

Animate3D: Animating Any 3D Model with Multi-view Video Diffusion

Yanqin Jiang^{1,2*} Chaohui Yu^{3,4*} Chenjie Cao^{3,4} Fan Wang^{3,4}
Weiming Hu^{1,2,5} Jin Gao^{1,2†}

¹State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), CASIA

²School of Artificial Intelligence, University of Chinese Academy of Sciences

³DAMO Academy, Alibaba Group ⁴Hupan Lab

⁵School of Information Science and Technology, ShanghaiTech University

jiangyanqin2021@ia.ac.cn

{huakun.ych, caochenjie.ccj, fan.w}@alibaba-inc.com

{jin.gao, wmhu}@nlpr.ia.ac.cn

<https://animate3d.github.io/>

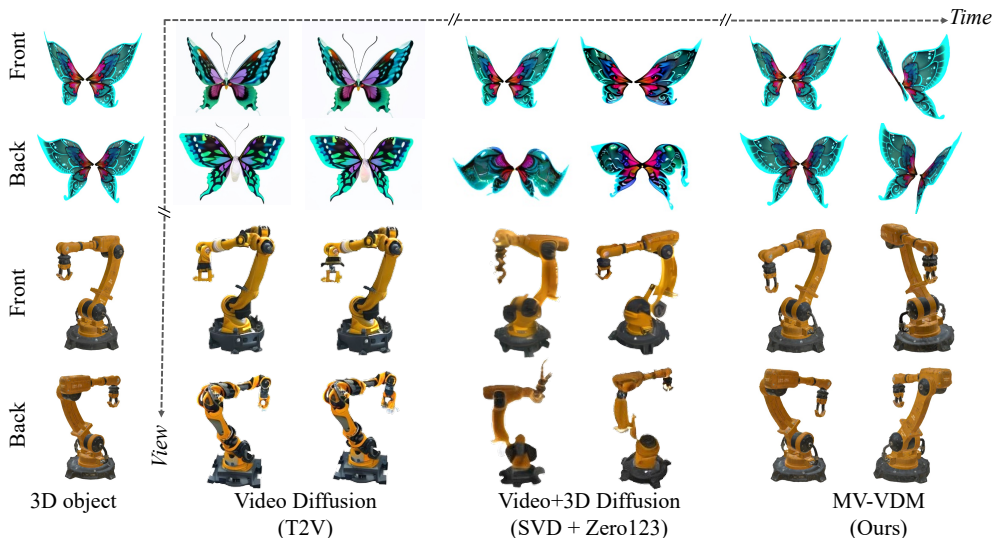


Figure 1: Different supervision for 4D generation. MV-VDM shows superior spatiotemporal consistency than previous models. Based on MV-VDM, we propose Animate3D to animate any 3D model.

Abstract

Recent advances in 4D generation mainly focus on generating 4D content by distilling pre-trained text or single-view image-conditioned models. It is inconvenient for them to take advantage of various off-the-shelf 3D assets with multi-view attributes, and their results suffer from spatiotemporal inconsistency owing to the inherent ambiguity in the supervision signals. In this work, we present Animate3D, a novel framework for animating any static 3D model. The core idea is two-fold: 1) We propose a novel multi-view video diffusion model (MV-VDM) conditioned on multi-view renderings of the static 3D object, which is trained on our presented large-scale multi-view video dataset (MV-Video). 2) Based on MV-VDM, we introduce a framework combining reconstruction and 4D Score Distillation Sampling (4D-SDS) to leverage the multi-view video diffusion priors for animating

*Equal contribution. Work done during Yanqin’s internship at DAMO Academy, Alibaba Group.

†Corresponding author.

3D objects. Specifically, for MV-VDM, we design a new spatiotemporal attention module to enhance spatial and temporal consistency by integrating 3D and video diffusion models. Additionally, we leverage the static 3D model’s multi-view renderings as conditions to preserve its identity. For animating 3D models, an effective two-stage pipeline is proposed: we first reconstruct motions directly from generated multi-view videos, followed by the introduced 4D-SDS to refine both appearance and motion. Benefiting from accurate motion learning, we could achieve straightforward mesh animation. Qualitative and quantitative experiments demonstrate that Animate3D significantly outperforms previous approaches. Data, code, and models are open-released.

1 Introduction

3D content creation has garnered significant attention due to its wide applicability in AR/VR, gaming, and movie industry. With the development of diffusion models [45, 44, 39, 20, 51, 6, 19] and large-scale 3D object datasets [13, 12, 37, 65, 70, 14], recent 3D foundational generations have seen extensive exploration through fine-tuned text-to-image (T2I) diffusion models [31, 40, 41, 52, 32, 49], as well as training large reconstruction models from scratch [22, 61, 47, 67, 54], leading the 3D assets creation to a new era. Despite the significant progress in static 3D representation, this momentum has not been paralleled in the realm of dynamic 3D content generation, also known as 4D generation.

4D generation is more challenging due to the difficulty of simultaneously maintaining spatiotemporal consistency in visual appearance and dynamic motion. In this paper, we mainly focus on two challenges: 1) *No foundational 4D generation models to unify both spatial and temporal consistency.* Though recent 4D generation works [43, 24, 5, 69, 29, 38, 63, 59, 4, 57, 66, 16] separately distill knowledge from pre-trained T2I and 3D diffusion models [39, 41, 31] and video diffusion models [51, 2, 6] to model multi-view spatial appearance and temporal motions respectively, we clarify that such a detached learning way suffers from inevitable accumulation of appearance degradation as the motion changed, as shown in Fig. 1 (SVD+Zero123). 2) *Failing to animate existing 3D assets through multi-view conditions.* With the development of 3D generations, animating existing high-quality 3D content becomes a common demand. However, previous works about 4D modeling from video [24, 57] or based on generated 3D assets [5, 29, 38, 66] are all based on text [39, 41] or *single-view* [31, 40] conditioned models, struggling to faithfully preserve multi-view attributes during the 4D generation, such as the back of butterfly in Fig. 1 is ignored by Zero123 [31].

To address these issues, we advocate for an approach better suited for 4D generation, that is, **animating any off-the-shelf 3D models with unified spatiotemporal consistent supervision**. In this way, it would be convenient to directly take advantage of various fast-developing 3D generation and reconstruction approaches based on a single foundational model, eliminating the accumulation of errors in modeling appearance and motion.

To this end, we propose a novel 4D generation framework called Animate3D in this paper, which can be divided into a foundational 4D generation model and a joint 4D Gaussian Splatting (4DGS) optimization. Formally, the foundational 4D model is a Multi-View Video Diffusion Model (MV-VDM) built upon the 3D generation model, MVDream [41], which can synchronously synthesize multi-view images with various temporal motions. Specifically, to better inherit the prior in previous 3D and video diffusion models trained on large-scale data, we propose a learnable plug-and-play spatiotemporal attention module, building upon the motion module in video diffusion [18, 6, 20] to expand the attention learning from the temporal domain to the spatial and temporal domain. Moreover, MV-VDM also includes the ability to refer to multi-view images, sufficiently preserving the identity and details of off-the-shelf 3D assets. Specifically, given multi-view images rendered from existing 3D assets or collected from real-world objects, we expand adaptive image-to-video work, I2V-Adapter [17], to multi-view version, called MV2V-Adapter, incorporating multi-view conditions to 4D learning through additionally spatial features and text embeddings. Enhanced by the multi-view appearance injection, we can disentangle the appearance learning from the motion learning, ensuring MV-VDM focuses on learning natural and coherent dynamic motions.

To further enable impressive animations from 3D objects that can be observed at any viewpoint and time, we jointly optimize the 4DGS [55] through both reconstruction and 4D Score Distillation Sampling (4D-SDS) losses based on our unified MV-VDM. Benefiting from the spatiotemporal

consistent multi-view video generations, 4DGS can be roughly converged to proper results with only reconstruction loss, while 4D-SDS further improves the details and fine-grained motions. The Gaussian trajectory learned by our framework is surprisingly accurate and could be used to directly animate the mesh.

The main dilemma in building a foundational 4D generation model lies in the rarity of large-scale 4D datasets, which is non-trivial to collect but the key factor to drive our MV-VDM. In this work, we make the first attempt to build a large-scale multi-view video (4D) dataset, dubbed MV-Video. Specifically, MV-Video comprises about **84K** animations that are available under a public license, consisting of about **38K** animated 3D objects at all, which are rendered into over **1.3M** multi-view videos with minigt4-video [3] generated prompts, to serve as the training dataset for our 4D foundation model.

We highlight the contribution of this paper as follows: 1) Animate3D is the first 4D generation framework to animate any 3D objects with detailed multi-view conditions. The framework is further extended to achieve mesh animation without skeleton rigging. 2) We propose the foundational 4D generation model, MV-VDM, to jointly model spatiotemporal consistency. 3) We present the largest 4D datasets MV-Video collected with about 84K animations and over 1.3M multi-view videos. Extensive experiments demonstrate that our data-driven approach can generate spatiotemporal consistent 4D objects, significantly outperforming previous counterparts.

