

Articulated Kinematics Distillation from Video Diffusion Models

Xuan Li^{1,2,*} Qianli Ma² Tsung-Yi Lin² Yongxin Chen²
 Chenfanfu Jiang¹ Ming-Yu Liu² Donglai Xiang²
¹ UCLA, ² NVIDIA

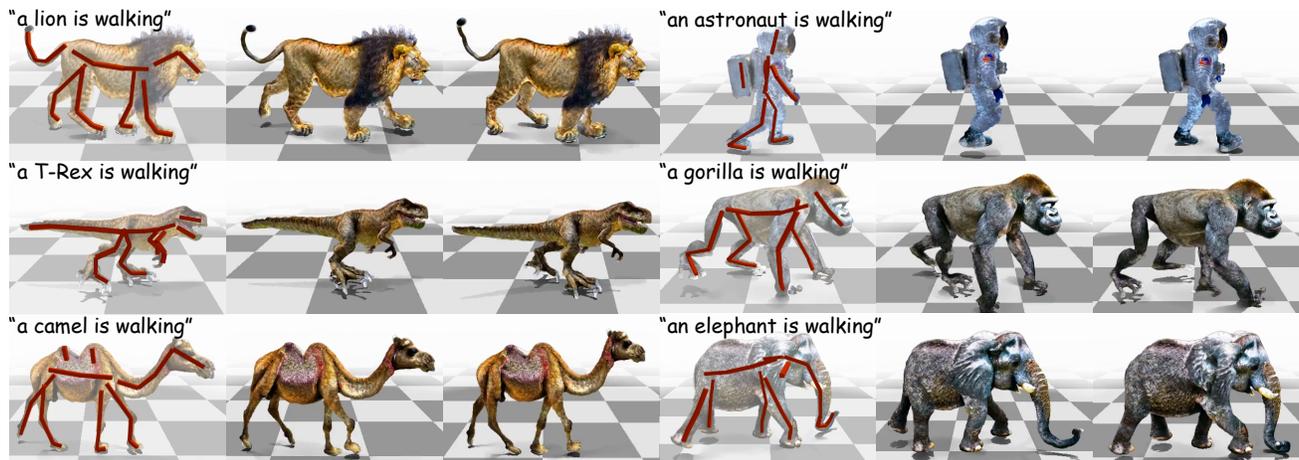


Figure 1. By incorporating articulation into static assets, AKD synthesizes realistic motions distilled from large video diffusion models.

Abstract

We present *Articulated Kinematics Distillation (AKD)*, a framework for generating high-fidelity character animations by merging the strengths of skeleton-based animation and modern generative models. AKD uses a skeleton-based representation for rigged 3D assets, drastically reducing the Degrees of Freedom (DoFs) by focusing on joint-level control, which allows for efficient, consistent motion synthesis. Through *Score Distillation Sampling (SDS)* with pre-trained video diffusion models, AKD distills complex, articulated motions while maintaining structural integrity, overcoming challenges faced by 4D neural deformation fields in preserving shape consistency. This approach is naturally compatible with physics-based simulation, ensuring physically plausible interactions. Experiments show that AKD achieves superior 3D consistency and motion quality compared with existing works on text-to-4D generation.

1. Introduction

In traditional 3D graphics, a skeleton-based character animation pipeline involves steps like shape modeling, rig-

ging, motion capture, motion retargeting, and editing. As a mature technology, such pipelines can achieve high realism and good controllability over the motion, but they typically require extensive manual work from digital artists, making the process time-consuming and thus hardly scalable. Recent advances in video generation models [6, 59] offer a promising avenue for streamlining the animation authoring process: with a text-to-video model, generating a sequence of character animation only requires a text prompt. However, existing video generation models still struggle to generate high-fidelity dynamics for real-world objects because of a lack of 3D information. Common issues include failing to preserve the 3D structure consistency (e.g. number of limbs of a character) during animation, or producing physically implausible articulated motion, such as foot-skating and ground penetration.

Recent works on text-to-4D generation [1, 2] leverage these video generation models to distill the learned dynamic motion into consistent 4D sequences. These frameworks commonly rely on neural deformation fields which predict displacements at each location in a pre-defined 3D volume to deform a 3D shape. Animation is thus a temporal sequence of such deformed shapes. While flexible, this approach introduces a large number of Degrees of Freedom

* Work done during an internship at NVIDIA.

(DoFs), making optimization challenging and often resulting in suboptimal quality. Is it possible to have the best of both worlds, where generative models provide extensive knowledge of diverse motions from internet-scale data, while skeleton-based 3D animation allows low-DoF control, permanence of articulated structures, and even physical grounding via simulation?

To answer this question, we introduce Articulated Kinematics Distillation (AKD), a motion synthesis system that bridges traditional character animation pipelines and generative motion synthesis. Given a rigged 3D asset, we distill articulated motion sequences from a pre-trained video diffusion model using Score Distillation Sampling (SDS). The skeleton-based representation simplifies the distillation process by limiting the number of DoFs to that of a few joints, in contrast to all query points in space-time as in the text-to-4D works [2]. It also offers an effective regularization of the deformation space, enabling the distillation to concentrate on overall motion styles without worrying about maintaining reasonable local structures. More importantly, the skeleton-based formulation is naturally compatible with physics-based simulation, allowing the generated motion to be grounded by physics-based motion tracking to ensure physical plausibility.

