rendering. We employ NerfAcc [24] as our rendering pipeline, which leverages an occupancy grid to store the opacity of a scene. This approach accelerates volume rendering and reduces computations. We adopt a shared occupancy grid by representing the maximum opacity of the scene across all frames, facilitating its application to dynamic scenes. Additionally, we set the background of the rendered images to white. For the resolution of rendered images, we follow the

15
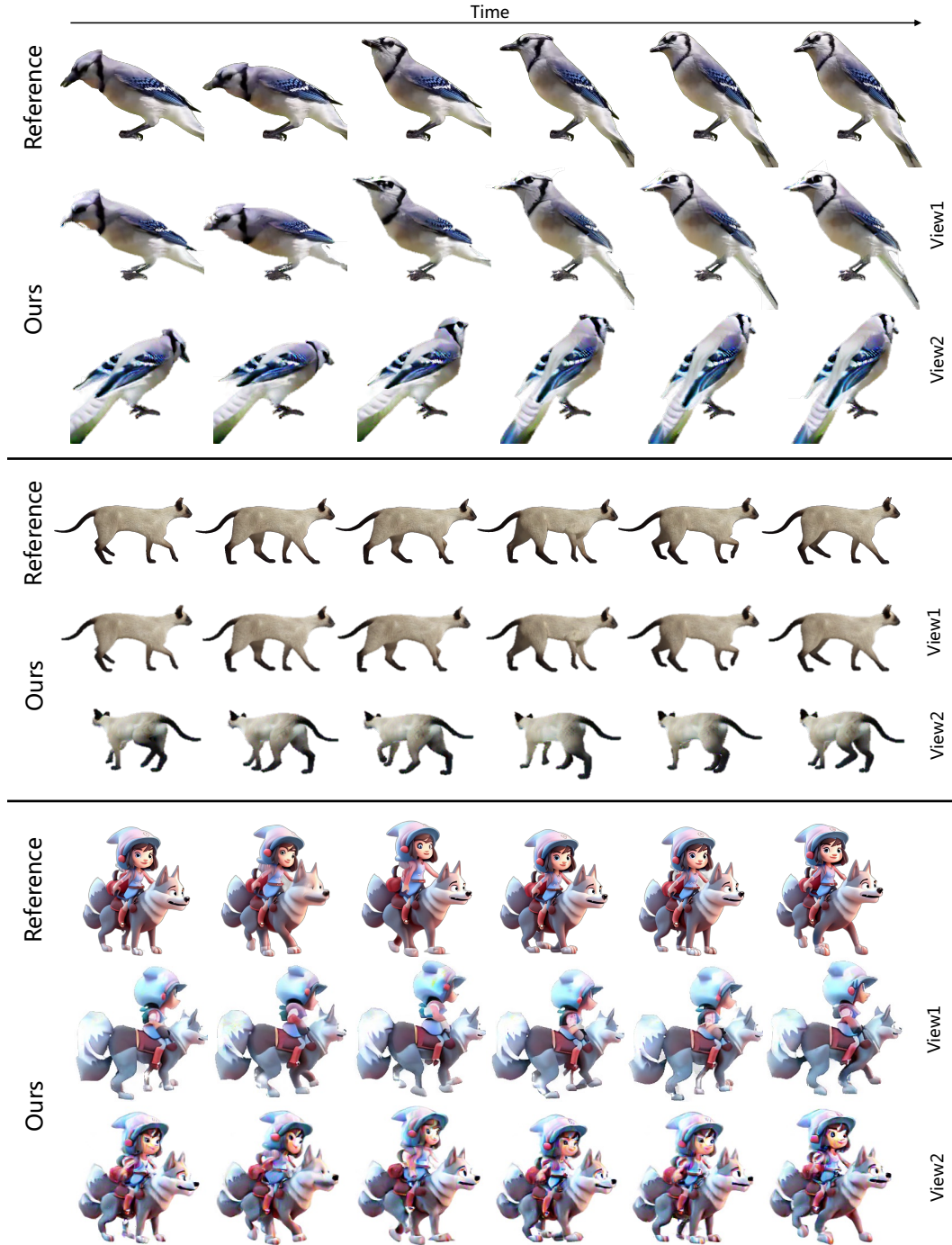
Figure 11: The illustration of 4D generation results of 4Diffusion.

configuration of ImageDream [52]. We maintain fixed camera distances at 1.1 to enhance the stability of the optimization process.