2 Related Work

3D Generation. Early 3D generation works optimized single 3D object with CLIP loss [36] or Score Distillation Sampling (SDS) [33] from 2D text-to-image (T2I) diffusion models. Since the models providing supervision lacked 3D prior, those works usually suffered from spatial inconsistency, *i.e.*, multi-face Janus problem [50, 28, 53]. To tackle this problem, on the one hand, some works [31, 40, 41, 32, 35] lifted the T2I diffusion to multi-view image diffusion by injecting new spatial attention layers and fine-tuning on large-scale 3D synthetic datasets [13, 12]. Although 3D consistency was improved, these optimization-based methods still required a relatively long time to optimize a 3D object. On the other hand, some feed-forward 3D generation foundation models [22, 61, 47, 67, 54, 60, 30], also trained on large-scale 3D datasets, were able to produce a good-quality 3D object in several seconds in an inference-only way. Inspired by the success of the data-driven approaches in 3D generation, we aim to construct a large-scale 4D generation dataset and take the pioneering step towards developing foundation models for 4D generation.

Video Generation. Video generation works started with text-to-video (T2V) generation [20, 42, 2, 18, 6, 19, 7, 10], subsequently followed by image-to-video (I2V) approaches [64, 17, 58, 6]. Previous T2V works usually built upon T2I diffusion models [20, 42, 19, 18, 21], leveraging their pre-trained weights by leaving the spatial blocks unchanged and inserting new temporal blocks to model the temporal camera or object motions. The I2V works [58, 17, 64], building upon the aforementioned T2V methods, typically incorporate image semantics into video models. This is achieved through cross-attention mechanisms between noisy frames and the conditional image, while retaining the motion module design from the T2V models unaltered. We draw inspiration from the development paradigm of video generation to design our 4D generation foundation model, which is a multi-view image conditioned multi-view video diffusion model building upon pre-trained multi-view 3D and video diffusion models.

4D Generation. The pioneering work of 4D generation is MAV3D [43], which is a text and image-conditioned 4D generation framework. MAV3D first proposed a multi-stage pipeline to optimize the static 3D object generation through the T2I model and subsequently learn motions from the T2V model [42]. Following works [68, 5, 29, 69, 4, 59] adopted a similar pipeline, and they further found that employing T2I [39] and 3D-SDS [41] are crucial for both object generation and motion learning stages. Without them, the quality of the generated object’s appearance suffered a remarkable decline, and the motion-learning process was prone to failure. Very recently, Consistent4D [24] proposed a video-to-4D generation task, which used single-view video reconstruction and SDS from Zero123 [31] for motion and appearance learning. This paradigm was adopted by following works [38, 63, 66, 57, 16, 11] and extended to text/image-to-video then video-to-4D generation. All aforementioned works heavily depend on the foundational model for SDS to preserve objects’ appearance and attributes. However, existing 3D diffusion models struggle to refer to multi-view

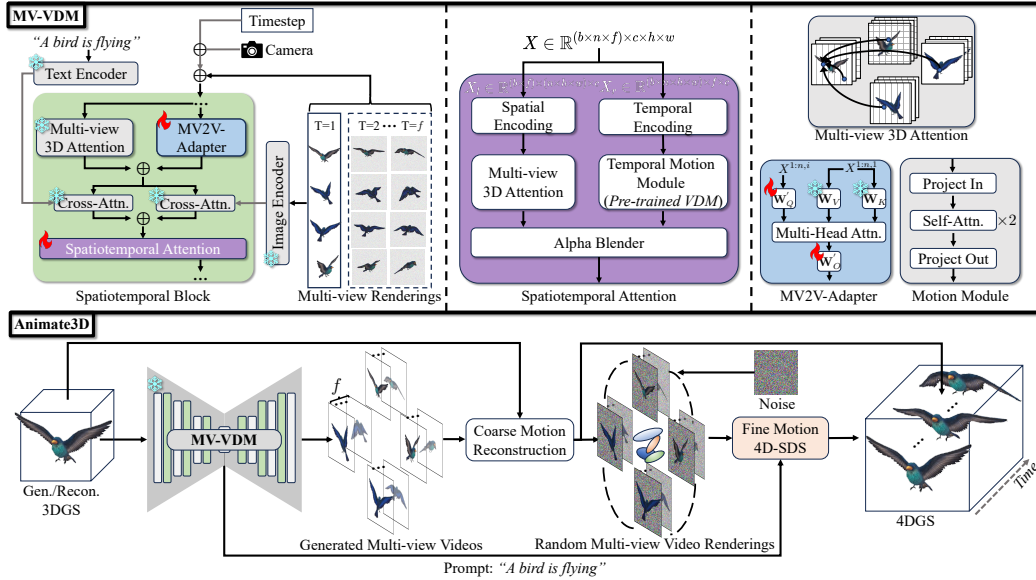


Figure 2: Illustration of our proposed multi-view video diffusion model—**MV-VDM** (upper part) and our **Animate3D** framework (lower part). MV-VDM, trained on our presented large-scale 4D dataset MV-Video, can generate spatiotemporal consistent multi-view videos. Animate3D, based on MV-VDM, combines reconstruction and 4D-SDS optimization to animate any static 3D models.

conditions, restricting their broader applications to animate various off-the-shelf 3D assets without losing their multi-view attributes.

Furthermore, it is worth noting that existing 4D generation methods suffer from another issue, *i.e.*, spatial and temporal inconsistency [43, 5, 29, 24, 38]. Because the diffusion models used for SDS were never trained with multi-view video (4D) datasets, missing the critical capacity to formulate spatial and temporal consistency simultaneously. Thus previous methods failed to properly trade off a good balance between appearance and motion learning. Please refer to Sec. B in appendix for detailed discussion and comparison.

In this work, we resort to disentangle 3D object generation/reconstruction and motion learning through a foundational 4D generation model, and propose a novel framework to animate any static 3D object with consistent multi-view attributes.

3 Method

Given a static 3D model, our goal is to animate it with a text prompt and use its multi-view renderings as image condition. This 4D generation task is particularly challenging as it requires ensuring spatial and temporal consistency of the appearance and motion, compatibility with the prompt, and preserving the identity of the static object. To address these challenges more fundamentally, we propose a novel framework, Animate3D, to animate any static 3D object. As depicted in Fig. 2, we divide the task into two parts: learning a multi-view video diffusion model (MV-VDM), and animating 3D object with multi-view videos generated by MV-VDM.

3.1 Multi-view Video Diffusion Model (MV-VDM)

We propose a novel multi-view image conditioned multi-view video diffusion model, named MV-VDM. To inherit prior knowledge acquired by spatially consistent 3D models and temporally consistent video models trained on large-scale datasets, we advocate a baseline architecture by integrating them to utilize their pre-trained weights. In this work, we take MVDream [41] and AnimateDiff [18] for the 3D and the video diffusion model, respectively. To enhance the spatiotemporal consistency and ensure compatibility with the prompts and the object’s multi-view images, we propose an efficient plug-and-play spatiotemporal attention module combined with an image-conditioning approach. Our

MV-VDM is trained on our presented large-scale multi-view video dataset, MV-Video, which is introduced in Sec. 4.

Spatiotemporal Attention Module. As illustrated in Fig. 2, we insert a novel spatiotemporal attention module after cross-attention layers. The proposed spatiotemporal attention module comprises two parallel branches: the left branch is for spatial attention, and the right branch is for temporal attention. For spatial attention, we adopt the same architecture as the multi-view 3D attention in MVDream [41]. Specifically, the original 2D self-attention layer is converted into 3D by connecting n different views. In addition, we incorporate 2D spatial encoding, specifically sinusoidal encoding, into the latent features to enhance spatial consistency. As for temporal attention, we keep all designs of the temporal motion module from the video diffusion model [18] unchanged in order to reuse their pre-trained weights. Based on the features of these two branches, we employ an alpha blender layer with a learnable weight to achieve features with enhanced spatiotemporal consistency. It is worth noting that we do not apply spatiotemporal attention across all views and frames due to the prohibitive GPU memory requirements that render training infeasible. Instead, our parallel-branch design offers an efficient and practical alternative. Specifically, we first reshape the input feature of spatiotemporal attention module $X \in \mathbb{R}^{(b \times n \times f) \times c \times h \times w}$ into two forms, $X_l \in \mathbb{R}^{(b \times f) \times (n \times h \times w) \times c}$ for spatial branch and $X_r \in \mathbb{R}^{(b \times n \times h \times w) \times f \times c}$ for temporal branch. The spatiotemporal attention is then computed as:

$$X_{out} = \mu \cdot \text{Attention}_{\text{spatial}}(X_l W_Q^s, X_l W_K^s, X_l W_V^s) W_O^s + (1 - \mu) \cdot \text{Attention}_{\text{temporal}}(X_r W_Q^t, X_r W_K^t, X_r W_V^t) W_O^t, \quad (1)$$

where μ denotes the learnable weight, $W_Q^{s/t}$, $W_K^{s/t}$, $W_V^{s/t}$, $W_O^{s/t}$ represent the corresponding projection matrices. b, n, f, h, w, c are the batch size, views, frames, height, width, and channels of the image features, respectively.