Experiments verify the effectiveness of our design: compared to previous text-to-4D methods, our framework produces results with better 3D shape consistency and more expressive motions. We summarize our contributions:

- We introduce a novel text-driven motion synthesis framework for static 3D assets, combining articulated rigging systems and large video diffusion models.
- We demonstrate that incorporating non-uniform ground renderings enhances the video model’s adherence to basic physics between the character and the ground.
- Extensive experiments show that our generated motions exhibit higher quality than the state-of-art methods that can synthesize long-trajectory motions.
- Our generated motion can be used in physics-based motion tracking with differentiable physics to further boost its physical realism.

2. Related Work

Deformable Gaussian Splatting In recent years, different types of 3D representations have been introduced to facilitate the reconstruction and generation of 3D/4D scenes, such as neural fields [32], iNGP [34], and 3D Gaussian Splatting [21]. Among these representations, 3D Gaussian Splatting is particularly suitable for representing dynamic scenes due to its explicit nature [52] compared to the NeRF based on neural implicit fields [35, 38], whose deformations are achieved by bending rendering rays [9, 37]. This advantage of 3DGS has sparked a lot of works on 4D reconstruction and modeling from multi-view input, including

general scenes [62], facial avatars [10, 54], and full-body avatars [15, 22, 27, 33, 43, 67], where 3D Gaussian kernels are bound to an articulated human model SMPL [28] through learned skinning weight. We adopt GS as our 3D shape representation, which naturally allows SDS gradients to smoothly propagate through the articulated deformation and rendering pipeline.

Articulated Motion Reconstruction A closely related topic to our work is the reconstruction of articulated motions of deformable objects in under-constrained settings, especially from a monocular view. As a special case, the reconstruction of human body poses [11, 20, 51] benefits strongly from the availability of dedicated deformable models such as SMPL [28] and the abundance of 2D/3D pose data [25, 31]. In contrast, the reconstruction of general objects, such as animals and humans in loose clothing, has remained more challenging due to the inherent 3D ambiguity from a monocular view and a lack of reliable priors for their articulation. One line of works following BANMO [23, 46, 56–58] solve for a static 3D template and articulated motion simultaneously from a monocular video by leveraging various image measurements such as segmentation, optical flow, and DensePose. Several works from another line [16, 24, 49, 60, 61] train a neural network that predicts the shape template of animals and their body deformation conditioned on a single input image in a weakly-supervised manner. Ponymation [45] learns a motion VAE for horse motions from a collection of horse videos. There are also works [17, 39, 47, 63] that directly train generative models on articulated motions. Instead of extracting articulation from a particular input, our method focuses on generating novel skeleton motion utilizing video diffusion priors that have learned general visual knowledge.

Generating 4D and Physics-Based Dynamics Generating 4D content is inherently challenging as it demands high consistency not only along the temporal axis to maintain motion, but also across multiple viewpoints to ensure spatial and structural accuracy in the generated content. Some works attempt to directly build priors models in the 4D domain, including diffusion models [19, 53] and reconstruction model [41], where the limited availability of 4D data pose a challenge. Therefore, a large body of works [1, 18, 26, 40, 44, 48, 64, 66] distill 4D motion from a combination of generative models that operate in lower dimensions, including images, videos, and multi-view images (3D), which is a difficult problem to due to sparsity of supervision, noisy gradient from SDS, and high-degree of freedom in the optimization variables. Some works aim to improve the controllability of the 4D generation by introducing conditioning of trajectory [50] or sparse control points [50]. None of the works above allow the generated contents to be grounded in a physics-based manner. The exceptions

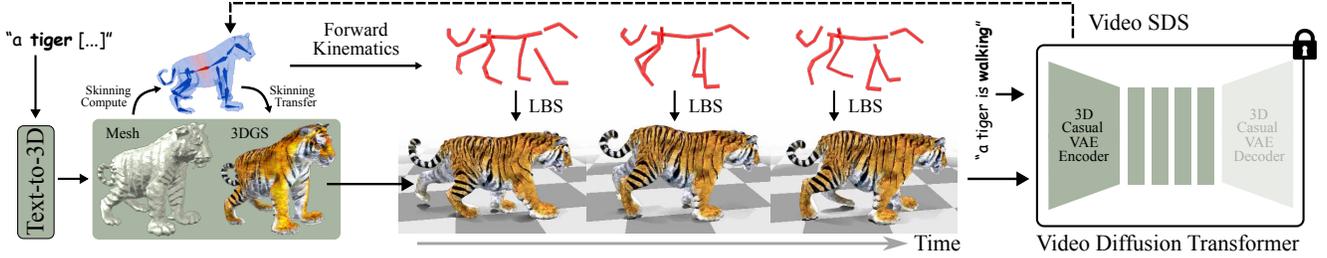


Figure 2. **Pipeline.** We novelly incorporate articulated skeletons into generative motion synthesis. With the low-dimensional parameterization of motions (a sequence of joint angles for articulated bones), the synthesis can focus more on motion modes instead of local-scale deformations. Given a text prompt, we use a text-to-3D method to generate a 3D asset. The asset is deformed by the skeleton and differentially rendered into videos. The SDS gradient is evaluated by a pre-trained video diffusion transformer and backpropagated to joint angles.

are PhysGaussian [52] and PhysDreamer [65], which models scene deformation by a Material-Point-Method (MPM) simulator. Such a formulation is more suitable for modeling the volumetric solid deformation and fluid dynamics, whereas our work focuses on articulated motion which can be simulated more efficiently by a skeleton-based representation.