A.2 More Results

4D Generation. In Fig. 11, we showcase additional results of our 4D generation results. To gain a more intuitive understanding, we encourage readers to view the supplementary videos. Moreover, we

16

| Input | 4D-fy | Dream Gaussian4D | Consistent4D | Ours | GT |
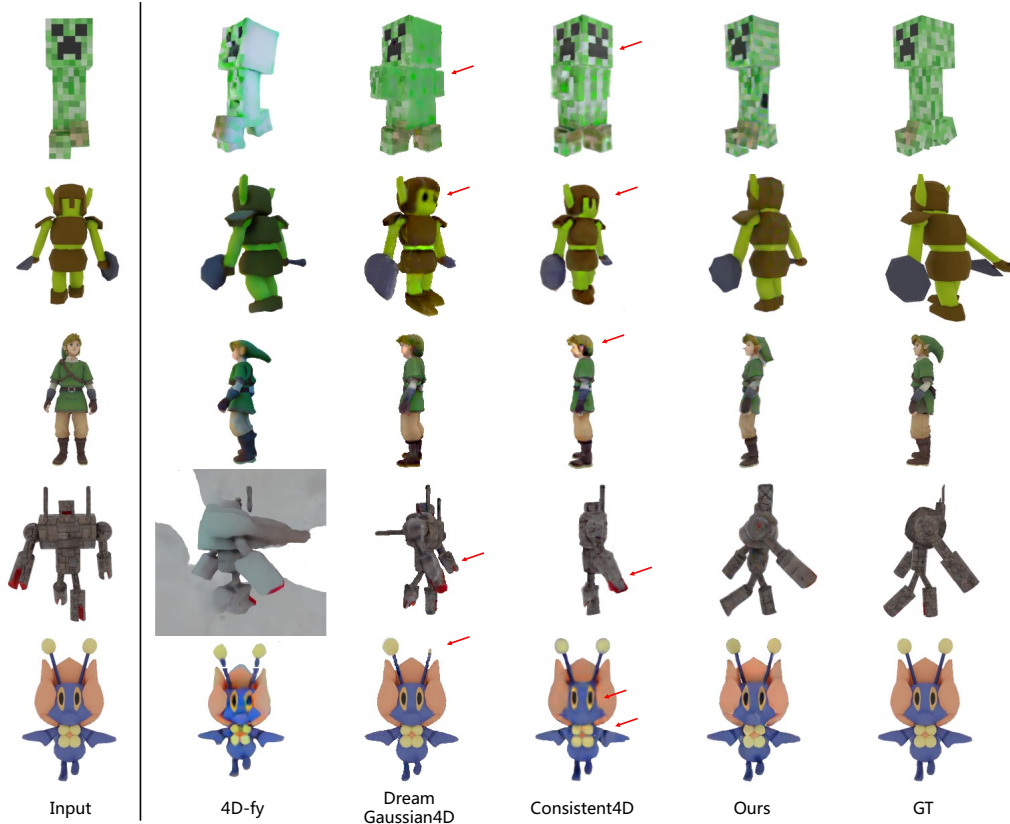
Figure 12: 4D generation comparisons with 4D-fy, DreamGaussian4D, Consistent4D. These test cases are selected from Objaverse dataset, which are not included in the training data of 4DM.

show more comparisons on 4D generation as shown in Fig. 12, Consistent4D and DreamGaussian4D encounter the multi-face problem while our method generates spatial-temporally consistent contents.

Text-to-4D. To Further validate the effectiveness of 4Diffusion, we conduct experiments on text-to-4D task. Specifically, we first employ SDXL to generate images conditioned on text prompts. Subsequently, we utilize SVD V1.1 to produce monocular videos for 4D generation. Finally, we follow the procedure outlined in the main paper to generate 4D content from the monocular videos. As illustrated in Figure 13, our approach yields high-quality 4D content from text prompts, thereby demonstrating its effectiveness.