Multi-view Images Conditioning. Inspired by I2V-Adapter [17], we add a new attention layer, termed MV2V-Adapter, parallel to the existing frozen multi-view 3D self-attention layer within the proposed spatiotemporal block, as shown in Fig. 2. Concretely, noisy frames are first concatenated along the spatial dimension. These are then used to query the rich contextual information from the multi-view conditional frames, which are extracted using the frozen 3D diffusion model. Next, we add the output of the MV2V-Adapter layer to that of the original multi-view 3D attention layer of MVDream. Thus, for each frame $i \in \{1, \dots, f\}$, denoting the multi-view input, output, and conditional frames' features as $X^{1:n,i}$, $X^{1:n,i}_{out}$, and $X^{1:n,1}$, we have:

$$X_{out}^{1:n,i} = \text{Attention}(X^{1:n,i} W_Q, X^{1:n,i} W_K, X^{1:n,i} W_V) W_O + \text{Attention}(X^{1:n,i} W_Q', X^{1:n,1} W_K, X^{1:n,1} W_V) W_O', \quad (2)$$

where W_Q, W_K, W_V and W_O are projection matrices in original self-attention layer, while W_Q' and W_O' are those in the newly added layer. We find this simple cross-attention operator can effectively improve the object's appearance consistency in the generated video. After that, as shown in the spatiotemporal block in Fig. 2, we employ two cross-attention layers to align the text prompt and preserve the object's identity, respectively. The left one is inherited from MVDream, while the right one is pre-trained in IP-Adapter [62].

Training Objectives. The training process of our multi-view video diffusion model is similar to Latent Diffusion Model [39]. Specifically, the sampled multi-view video data $q_0^{1:n,1:f}$ are first encoded into latent feature $z_0^{1:n,1:f}$ via encoder \mathcal{E} frame by frame and view by view. Then we add noise using the forward diffusion scheduler: $z_t^{1:n,2:f} = \sqrt{\alpha_t} z_0^{1:n,2:f} + \sqrt{1 - \alpha_t} \epsilon$, where α_t is a weighted parameter and ϵ is Gaussian noise. Note that, following I2V-Adapter, we keep the first frame, *i.e.*, the condition multi-view frames clean, and only add noise to the rest of the frames. During training, the proposed MV-VDM takes as input the clean latent code $z_0^{1:n,1}$, noisy latent code $z_0^{1:n,2:f}$, text prompt embedding y , and the camera parameters $\Sigma^{1:n}$, and outputs the noise strength, supervised by \mathcal{L}_2 loss. The training objective of our MV-VDM is calculated as:

$$\mathcal{L}_{\text{MV-VDM}} = \mathbb{E}_{\mathcal{E}(q_0), y, \epsilon \in \mathcal{N}(0, I), t} [\|\epsilon - \epsilon_\theta(z_0^{1:n,1}, z_t^{1:n,2:f}, t, y, \Sigma^{1:n})\|_2^2], \quad (3)$$

where θ denotes the diffusion model. It is important that we keep the entire multi-view 3D attention module frozen and only train the MV2V-Adapter layer and the spatiotemporal attention module to conserve GPU memory and accelerate training. Moreover, as the multi-view images of the first frame, $z_0^{1:n,1}$, serves as the condition images, we calculate the loss only for the latter $f - 1$ frames, *i.e.*, $z_0^{1:n,2:f}$.

3.2 Reconstruction and Distillation of 4DGS

Based on our 4D generation foundation model MV-VDM, we propose to animate any off-the-shelf 3D object. For efficiency, we take 3D Gaussian Splatting (3DGS) [25] as the static 3D object representation, and animate it by learning motion fields represented by Hex-planes, as in [55].

4D Motion Fields. As in 4D Gaussian Splatting (4DGS) [55], we represent the motion fields by Hex-planes [15, 8]. Denoting the static 3DGS as $\mathcal{G} = \{\mathcal{X}, \mathcal{C}, \alpha, r, s\}$, where \mathcal{X} , \mathcal{C} , α , r , and s represent the position, color, opacity, rotation, and scale, respectively. The motion module \mathcal{D} predicts changes in position, rotation, and scale for each Gaussian point in frame i by interpolating the Hex-planes R . The motion fields computation can be formulated as:

$$\mathcal{F} = \bigcup_l \prod_{\zeta} \text{interp}(R^{\zeta}, (\mathcal{X}, i)), \quad (4)$$

$$\Delta\mathcal{X} = \phi_{\mathcal{X}}(\mathcal{F}), \Delta r = \phi_r(\mathcal{F}), \Delta s = \phi_s(\mathcal{F}), \quad (5)$$

where l equals to the scales in Hex-plane, and $\text{interp}()$ denotes interpolating the Gaussian points on the specific plane ζ to obtain corresponding motion features. We have $\zeta \in \{(x, y), (x, z), (y, z), (x, t), (y, t), (z, t)\}$. Therefore, Gaussian \mathcal{G}' at time t is updated as follows:

$$\mathcal{G}' = \{\mathcal{X} + \Delta\mathcal{X}, \mathcal{C}, \alpha, r + \Delta r, s + \Delta s\}. \quad (6)$$

To better preserve the appearance of static 3D objects, we keep certain attributes, specifically opacity α and color \mathcal{C} , unchanged.

Coarse Motion Reconstruction. Based on the spatiotemporal consistent multi-view videos generated by MV-VDM, we first leverage a 4DGS reconstruction stage to directly reconstruct the coarse motions. Specifically, we use a simple but effective \mathcal{L}_2 loss as our \mathcal{L}_{rec} , which is calculated as:

$$\mathcal{L}_{\text{rec}} = \sum_{i=1}^n \sum_{j=1}^f \|\mathcal{C} - \widehat{\mathcal{C}}\|^2, \quad (7)$$

where \mathcal{C} and $\widehat{\mathcal{C}}$ denote the multi-view and multi-frame renderings and the corresponding ground truth. As verified in Fig. 3, this reconstruction stage can already learn high-quality coarse motions by leveraging the generated multi-view videos of MV-VDM.

4D-SDS Optimization. To better model the fine-level motions, we introduce a 4D-SDS optimization stage to distill the knowledge of our multi-view video diffusion model. The 4D-SDS loss $\mathcal{L}_{4\text{D-SDS}}$ is a variant of \mathbf{z}_0 -reconstruction SDS loss and can be formulated as:

$$\mathcal{L}_{4\text{D-SDS}}(\mathcal{G}, \mathcal{D}, z = \mathcal{E}(g(\mathcal{D}(\mathcal{G})))) = \mathbb{E}_{t, \Sigma, \epsilon} [\|z - \hat{z}_0\|_2^2], \quad \hat{z}_0 = \frac{z_t - \sigma_t \epsilon_{\theta}}{\alpha_t}, \quad (8)$$

where z and z_0 are latent feature of the rendered image and the estimation of clean latent feature from current noise prediction ϵ_{θ} , respectively, g represents the rendering function. α_t and σ_t are the signal and noise scale controlled by the noise scheduler, respectively.

Training Objectives. In addition to \mathcal{L}_{rec} and $\mathcal{L}_{4\text{D-SDS}}$, we introduce a variant of As-Rigid-As-Possible (ARAP) loss [46] to facilitate the rigid movement learning as well as the maintenance of the high-quality appearance of the static object. The ARAP loss $\mathcal{L}_{\text{arap}}$ in our work is defined as:

$$\mathcal{L}_{\text{arap}}(p_j) = \sum_{i=2}^f \sum_{k \in \mathcal{N}_{c_i}} w_{j,k} \|(p_j^i - p_k^i) - R_j((p_j^1 - p_k^1))\|^2, \quad (9)$$

where \hat{R}_j is estimated from a rigid transformation using Singular Value Decomposition (SVD) according to [46]:

$$\hat{R}_j = \underset{R \in \text{SO}(3)}{\text{argmin}} \sum_{k \in \mathcal{N}_{c_i}} w_{j,k} \|(p_j^i - p_k^i) - \hat{R}_j((p_j^1 - p_k^1))\|^2. \quad (10)$$

\mathcal{N}_{c_j} denotes the set of points within a fixed radius of p_j , and $w_{j,k} = \exp(-\frac{d_{jk}}{d})$ where d_{jk} is the distance between center of p_j and p_k , measuring the impact of p_k on p_j . This loss encourages the generated dynamic object to be locally rigid, and it enhances the learning with rigid movement.

In summary, the training objectives for animating off-the-shelf 3DGS object is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{rec}} + \lambda_2 \mathcal{L}_{4\text{D-SDS}} + \lambda_3 \mathcal{L}_{\text{arap}}, \quad (11)$$

where λ_1 , λ_2 , and λ_3 are weighted parameters.