3. Background

3.1. Deformable Gaussian Splatting

3D Gaussian Splatting (3DGS) [21] utilizes a set of 3D Gaussian kernels to represent a 3D scene. These kernels can be rendered into images using a customized differentiable rasterizer. Specifically, each kernel is defined by a set of parameters, $\sigma_p, \mathbf{x}_p, \Sigma_p, \mathcal{C}_p$, where σ_p denotes the opacity, \mathbf{x}_p and Σ_p represent the center and covariance matrix of the Gaussian ellipsoid, and \mathcal{C}_p is the coefficient set of spherical harmonics that determines the view-dependent colors of the kernel. Given a camera view, the color of a pixel is rendered by α -blending all kernels along the ray direction:

$$C = \sum_i c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (1)$$

where i is the sorted index of the kernels in ascending order of z -depth, c_i represents the kernel's color under the given view, evaluated from spherical harmonics, and α_i is the opacity σ_i weighted by the kernel's 3D Gaussian distribution. Given a set of images with known camera parameters, we can reconstruct the 3D scene using only RGB loss.

Following PhysGaussian [52], given a warping function $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ (defined in Sec. 4.1), we can deform the shape of each Gaussian kernel using the local linearization of ϕ at its center \mathbf{x}_p : $\phi(\mathbf{x}) \approx \phi(\mathbf{x}_p) + \nabla\phi(\mathbf{x}_p)(\mathbf{x} - \mathbf{x}_p)$, resulting in:

$$\tilde{\mathbf{x}}_p = \phi(\mathbf{x}_p), \quad \tilde{\Sigma}_p = \nabla\phi(\mathbf{x}_p)\Sigma_p\nabla\phi(\mathbf{x}_p)^T, \quad (2)$$

where (\mathbf{x}_p, Σ_p) and $(\tilde{\mathbf{x}}_p, \tilde{\Sigma}_p)$ denote the Gaussian parameters before and after the deformation respectively. For sim-

plicity, we can hold opacities and spherical harmonics constant. Our deformation system adheres to this convention to generate Gaussian kernels in motion.

3.2. Video Score Distillation Sampling

The Score Distillation Sampling (SDS) method, proposed to distill 3D models from text-to-image priors [36], uses an approximated gradient of the diffusion loss [14]:

$$\mathcal{L}_{\text{Diff}}(\theta, \mathbf{z}, y) = \mathbb{E}_{t, \epsilon} [w(t) \|\hat{\epsilon} - \epsilon\|_2^2], \quad (3)$$

where $t \sim \mathcal{U}(0, 1)$ is the diffusion time step, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is a Gaussian noise, $\mathbf{z}_t = \sqrt{\alpha_t}\mathbf{z} + \sqrt{1 - \alpha_t}\epsilon$ is the noisy image at step t , \mathbf{z} is the rendered image, y is the input text, and $\hat{\epsilon} = \epsilon_\theta(\mathbf{z}_t; t, y)$ is the noise predicted by the diffusion model. Omitting the gradient through the noise-predicting module, the gradient of \mathcal{L} w.r.t. \mathbf{z} is given by:

$$\nabla_{\mathbf{z}} \mathcal{L}_{\text{SDS}}(\mathbf{z}, y) = \mathbb{E}_{t, \epsilon} [w(t) (\hat{\epsilon} - \epsilon)]. \quad (4)$$

This pixel-level gradient is then backpropagated to the 3D model parameters via a differentiable renderer, guiding the generation of a 3D representation consistent with the input text. Videos, which are essentially 3D image tensors with an additional temporal axis, are handled similarly; the SDS gradient can be backpropagated to 4D representations.

4. Method

We present Articulated Kinematics Distillation, a text-driven articulated motion synthesis system for animating 3D assets, as illustrated in Fig. 2. We assume a rigged 3D asset as input. For results in this paper, we use an off-the-shelf text-to-3D generator, although assets from photogrammetry or created by artists should work similarly. We then convert the asset into a dual mesh-3DGS representation: the mesh is used to compute appropriate skinning weights, and the 3DGS supports differentiable deformations and rendering during SDS optimization. We rig the asset by manually embedding an articulated skeleton, and compute Linear

Blend Skinning (LBS) weights, which we transfer to nearby Gaussian kernels. To generate the motion, we optimize a sequence of joint angles within the articulation tree, which pass through a differentiable Forward Kinematics (FK) and 3DGS rasterization pipeline to render a video sequence. The rendered video is then fed into a pre-trained video diffusion model, which provides guidance for adjusting joint angles at each frame to produce text-aligned motion. In the following sections, we present the technical details of each component.