A.3 Limitations

Our method can be improved in the following aspects: **1)** Our multi-view video diffusion model is constrained by the capability of the base model and the scale of high-quality training data. We believe that improving the base model and scaling up the high-quality dataset can obtain a better model. **2)** Our 4D generation pipeline relies on heavily volumetric rendering, causing slow training speed. We believe advances in 3D and GS can potentially solve these problems.

A.4 Broader Impacts

Our work paves the way for high-quality 4D content generation, reducing the extensive manual effort for artists and novices. Although our method is not designed for generating humans, it may be extended and misused, potentially influencing human perceptions.

17
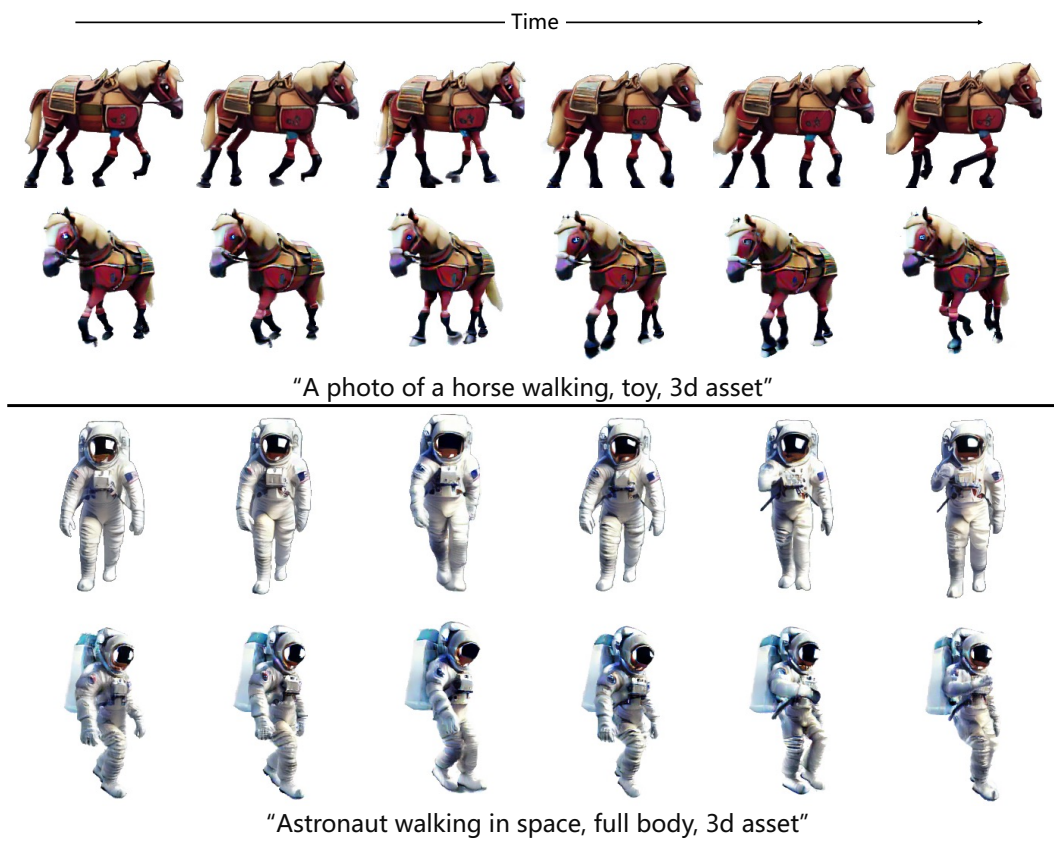
Figure 13: The illustration of text-to-4D results of 4Diffusion.

18

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim the contributions and describe the challenges, which our method tries to solve.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In our supplementary material, we discuss the limitations of our method and the potential future work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

19

Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We describe the implementation details for our experiments in our paper and all the results can be reproduced.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

20

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We release our code and data at `https://aejion.github.io/4diffusion`.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide training and test details in experiment which can help readers to understand the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: In our experiments, we do not report error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

21

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide the computation resources required to reproduce our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Our paper follows the NeurIPS Code of Ethics in every respect.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss broader impacts in our supplementary material.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

22

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our pre-trained model and method present a low risk of misuse or dual-use, as the model is trained on object-centric datasets that contain no human data.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We follow the license of each asset we have used and cite them properly. Moreover, we also state the version of these assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

23

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: Our paper provides no new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

24