3.3 Extension to Mesh Animation

To directly utilize high-quality mesh generated from commercial 3D generation tools and crafted by human experts, we extent our framework to mesh animation, producing animated mesh compatible with standard 3D rendering pipelines.

We initialize the 3DGS representation of the given object by vertices and triangles of the static mesh. Specifically, the color is determined by vertex color and we average the connected edges for the scale. Opacity and rotation are set to fully visible and zero rotation quaternion, respectively. The coarse 3DGS is animated following the motion reconstruction steps as described in the above sections. We utilize the per-vertex Gaussian trajectory to deform the static mesh in a straightforward way without skeleton rigging, control point selection or complicated deformation algorithms. As shown in Fig. 6 and our project page, the results are surprisingly good despite the simplicity of the solution.

4 Experiment

4.1 Setup

Training Dataset. To train our MV-VDM, we build a large-scale multi-view video dataset, MV-Video. Concretely, we render multi-view videos of **37,857** animated 3D models collected from Sketchfab [1]. Each model has **2.2** animations on average, resulting in **83,716** animations in total. Each animation is 2 seconds long at 24 fps. *Note that animated models that are not allowed to be used to generate AI programs are filtered.* The statistical information of our MV-Video dataset is reported in Table 1. For more details about the rendering settings and data examples, please refer to our Appendix (D). We will release this dataset to further advance the field of 4D generative research.

Table 1: Statistical information for our multi-view video (MV-Video) dataset.

Model ID	Animations	Avg. Animations per ID	Max Animations per ID	Multi-view Videos
37,857	83,716	2.2	6.0	1,339,456

Implementation Details. We sample 8 frames evenly for each animation to train our MV-VDM. We use the Adam optimizer with a learning rate of $4e - 4$ and a weight decay 0.01, and train the model for 20 epochs with a batch size of 2048. When inference, we set the sampling step to 25 and adopt freeinit [56] to get stable results when animating 3D objects. As for 4D generation, the resolution and feature dimension of the Hex-planes are set to [100, 100, 8] and 16, respectively. We perform coarse motion reconstruction for the first 1000 iterations with a batch size of 32 (4 views, 8 frames), and then add 4D-SDS optimization for another 400 iterations. Learning rate is 0.0015 initially and decreases linearly to 0.0005 at the end of reconstruction stage. λ_1 , λ_2 and λ_3 in Eq. 11 are set to 10.0, 0.01 and 0.5, respectively. It costs 3 days to train our MV-VDM on 32 80G A800 GPUs, and the optimization for 4D generation takes around 30 minutes on a single A800 GPU per object.

Evaluation Dataset. For the evaluation of MV-VDM, we render multi-view images from 128 static 3D objects and then generate multi-view videos conditioned on them. Following the evaluation setting of VBench [23], we use four different random seeds for each object and report the average results. For the evaluation of 4D generation, we generate 25 objects across various categories using the large 3DGS reconstruction model GRM [61]. Input images for GRM and animation prompts used in this work are provided in our Appendix (E.1).

Evaluation Metrics. We adopt the evaluation protocol proposed in VBench [23], which is a popular video generation benchmark consisting of both T2V and I2V evaluation tools. The I2V evaluation protocol contains 9 evaluation metrics, and we choose 4 for our evaluation, *i.e.*, I2V Subject, Motion Smoothness, Dynamic Degree, and Aesthetic Quality, measuring the consistency with the given image, the motion smoothness, the motion degree, and the appearance quality, respectively. We abbreviate them as **I2V**, **M. Sm.**, **Dy. Deg.** and **Aest. Q.** in the experiment section. Values of all metrics are the higher, the better, except for Dynamic Degree, since we observe that sometimes completely failed results present extremely high dynamic degree. For more details about the introduction and calculation of the evaluation metrics, please refer to our Appendix (E.2).

Comparison Methods. We compare our work with 4Dfy [5] and DreamGaussian4D (DG4D) [38] on the task of animating any given 3D object using their official implementations. They represent the state-of-the-art in 4D generation methods, starting by generating a static 3D object using 3D-SDS in

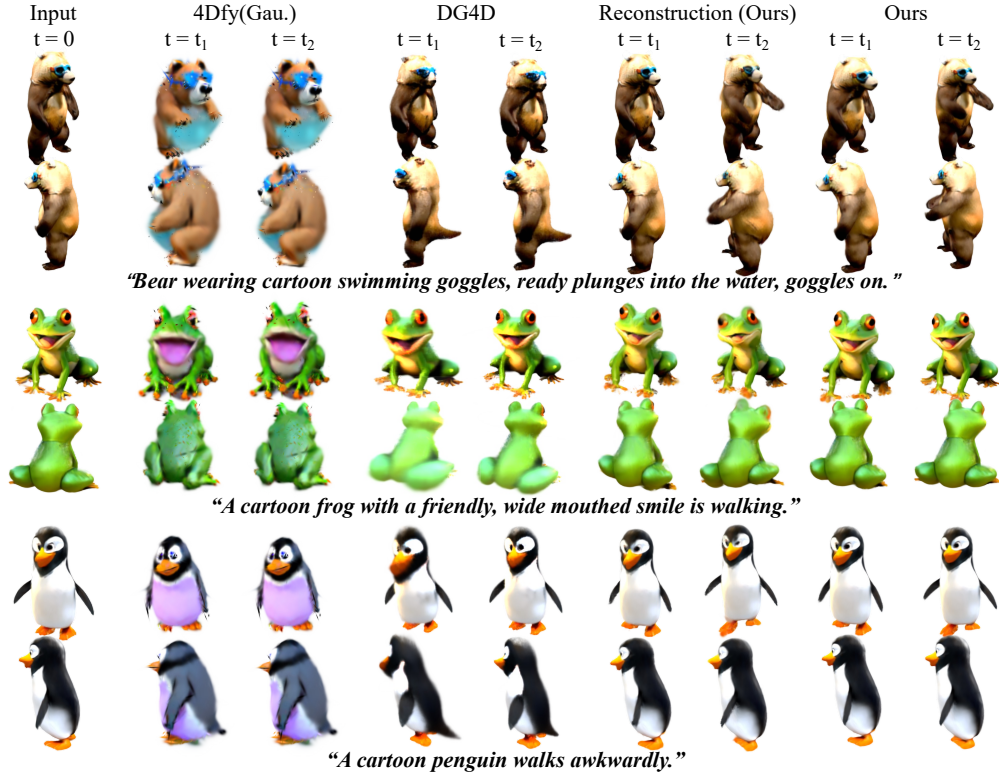


Figure 3: Qualitative comparison with state-of-the-art methods.

the initial stage, and subsequently animating it via video SDS and single-view video reconstruction in later stages. For 4D representation, 4Dfy and DG4D adopt dynamic NeRF [34, 26] and 4DGS [55], respectively. For a fair comparison, we replace the dynamic NeRF in 4Dfy with 4DGS used in both our work and DG4D, and also apply ARAP loss for motion regularization. For DG4D, we keep the 4DGS representation and motion regularizations in their work unchanged.

4.2 Comparison with State-of-the-Art

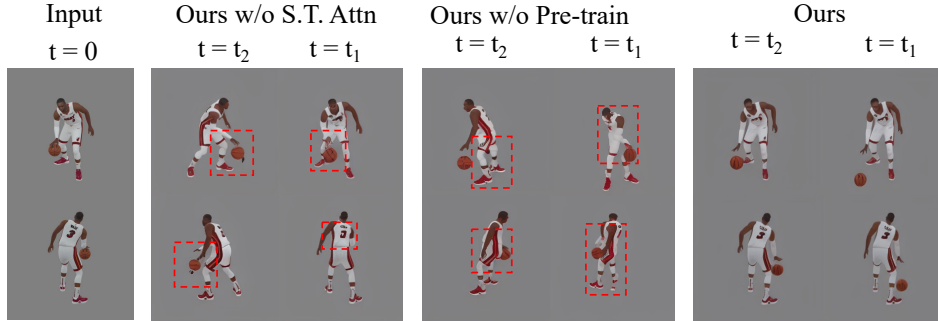
In this section, we perform comprehensive comparisons with previous works, including quantitative and qualitative comparisons and user studies.

Table 2: Quantitative comparisons with state-of-the-art methods.
(a) Comparison on video generation metrics. (b) Comparison via user study.