4.1. Rigging and Skinning for Gaussian Splatting

The rigging system is widely used in the animation industry, where 3D character meshes are controlled via an embedded articulated skeleton. Artists manipulate joint angles between connected bones to create 3D animations. The transformations of the deformed bones are evaluated using Forward Kinematics [8] of the skeleton tree, and then mesh vertices are moved by interpolating transformations from nearby bones. The most common interpolation scheme is Linear Blend Skinning (LBS) [30], which defines the deformation function as:

$$\phi(\mathbf{x}) = \sum_{i=1}^B w_i (\mathbf{R}_i \mathbf{x} + \mathbf{T}_i), \quad (5)$$

where B is the number of bones, $(\mathbf{R}_i, \mathbf{T}_i)$ is the rigid transformation of bone i , and w_i is the skinning weight, satisfying $\sum_i w_i = 1$. This deformation field is usually applied to mesh vertices, but it is defined on the entire 3D space, meaning that it can also deform Gaussian kernels following Eq. (2), where $\nabla \phi(\mathbf{x}) = \sum_{i=1}^B w_i \mathbf{R}_i$. We transfer skinning weights to Gaussian kernels by barycentric interpolation of skinning weights at their nearest points on the template mesh. For the assets generated by text-to-3D methods, we use Pinocchio [4] to compute skinning weights for mesh vertices automatically. Please refer to our supplementary document for more details.

4.2. Rendering

For each deformed frame of the 3DGS asset, we use the differentiable Gaussian Splatting rasterizer [21] to render it into an image. These rendered frames are then concatenated sequentially to form a video tensor.

Ground Rendering. Unlike previous text-to-4D frameworks [2, 66] which often render environments in nearly uniform colors, we incorporate a checkerboard ground. We find that this setup can provide the distillation with clues about the interaction between assets and the ground. It helps reduce relative motions between contacting parts and the ground, and helps keep assets grounded. We render the checkerboard ground as a background layer by ray casting and blend it with the rendering of the asset. For each ray

corresponding to a pixel, the color is determined by the intersection point between the ray and the ground; if there is no intersection, the color is set to a pre-assigned value. We also set the opacities of kernels located below the ground to zero to account for occlusion.

Camera Trajectory. We let the camera smoothly follow the bounding box center of the deformed shape in each frame, as the video model we use often generates object-centric videos. It is equivalent to moving the entire scene in the opposite direction while keeping the camera view fixed.

4.3. SDS for V-Prediction Diffusion Models

The video model we use, CogVideoX-5B [59], is a Diffusion Transformer (DiT) trained with a v -prediction formulation [42]. The diffusion loss is defined as:

$$\mathcal{L}_{\text{Diff}}(\theta, \mathbf{z}, y) = \mathbb{E}_{t, \epsilon} \left[w(t) \|\mathbf{z} - \hat{\mathbf{z}}\|_2^2 \right], \quad (6)$$

where $\hat{\mathbf{z}} = \sqrt{\alpha_t} \mathbf{z}_t - \mathbf{v}_\theta(\mathbf{z}_t; t, y)$ is the reconstruction based on the predicted velocity by the diffusion model \mathbf{v}_θ . Omitting the gradient through the velocity-predicting transformer, the SDS gradient is evaluated as:

$$\nabla_{\mathbf{z}} \mathcal{L}_{\text{SDS}}(\mathbf{z}, y) = \mathbb{E}_{t, \epsilon} [w(t) (\mathbf{z} - \hat{\mathbf{z}})]. \quad (7)$$

Derivations are provided in the supplementary document.

Modern video diffusion models are trained in latent space by encoding raw videos using Variational Autoencoders (VAE). Correspondingly, the SDS gradient is computed on the latent codes, and then needs to be back-propagated through the VAE encoder into the pixel space. This process can be extremely memory-intensive, especially when handling a large number of frames. To reduce the memory footprint, we employ the gradient checkpointing techniques [7]. These techniques save memory by selectively storing intermediate activations and recomputing them during the backward pass, making it feasible to perform SDS optimization with large DiTs.

4.4. Optimization

Given all the components, we now present the optimization process of our framework. Our approach uses 3-DoF compound spherical joints to connect bones, where each DoF represents the rotation angles around one of three linearly independent axes. The optimizable variables for each asset consist of $\Theta = \left\{ \{\mathcal{A}_i^j\}_{j=1}^{B-1}, \mathcal{T}_i \right\}_{i=0}^{F-1}$, where F is the number of frames, \mathcal{A}_i^j is the 3D angle vector for joint j at frame i , and \mathcal{T}_i denotes the 6-DoF rigid transformation of the root bone in the articulation tree at frame i . We use the following loss function for SDS optimization:

$$\mathcal{L} = \mathcal{L}_{\text{SDS}} + \lambda_1 \mathcal{L}_{\text{smooth}} + \lambda_2 \mathcal{L}_{\text{ground}}. \quad (8)$$

Besides SDS loss, we incorporate two regularizers: a smoothness penalty $\mathcal{L}_{\text{smooth}}$ over time to encourage time-consistent deformations, and a ground penetration penalty $\mathcal{L}_{\text{ground}}$ to enforce the assets to stay above the ground.

Since our parameters have physical meanings, we can directly enforce smoothness on the control parameters, which is defined as the mean absolute error (MAE) of the Laplacian of the control parameters w.r.t. time, excluding the first and last frames:

$$\mathcal{L}_{\text{smooth}} = \text{MAE}(\Delta_t \Theta), \quad (\Delta_t \Theta)_i = \Theta_{i-1} - 2\Theta_i + \Theta_{i+1}. \quad (9)$$

This loss helps ensure that changes in the parameters across consecutive frames are gradual, annealing the noisy nature of the SDS loss. The ground penetration loss is defined as

$$\mathcal{L}_{\text{ground}} = \frac{1}{|V|} \sum_{v \in V} \max(-v^y, 0). \quad (10)$$

Here, we conceptualize bones as a set of transformed cuboids, and V represents the set of vertices of these cuboids. This loss penalizes any penetration of the asset below the ground plane by applying a penalty proportional to the depth below ground for each vertex.

We focus on motion synthesis for legged characters, including animals and humanoids, moving across the ground. In our experiments, we initialize the asset’s forward motion along x -axis by setting an initial displacement as $\mathcal{T}_i^x = vi$ for some constant velocity v . The pace and trajectories of the assets can be further optimized during the distillation.

4.5. Physics-Based Motion Tracking

The articulated skeletons embedded in the asset can be deployed in articulated rigid body simulators. To ground the synthesized motion in physics, we can further project the distilled motion trajectory to the nearest solution achievable in physics-based tracking in a simulation environment. We accomplish this generation-to-simulation transition by searching for a physical joint control sequence that minimizes the difference between simulated and synthesized bone trajectory.

Specifically, we treat each bone as a rigid cuboid and use compound joints to connect bones, which is consistent with the motion DoFs during motion distillation. We use a semi-implicit articulated rigid body simulator to simulate the skeleton under gravity and ground collision. The simulation process can be abstracted as a state transition \mathcal{S} :

$$[\mathbf{q}^{k+1}, \dot{\mathbf{q}}^{k+1}] = \mathcal{S}([\mathbf{q}^k, \dot{\mathbf{q}}^k], \Delta t, \boldsymbol{\tau}^k) \quad (11)$$

where \mathbf{q}^k is the concatenation of generalized coordinates of bones at the time step k , which represent 6-DoF rigid transformations, $\dot{\mathbf{q}}^k$ is the generalized velocity, Δt is the simulation time step size, and $\boldsymbol{\tau}^k$ is active joint torques applied

to the connected bones at joints. We use a Proportional-Derivative (PD) controller to provide active joint torques, where the torque at joint j around axis l is given by:

$$\boldsymbol{\tau}^k|_l = k_e(\hat{\theta}_{jl} - \theta_{jl}) - k_d\dot{\theta}_{jl} \quad (12)$$

where $\hat{\theta}_{jl}$ is the control variable, and θ_{jl} is the current joint angle. This formulation tends to pull the joint angle θ_{jl} towards $\hat{\theta}_{jl}$ subject to damping. k_e, k_d are globally defined parameters that control joint elasticity and damping, respectively. The whole simulation process can be implemented using AutoDiff frameworks such as Warp [29]. Altogether, we use the following motion-tracking loss:

$$\mathcal{L}_{\text{track}}(\hat{\Theta}, \hat{\mathbf{q}}^0) = \frac{1}{F-1} \sum_{i=1}^{F-1} \|\mathbf{q}^{iN} - \hat{\mathbf{q}}^i\|_1 + \lambda_3 \text{MAE}(\Delta_t \hat{\Theta}). \quad (13)$$

Here, N is the ratio between the frame time of the target motion $\{\hat{\mathbf{q}}^i\}$ and the simulation time step Δt , and $\hat{\Theta}$ is the concatenation of control variable per simulation step. Like in the distillation, we add a smoothness regularizer. We also optimize the initial generalized velocity $\hat{\mathbf{q}}^0$ as well. The simulation starts from $\mathbf{q}^0 = \text{Proj}(\hat{\mathbf{q}}^0)$, where $\hat{\mathbf{q}}^0$ is vertically shifted such that the lowest bone touches the ground.

However, N is typically in the hundreds for explicit simulators, resulting in thousands of simulation steps in total. This can cause severe gradient explosion issues during backpropagation. A common technique to alleviate this issue is to apply gradient clipping before each optimizer step. However, the explosion may occur multiple times throughout backpropagation, which can make the final gradient useless. To address this, we employ a fine-grained gradient clipping strategy, applying gradient clipping every few tens of backward steps. We achieve this by wrapping chunks of substeps into a PyTorch layer and manually defining the backpropagation logic.

5. Implementation

We use an off-the-shelf SDS-based text-to-3D system, Tet-Splatting [12], to generate static 3D mesh assets from texts. The text-to-3D module is not the focus of this work and can be replaced by any suitable framework. We set up a skeleton inside the mesh manually in Blender [5] using its armature system. This manual setup typically takes only a few minutes per instance. We convert the generated mesh into the Gaussian Splatting representation by applying 3DGS reconstruction [21] to images of the mesh from 200 random views, starting from mesh vertices as initial kernel centers.

We adopt threestudio [13] for SDS optimization, and CogVideoX-5B [59] as the video diffusion model. By leveraging the gradient checkpointing mechanism of the VAE, all our experiments can be run on a single NVIDIA A100-40GB graphics card. We adopt forward kinematics and articulated rigid body simulation from Warp [29] and integrate it with PyTorch for data and gradient interchange.