	I2V \uparrow	M. Sm. \uparrow	Dy. Deg.	Aest. Q. \uparrow	Align. Text	Align. 3D.	Mot.	Appr.
4Dfy (Gau.) [5]	0.783	0.996	0.0	0.497	2.028	1.608	1.534	1.84
DG4D [38]	0.898	0.986	0.477	0.529	2.824	3.52	2.284	3.108
Ours	0.982	0.991	0.597	0.581	4.386	4.734	4.288	4.528

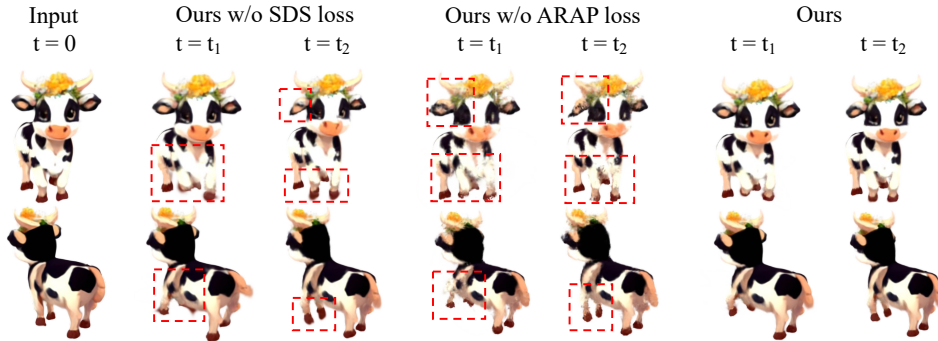
Quantitative Comparison. As shown in Tab. 2a, our method significantly outperforms 4Dfy and DG4D in terms of **I2V**, **Dy. Deg.**, and **Aest. Q.**. This indicates our generation results have good alignment with the given static 3D object (I2V Subject), dynamic motion (Dynamic Degree), and superior appearance (Aesthetic Quality). For Motion Smoothness, we slightly lag behind 4Dfy, since 4Dfy always generates nearly static results, as illustrated by the 0.0 Dynamic Degree in the first row of Tab 2a. Generally, our method is able to animate 3D object with smooth and dynamic motion, at almost no cost of sacrificing their high-quality appearance, facilitating customized and high-quality dynamic 3D object creation.

Qualitative Comparison. As shown in Fig. 3, it is obvious that 4Dfy’s results are blurred and deviate much from the given 3D object, owing to the use of text-conditioned diffusion models to optimize the motion and appearance. Additionally, its generated objects are almost static. This is because at the beginning of the training process, the noisy rendered image sequence, *i.e.*, the input to the T2V model, has no temporal changes, which misleads the video diffusion model to generate almost static supervisions, as illustrated in Fig 1. For DG4D, its results align well with the given 3D object in front



“A basketball player is playing basketball.”

Figure 4: Ablation for multi-view video diffusion.



“Cow in cartoon style lifts its front hooves.”

Figure 5: Ablation for 3D object animation.

view, *i.e.*, the view used for generating guided video. However, it doesn’t align with the object in the novel view, as indicated by the *tail in Bear and Penguin*, *distorted goggles in bear*, and *blurred back and side views* in Fig 3. This is because it adopts Zero123 to optimize the novel views. Zero123 only conditions on the front view, leading to NVS optimization favoring pre-trained data distributions, which could lead to potential appearance degradation. More importantly, DG4D fails when the object in the guided video is assigned with movements toward the camera. For example, the frog is moving forth and back in the guided video, however, DG4D interprets it as enlargement and reduction of the object. The same thing comes to the penguin which nods towards the camera and leans forward. This misinterpretation usually results in blurry effect and strange appearance.

In contrast, our method, leveraging the spatiotemporal consistent multi-view prior, manages to deal with motion towards the camera, as demonstrated by the bear’s raised front paw (our model takes the front view and its orthogonal views as the condition view, not depicted in the image). Besides, we successfully maintain the high quality appearance of the given 3D object when generating natural motion. Please refer to the videos in our *supplementary material* for a more intuitive comparison.

User Study. We conduct a user study among 20 people on the 25 dynamic objects and report the averaged results in Tab. 2b. The participants are asked to score the generated dynamic objects from 1 to 5, according to the alignment with the given text (**Align. Text**) and static object (**Align. 3D**), motion quality (**Mot.**), and appearance quality (**Appr.**). The user study proves the superiority of our method. Please refer to the Appendix (E.3) for more details.

4.3 Ablation

Table 3: Ablation Studies

	(a) Ablation of Multi-View Diffusion				(b) Ablation of 4D Generation				
	I2V \uparrow	M. Sm. \uparrow	Dy. Deg.	Aest. Q. \uparrow	I2V \uparrow	M. Sm. \uparrow	Dy. Deg.	Aest. Q. \uparrow	
w/o S.T. Att.	0.915	0.980	0.958	0.531	w/o SDS loss	0.978	0.990	0.657	0.572
w/o Pre-train	0.910	0.981	0.944	0.531	w/o ARAP loss	0.970	0.990	0.573	0.557
Ours	0.935	0.988	0.710	0.532	Ours	0.983	0.997	0.597	0.581

Multi-view Video Diffusion. In Tab 3a, we validate the effectiveness of the proposed SpatioTemporal Attention (short as S.T. Att.) and the pre-trained weight from video diffusion model (short as Pre-train). We replace the proposed spatiotemporal block with temporal block from AnimateDiff, and

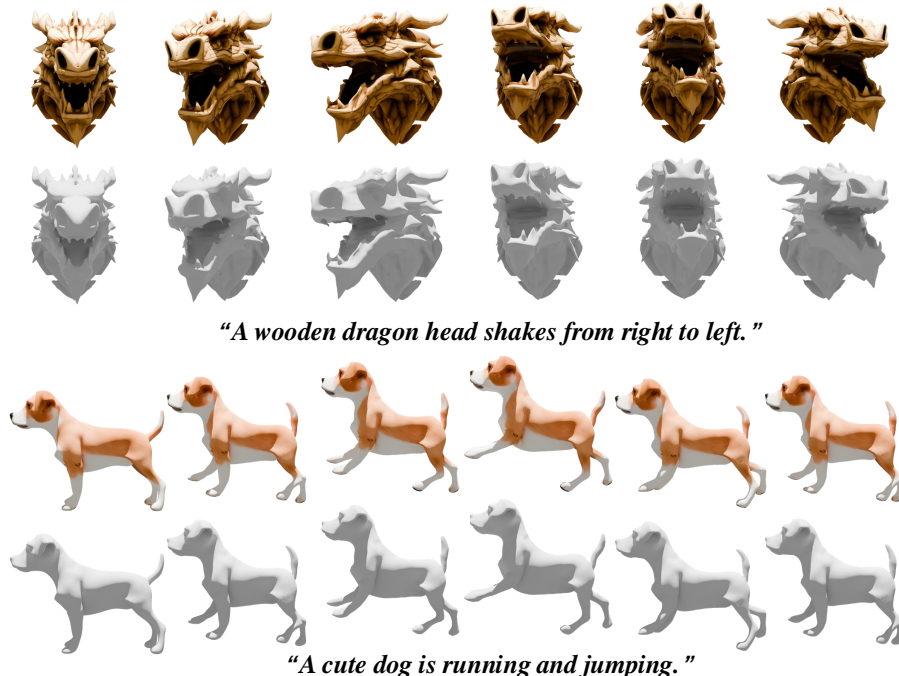


Figure 6: Visualizations of mesh animation. We present RGB and textureless renderings of two mesh animation results. Best viewed by zooming in.

this leads to performance drop in I2V Subject, Motion Smoothness and Aesthetic Quality. Dynamic Degree seems to be enhanced, but that is caused by the increase of unstable failure cases. The same tendency could be observed in experiments w/o pre-trained video diffusion weight. Therefore, we think both designs are necessary for generating multi-view videos consistent with the given multi-view images and with high-quality appearance and motion. Qualitative ablations in Fig 4 further demonstrate this point.

4D Object Optimization. The ablations for 4D object optimization are shown in Tab. 3b and Fig. 5. The quantitative results in Tab. 3b indicate both SDS and ARAP losses improve the alignment with the 3D object (I2V Subject), Motion Smoothness, and Aesthetic Quality. However, the Dynamic Degree decreases. We suppose the decrease in dynamic degree is mainly caused by the removal of floaters and blurry effects, which are also taken into account of dynamic degree, as shown in Fig. 5. The two losses might slightly decrease the motion amplitude, but generally, we think the overall performance is improved when applying them.

4.4 Mesh Animation

We provide mesh animation results in Fig. 6. Static meshes are generated by commercial 3D generation tools. For more results, please visit our project page.

5 Conclusion

In this work, we present Animate3D, a novel framework for animating any off-the-self 3D object. Animate3D disentangles the 4D object generation into a foundational 4D generation model, MV-VDM, and a joint 4DGS optimization pipeline based on MV-VDM. MV-VDM is the first 4D foundation model, which can generate spatiotemporal consistent multi-view videos conditioned on multi-view renderings of a static 3D object. To train MV-VDM, we present the largest multi-view video (4D) dataset, MV-Video, containing about 84K animations with over 1.3M multi-view videos. Based on MV-VDM, we propose an effective pipeline to animate any static 3D object by jointly optimizing the 4DGS via both reconstruction and 4D-SDS. Animate3D is a highly practical solution for downstream 4D applications since it can animate any generated or reconstructed 3D objects. Data, codes, and pre-trained weights will be released to facilitate the research in 4D generation.