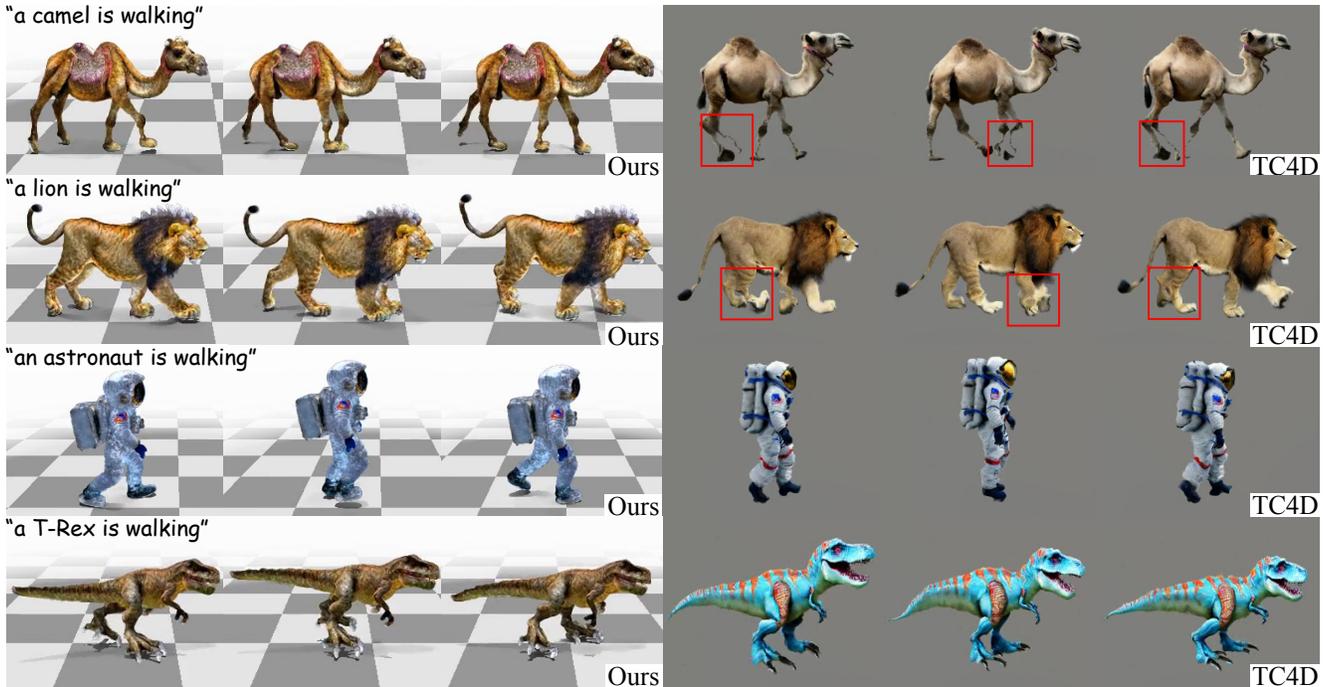


Figure 3. Qualitative comparisons with TC4D. The blurry artifacts generated by TC4D are highlighted. TC4D often fails to produce alternating leg movements (e.g., in the astronaut example), or shows limited local-scale motion (e.g., in the T-Rex example).

Each asset undergoes 10,000 iterations of SDS optimization, requiring approximately 25 hours.

6. Experiment

6.1. Quantitative Comparisons

We generate 29 static assets, including animals and humanoids, and compare our method with the state-of-the-art open-sourced approach, TC4D [2], that can synthesize long-trajectory motions. TC4D moves the bounding box of the object along a predefined path, while our method only initializes a path and allows adjustments based on the guidance from the video model. Other text-to-4D frameworks, such as those in [1, 66], are limited to synthesizing local-scale motions, which tend to have constrained magnitudes. We compare our method (without physics-based tracking for fairness) with TC4D using both automated video scores and a user study.

6.1.1. Metrics

Automated Metric We utilize VideoPhy [3] to automatically score videos. This model scores two types of metrics: Semantic Adherence (SA), which assesses the alignment between the text and the video content, and Physical Commonsense (PC), which evaluates whether the video adheres to basic physical laws. We observed that the evaluation of PC scores tends to concentrate on the local deformations of objects. For a fair comparison with TC4D using this metric,

we exclude ground rendering when using this metric.

User Study We recruited 20 human evaluators to compare video clip pairs generated by AKD and TC4D using the same text prompts. We asked the evaluators to assess each comparison across several aspects: Motion Amount (MA), Physical Plausibility (PP), and Text Alignment (TA). Evaluators were instructed to choose the clip they felt performed better for each aspect.

6.1.2. Result

The automatic scores evaluated by VideoPhy are provided in the inset table where the means and standard deviations of each score are reported.

	SA	PC
TC4D	0.40 ± 0.34	0.31 ± 0.15
Ours	0.81 ± 0.26	0.39 ± 0.17

Our method outperforms TC4D in terms of both metrics. The user study results show that our method was preferred in 51%, 53%, and 53% of the comparisons in terms of MA, PP, and TA, respectively. These results indicate that our method can synthesize motions in better quality.

6.2. Qualitative Comparisons

In Fig. 3, we visualize several comparisons between TC4D and ours. We refer readers to the supplemental video for complete results. Note that appearance is not our focus because the 3D assets can be generated by any text-to-3D method. We observe that TC4D rarely produces alternat-

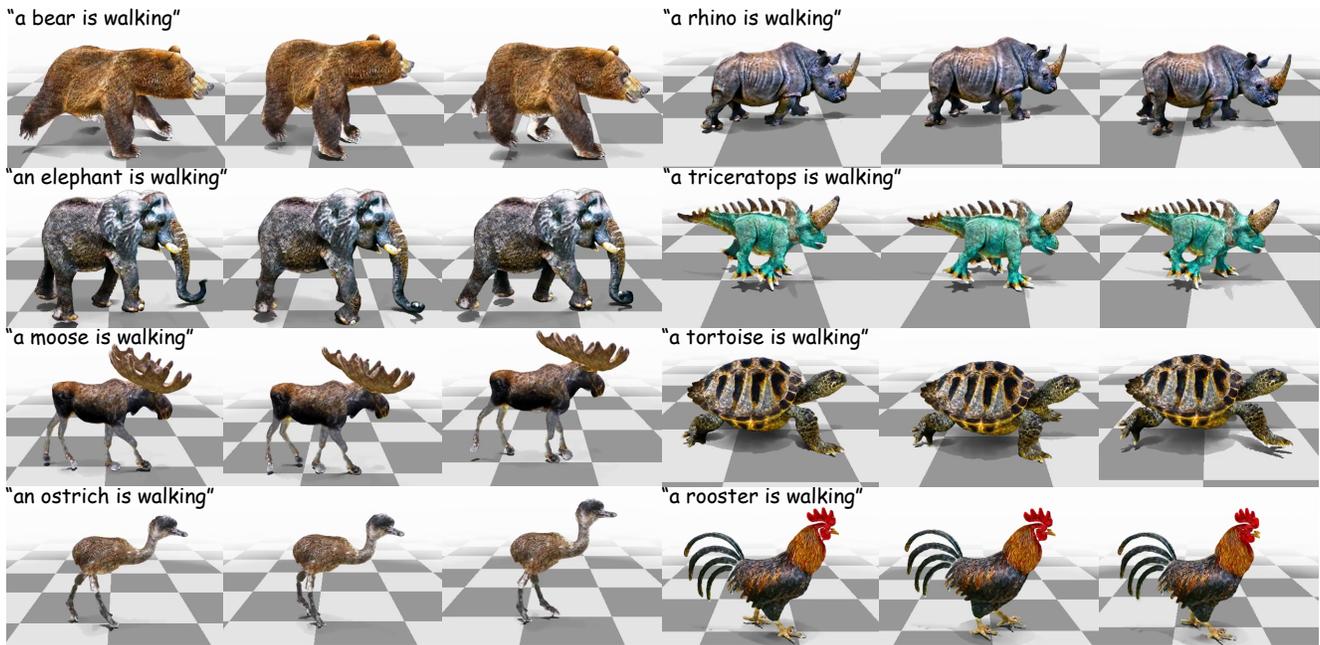


Figure 4. Examples of our synthesized motions.

ing leg movements, and areas where the legs converge often appear blurry. In contrast, our method, with its low-DoF parameterization, effectively maintains the geometric structure of each part within the assets during deformations.

6.3. Physics-based Motion Tracking

The synthesized articulated motions may still float above the ground or slide relative to it, as video diffusion models cannot provide guidance for such fine-grained placements. However, the overall styles of motions are well captured. In Fig. 5, we present examples where the synthesized motions appear above the ground, while the physics-based tracking results adhere to gravity and have frictional contact with the ground, yet still track the target motion styles. For more results, we refer the reader to the supplemental video.

6.4. Synthesis Diversity

Asset Diversity. In Fig. 4, we visualize more examples of motions by our method on different assets with varying articulated topologies. We refer readers to the supplemental video for animations.

Motion Diversity. Our method supports the synthesis of different motion modes based on different text prompts. In Fig. 6, we illustrate the differences between “walking” and “running” behaviors. For quadrupeds, the limbs on each side alternate during walking, whereas they move in sync during running.

6.5. Ablation Studies

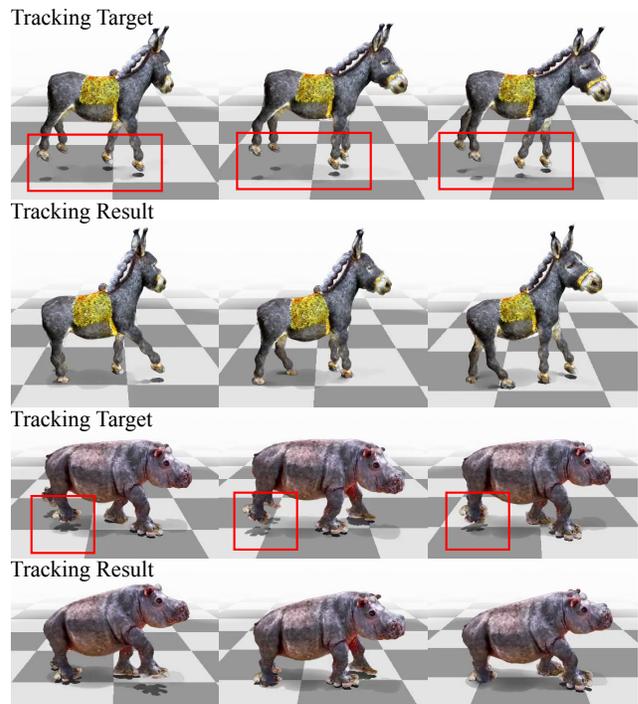


Figure 5. We use physics-based motion tracking to project synthesized motions onto physics-grounded trajectories.