6 Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported in part by the Natural Science Foundation of China (Grant No. 62192782, U22B2056, 62422317), the Beijing Natural Science Foundation (Grant No. L223003, JQ22014), the Natural Science Foundation of China (62036011, U2033210, 62102417, 62222206, 62172413), the Project of Beijing Science and technology Committee (Project No. Z231100005923046). Jin Gao was also supported in part by the Youth Innovation Promotion Association, CAS. This work was also supported by Damo Academy through Damo Academy Research Intern Program.

References

- [1] Sketchfab. <https://sketchfab.com/>. Accessed: 2024-5-21.
- [2] Zeroscope text-to-video model. https://huggingface.co/cerspense/zeroscope_v2_576w. Accessed: 2023-10-31.
- [3] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. Minigt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens. *arXiv preprint arXiv:2404.03413*, 2024.
- [4] Sherwin Bahmani, Xian Liu, Yifan Wang, Ivan Skorokhodov, Victor Rong, Ziwei Liu, Xihui Liu, Jeong Joon Park, Sergey Tulyakov, Gordon Wetzstein, et al. Tc4d: Trajectory-conditioned text-to-4d generation. *arXiv preprint arXiv:2403.17920*, 2024.
- [5] 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. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [6] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.
- [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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.
- [8] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023.
- [9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [10] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.
- [11] Wen-Hsuan Chu, Lei Ke, and Katerina Fragkiadaki. Dreamscene4d: Dynamic multi-object scene generation from monocular videos. *arXiv preprint arXiv:2405.02280*, 2024.
- [12] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024.
- [13] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.
- [14] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022.

- [15] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023.
- [16] Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024.
- [17] Xun Guo, Mingwu Zheng, Liang Hou, Yuan Gao, Yufan Deng, Chongyang Ma, Weiming Hu, Zhengjun Zha, Haibin Huang, Pengfei Wan, et al. I2v-adapter: A general image-to-video adapter for video diffusion models. *arXiv preprint arXiv:2312.16693*, 2023.
- [18] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animated-iff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.
- [19] 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.
- [20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.
- [21] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.
- [22] 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.
- [23] Ziqi Huang, Yanan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [24] Yanqin Jiang, Li Zhang, Jin Gao, Weiming Hu, and Yao Yao. Consistent4d: Consistent 360° dynamic object generation from monocular video. In *The Twelfth International Conference on Learning Representations*, 2024.
- [25] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.
- [26] Sixu Li, Chaojian Li, Wenbo Zhu, Boyang Yu, Yang Zhao, Cheng Wan, Haoran You, Huihong Shi, and Yingyan Lin. Instant-3d: Instant neural radiance field training towards on-device ar/vr 3d reconstruction. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–13, 2023.
- [27] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [28] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.
- [29] 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. *arXiv preprint arXiv:2312.13763*, 2023.
- [30] Minghua Liu, Chao Xu, Haiyan Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36, 2024.
- [31] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023.

- [32] 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.
- [33] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- [34] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.
- [35] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. *arXiv preprint arXiv:2311.16918*, 2023.
- [36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [37] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021.
- [38] Jiawei Ren, Liang Pan, Jiayang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023.
- [39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [40] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model, 2023.
- [41] 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.
- [42] 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.
- [43] 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. *arXiv preprint arXiv:2301.11280*, 2023.
- [44] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
- [45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [46] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116. Citeseer, 2007.
- [47] Jiayang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.
- [48] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.
- [49] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. *arXiv preprint arXiv:2403.12008*, 2024.
- [50] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023.

- [51] 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.
- [52] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.
- [53] 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. *Advances in Neural Information Processing Systems*, 36, 2024.
- [54] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrn: Large reconstruction model for high-quality mesh. *arXiv preprint arXiv:2404.12385*, 2024.
- [55] 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. *arXiv preprint arXiv:2310.08528*, 2023.
- [56] Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. Freeinit: Bridging initialization gap in video diffusion models. *arXiv preprint arXiv:2312.07537*, 2023.
- [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.
- [58] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023.
- [59] Dejia Xu, Hanwen Liang, Neel P Bhatt, Hezhen Hu, Hanxue Liang, Konstantinos N Plataniotis, and Zhangyang Wang. Comp4d: Llm-guided compositional 4d scene generation. *arXiv preprint arXiv:2403.16993*, 2024.
- [60] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.
- [61] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024.
- [62] Hu Ye, Jun Zhang, Sibao 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.
- [63] 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.
- [64] Jiwen Yu, Xiaodong Cun, Chenyang Qi, Yong Zhang, Xintao Wang, Ying Shan, and Jian Zhang. Animatezero: Video diffusion models are zero-shot image animators. *arXiv preprint arXiv:2312.03793*, 2023.
- [65] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Tianyou Liang, Guanying Chen, and Xiaoguang Han. Mvimagnet: A large-scale dataset of multi-view images. In *CVPR*, 2023.
- [66] 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.
- [67] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *arXiv preprint arXiv:2404.19702*, 2024.
- [68] 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.
- [69] 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. *arXiv preprint arXiv:2311.16854*, 2023.
- [70] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.



Figure 7: Qualitative comparison with state-of-the-art methods on reconstructed 3D objects.

A More Visualizations

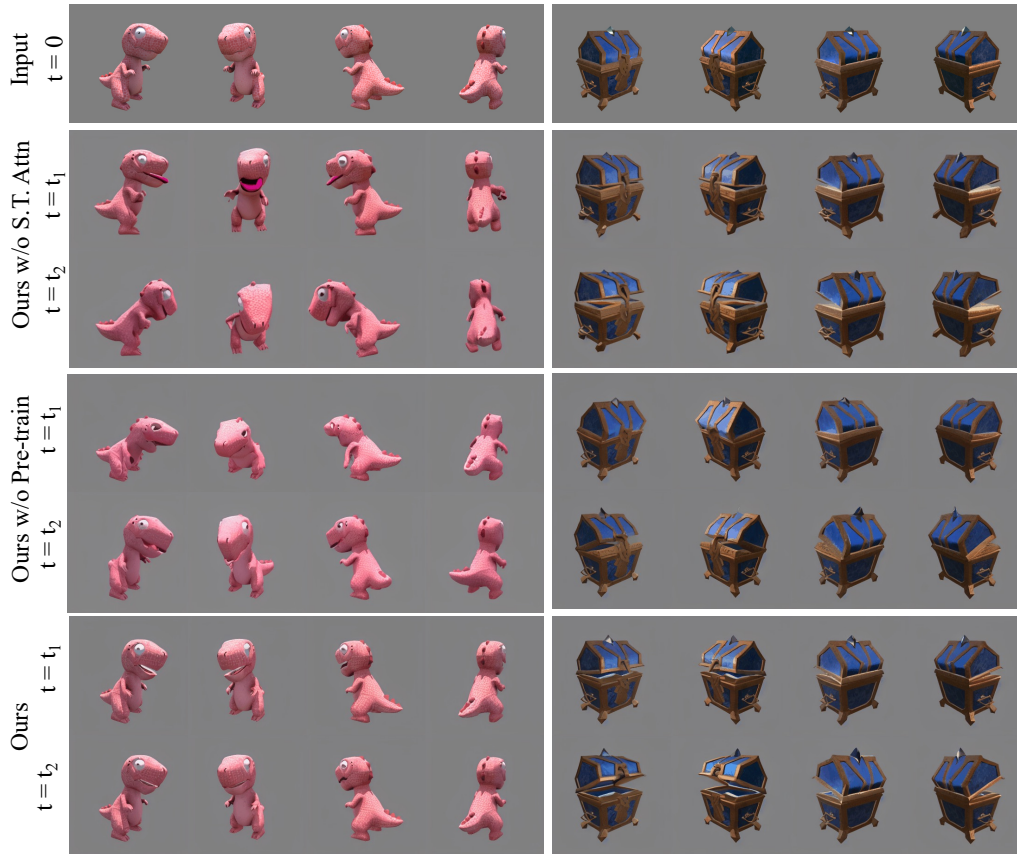
We provide more qualitative results as below. Please refer to the videos attached in our supplementary materials for more intuitive visualizations.

Comparison. In the main paper, we only present the animation results for 3D objects generated by large reconstruction models. But our method is able to animate reconstructed objects as well as generate 3D models from text/image/video, and shows a great advantage over other methods as shown in Fig. 7. In Fig. 7, 4Dfy produces results inconsistent with the given 3D object, and they are almost static. Distortions and blurry effects could be observed in results generated by DreamGaussian4D, and it sometimes deviates from the given object in novel views. In contrast, our approach can generate results not only spatially and temporally coherent but also consistent with the input object.

Ablation. We provide more qualitative ablations of MV-VDM and 4D generation in Fig. 8.

B Discussion with respect to Previous 4D Generation Methods

Previous two-stage 4D generation works attempted to disentangle motion learning from appearance learning by adopting different types of supervision signal, i.e., video diffusion/monocular video for



“A pink dinosaur is waving its little hands”

“A blue treasure chest is being opened.”

(a) More ablation studies of MV-VDM.



(b) More ablation studies of 4D generation.

Figure 8: Ablations of MV-VDM and 4D optimization.

motion and image/3D diffusion for appearance. However, the motion and appearance supervisions adopted in their work are not orthogonal, and sometimes have negative effect on each other.

For example, it is commonly agreed that video-diffusion-SDS usually brings unappealing visual effect to the appearance of the object [68, 69, 29]. Meanwhile, the appearance supervision signal prevents 4D object from updating along the direction that follows the exact score function of the video diffusion model, leading to less natural motion. Small motion amplitude in [5, 69] and shaky appearance in [29] could partly support this point. As for monocular-video-guided motion learning, previous work [38, 63, 66] rely on 3D diffusion model (Zero123 [31]) to supervise both motion and appearance in novel view. Since Zero123-SDS is applied per-frame, temporal consistency in novel view cannot be guaranteed. Moreover, monocular video doesn't provides information about depth/distance, so moving closer to or farther away from the camera can be perceived as the magnification or reduction of the object, resulting in appearance distortion.

In contrast, our method takes the unified supervision signal from MV-VDM for motion learning and appearance preservation. Our motion and appearance supervision signal inherently don't conflict with each other, since MV-VDM is conditioned on multi-view attributes of the 3D object to generate multi-view videos. Besides, multi-view motion supervision in our work enables more natural motion generation when compared with single-view motion supervision in other works. Thus, we achieve superior performance in terms of both motion generation and appearance preservation in the task of animating any off-the-shelf 3D object.

To provide a comprehensive understanding of the difference between our method and previous work, we conduct comparisons with more open-sourced previous works using their official implementations, and summarize the results in Tab. 4.

Table 4: Comparison on 4D Generation. Note that TC4D [4] takes pre-defined object trajectory as the input.

	I2V \uparrow	M. Sm. \uparrow	Dy. Deg.	Aest. Q. \uparrow	CLIP-I \uparrow
4Dfy [5] (4DGS)	0.783	0.996	0.0	0.497	0.786
4Dfy [5] (NeRF)	0.817	0.990	0.010	0.549	0.834
Animate124 [68]	0.845	0.986	0.313	0.563	0.845
4DGen [63]	0.833	0.994	0.187	0.453	0.776
TC4D [4]	0.856	0.992	0.830	0.565	0.859
Dream-in-4D [69]	0.938	0.994	0.0	0.551	0.895
DG4D [38]	0.898	0.986	0.477	0.529	0.860
Ours (8-frame)	0.982	0.991	0.597	0.581	0.946
Ours (16-frame)	0.983	0.991	0.750	0.572	0.937

C Limitations

Despite the promising performance in generating spatiotemporal consistent 4D objects, our method still has a few limitations. First, it takes a relatively long time (about 30 minutes) to animate an existing 3D object. Second, there is a trade-off between temporal coherence and motion amplitude in the multi-view videos generated from the proposed MV-VDM. Specifically, the larger the motion amplitude, the higher the risk of temporal incoherence. Third, our model sometimes fails to animate realistic scenes due to the domain gap between synthetic training data and real-world test data. At last, current evaluation metrics in 4D generation are not sufficient, as they mainly rely on video generation metrics and user studies. Designing more suitable metrics for 4D generation will be an important future work.

D More Details of MV-Video Dataset

D.1 Rendering Details.

For the rendering settings, we first centralize the model according to the bounding box of the object in the first frame. Then, we adjust the camera distance to make sure the object stays in the scope of view during the animation process. Sixteen views are evenly sampled in terms of azimuth, starting from

values randomly selected between -11.25° and 11.25° . The elevation angle is randomly sampled within the range of 0° to 30° . To stabilize training, we manually filter out objects that we identify as challenging to learn due to factors such as large movements, complex environmental interactions, high speed, or sudden changes in appearance.

D.2 Data Examples.

As shown in Fig. 12 and Fig. 13, we showcase more examples of our multi-view video dataset (MV-Video).

Furthermore, as shown in Fig. 9, we extracted all nouns from the text captions of our MV-Video dataset, which are generated by minigt4-video [3], and plotted a word cloud of the Top-1000 nouns. We can see that our MV-Video dataset contains diverse categories of animated 3D objects, including humans, characters, animals, plants, mechanical structures, and *ect.*

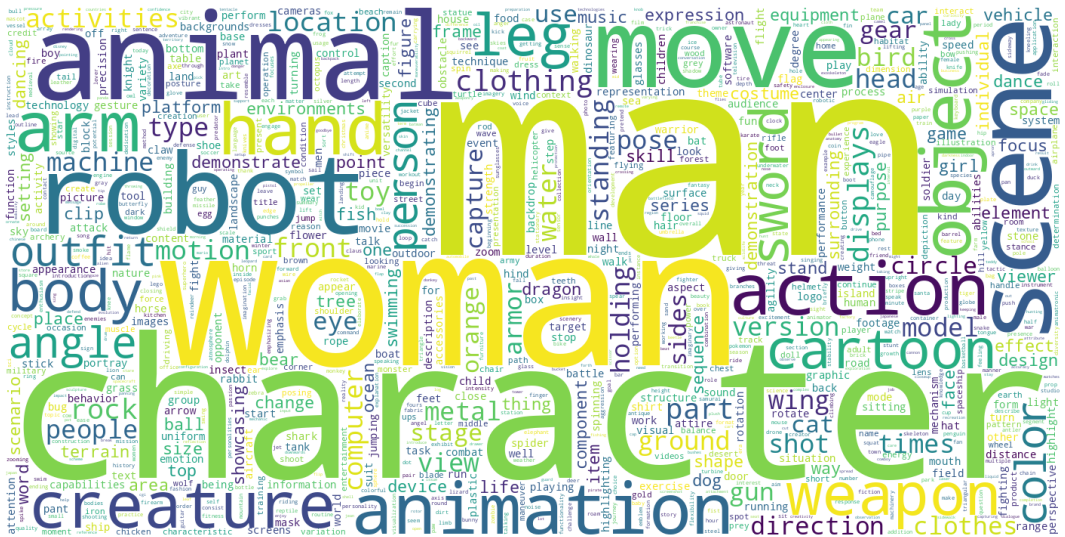


Figure 9: Illustration of the word cloud of the top 1000 nouns extracted from the text captions of our MV-Video dataset.

E 4D Generation Evaluation

E.1 Evaluation Dataset

In Fig. 10, we provide the input images for image-to-3D generation and corresponding text prompts for 4D animation used in Sec. 4.2.

E.2 Evaluation Metrics

VBench [23] provides six evaluation metrics for I2V evaluation³, *i.e.*, I2V Subject, I2V Background, Camera Motion, Subject Consistency, Background Consistency, Motion Smoothness, Dynamic Degree, Aesthetic Quality and Imaging Quality. Since our generated results have no background, and the evaluation cameras are fixed, metrics related to background and camera motion are not used. The metric Imaging Quality, which is affected by ambient light, is also not used here. Subject Consistency is also omitted since its calculation process is similar to I2V Subject, except for the choice of reference frame. It takes the first generated frame, instead of the input image used in I2V Subject, as the reference frame for similarity score calculation, which is not suitable for our task of animating 3D objects. We briefly introduce the evaluation metrics used in our work as below:

³https://github.com/Vchitect/VBench/tree/master/vbench2_beta_i2v

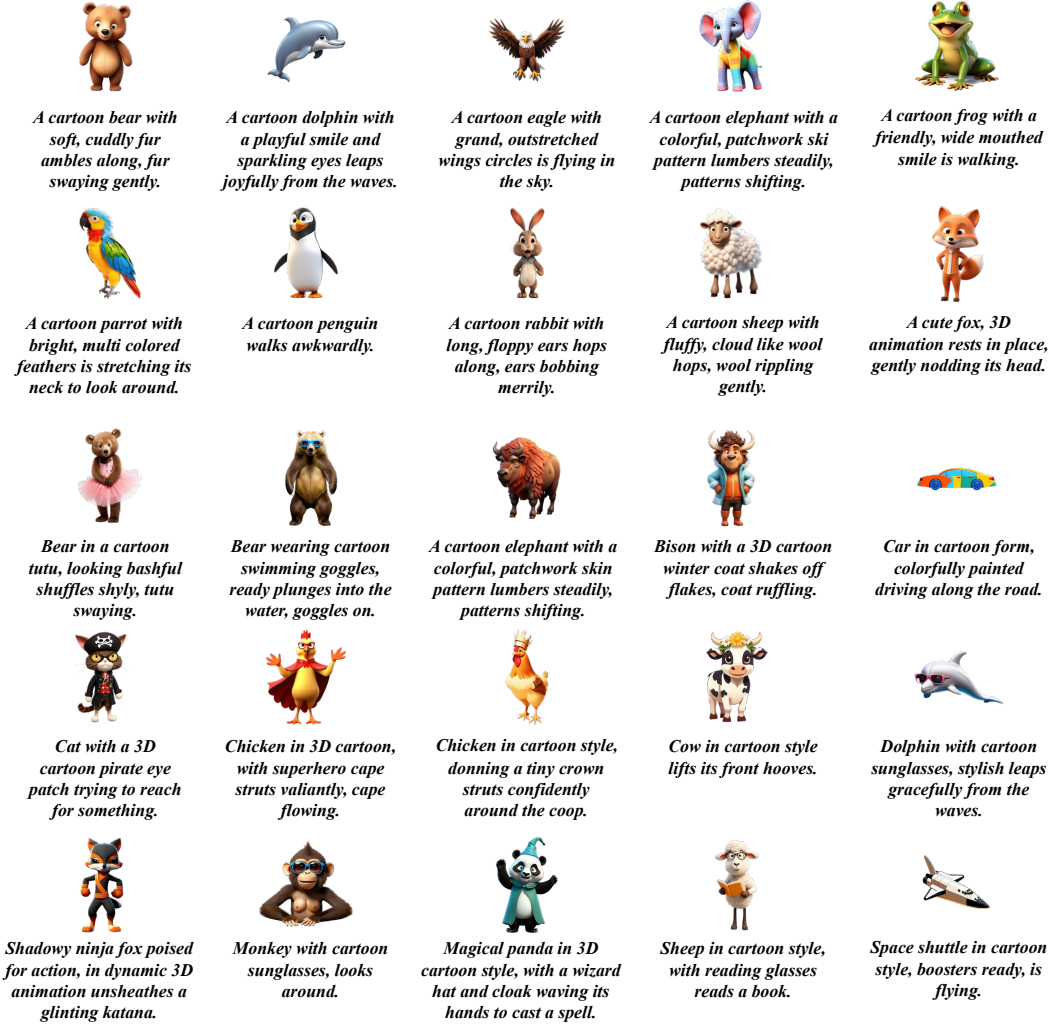


Figure 10: Illustration of the input images for image-to-3D generation and corresponding prompts for 4D animation.

I2V Subject assess whether the appearance of the object in the generated video remains consistent with that in the input image. To this end, DINO [9] feature similarity across frames is calculated.

Motion Smoothness evaluates whether the motion in the generated video is smooth, and follows the physical law of the real world. The motion prior in the video frame interpolation model [27] is utilized for evaluation.

Dynamic Degree employs RAFT [48] to estimate the degree of dynamics in synthesized videos.

Aesthetic Quality is calculated by the LAION aesthetic predictor, which reflects the artistic and beauty value perceived by humans towards each frame.


E.3 User Study Template

As illustrated in Fig. 11, a picture of the user study page is depicted. The survey contains 25 dynamic objects, which are shown in Fig. 10. The participants are asked to score the generated 4D objects from 1 to 5, according to the alignment with the given static object and text prompt, appearance quality, and motion quality.

Every three constitute a group, representing the animation generations for the same object using three different methods (the order has been shuffled). Please compare the generated results within the same group and then provide scores.

1. Please rate the animated results for the static 3D model in conjunction with the video. (The first row features reference images of the 3D model, while the second row displays the generated animation results. The first and second columns represent two different viewpoints, respectively.)

The desired text description for the animation is: "Sheep in cartoon style, with reading glasses, reads a book."

	Score:	1	2	3	4	5
	Consistency with the static 3D model:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Consistency with the text prompt:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Aesthetic quality of the results:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Movement quality of the results:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Gif

2. ...

Figure 11: The layout of our user study.

F Border Impacts

This paper exploited 4D generation based on our proposed multi-view video diffusion model, which can generate spatiotemporal consistent multi-view videos. Because of the advanced generative capacity, our models may output misinformation or fake videos. Thus, we sincerely remind the users to pay attention to it. Note that our method only focuses on the technical aspect. All the code, dataset, and trained models will be released.



Figure 12: More examples of our MV-Video dataset.

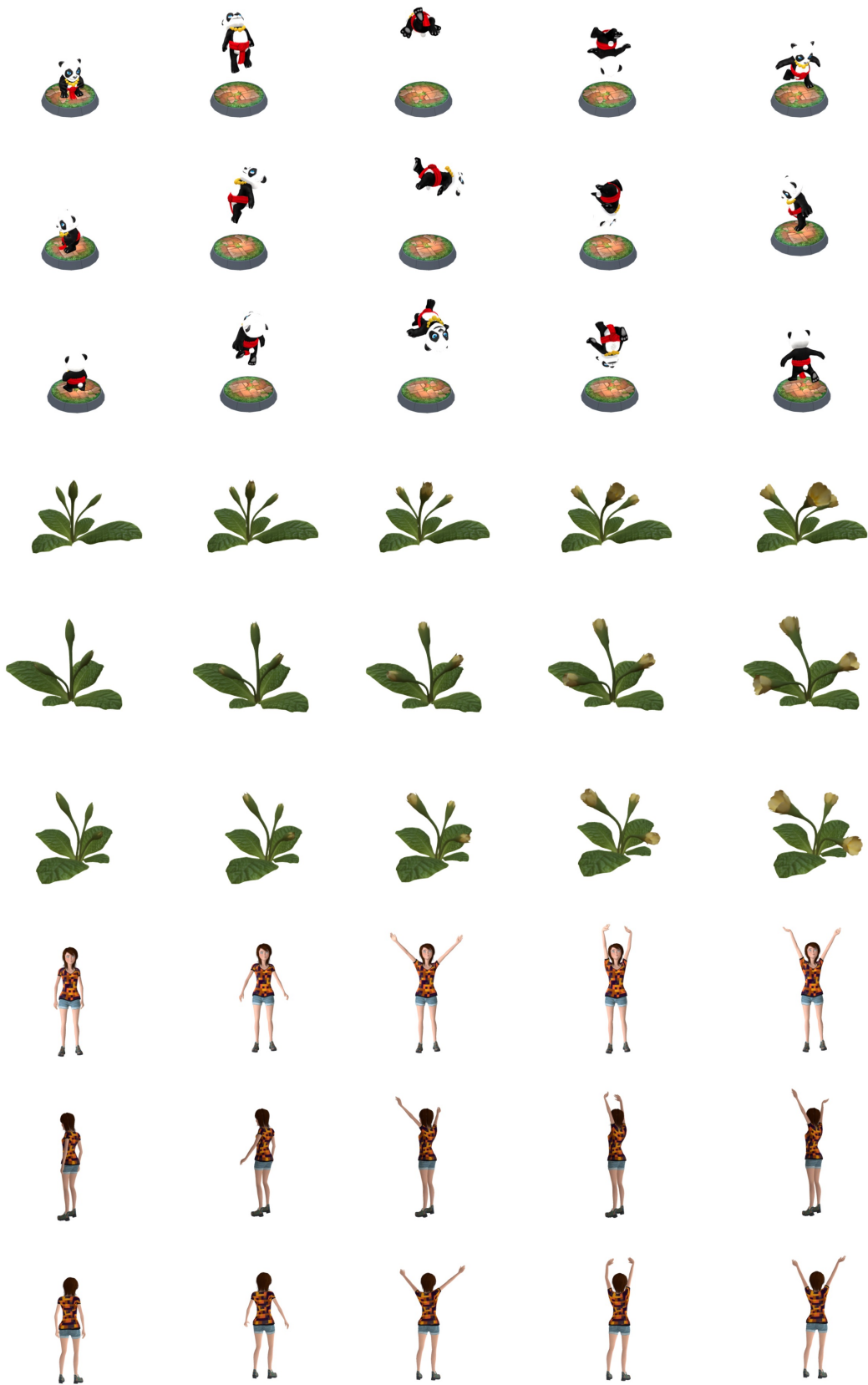


Figure 13: More examples of our MV-Video dataset.

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: our abstract clearly clarifies our contributions: 1) We propose Animate3D, the first 4D generation pipeline to animate any 3D objects. 2) We propose the first 4D generation foundation model, MV-VDM. 3) We present the largest 4D datasets, MV-Video.

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: we have included our limitation in Sec. C.

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?

Answer: [NA]

Justification: This paper did 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: our experiments are reproducible, and all codes, dataset, and pre-trained models will be open-released.

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

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: All data, codes, and pre-trained weights used in our experiments will be open-released.

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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: we include detailed implementations in Sec. 4.1. The full details can be seen in our codes, which will be open-released.

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: As a visual generative model, we provide sufficient visualization results instead.

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).
- 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 information on the computer resources in Sec. 4.1.

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 work conforms to the NeurIPS Code of Ethics.

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 provide the it in Sec. F.

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.
- 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 paper poses no such risks.

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 have cited related works and codes.

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.

- 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: the paper does not release new assets for now. Codes, datasets, and trained models will be open-released soon.

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: the 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: the 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.