Components in Training.

We conducted ablation studies on several components in addition to

	Ostrich	w.o. ground	Tiger	w.o. $\mathcal{L}_{\text{ground}}$	w.o. $\mathcal{L}_{\text{smooth}}$
SA	0.982	0.777	0.989	0.979	0.984
PC	0.294	0.107	0.321	0.269	0.294

SDS loss: ground rendering, ground penalty loss, and

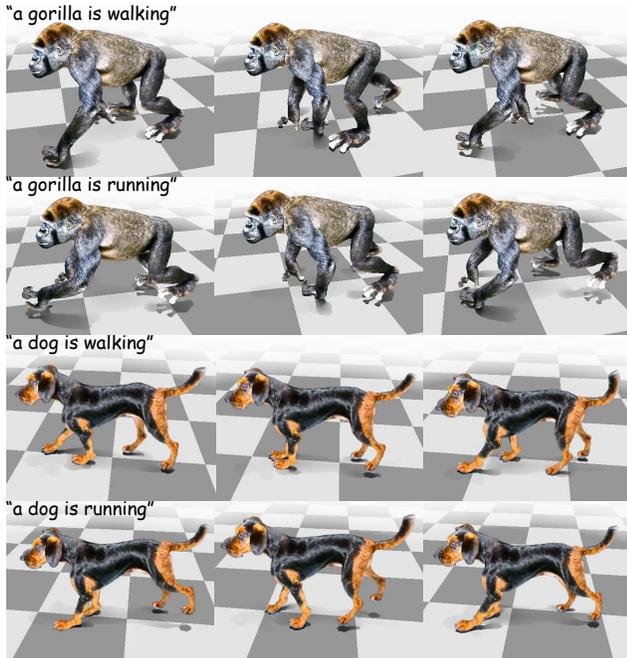


Figure 6. Our method supports synthesizing different motions based on varying text descriptions.

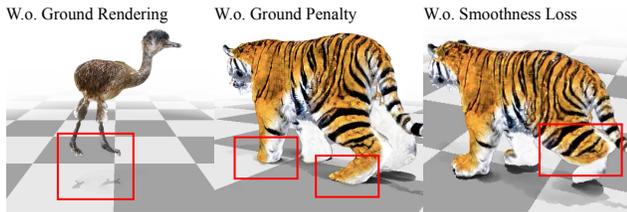


Figure 7. Ablation studies on ground rendering, ground penalty loss, and the smoothness loss. The artifacts are highlighted.

smoothness loss, each targeting specific types of artifacts. The visualizations are presented in Fig. 7. The automated quantitative metrics evaluated using VideoPhy are reported in the inset table. Without ground rendering, the synthesized motion appears above the ground because the video model lacks information about the ground’s location. The absence of the ground penalty loss results in penetration into the ground. The smoothness loss can encourage time consistency across frames; without it, dramatic and unnatural deformations may occur in certain frames. Both SA and PC scores decrease when one of these component is removed.

Text-to-3D Module. In this paper, the text-to-3D module is not our focus. We use Tet-Splatting because of its efficiency. This module can be replaced by arbitrary text-to-3D frameworks. For example, we can extract assets from the static phase of TC4D. One example is shown in Fig. 9.

Video Diffusion Model. The video model and the articulated representation both improve the quality of synthesized

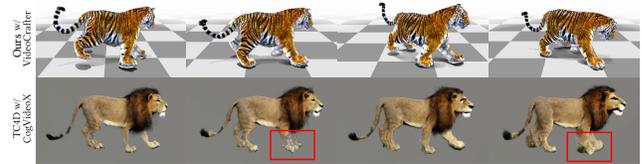


Figure 8. Ablation study on the base video diffusion model.

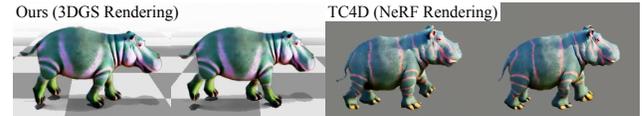


Figure 9. Ablation on the text-to-3D module. We extract an asset from TC4D and achieve a comparable appearance.

motions. In Fig. 8, our method with VideoCrafter generates alternating leg motions but suffers from severe foot-skating issues. TC4D with CogVideoX still fails to produce alternating leg motions and exhibits blurry artifacts.

7. Conclusion

In this paper, we present AKD, a text-driven method for articulated motion synthesis powered by large video diffusion models. By distilling the motion knowledge from a video generation model, our model offers an alternative to laborious motion authoring in traditional animation pipelines. The low-DoF, skeleton-based parameterization of motion allows the distillation process to focus more on overall articulated motion patterns rather than local-scale shape deformations, resolving the common issue of shape inconsistency as in previous work on 4D generation, and thus results in improved physical plausibility of the generated motion. We demonstrate that the generated skeleton motions can be transferred into simulation environments via physics-based motion tracking. We hope this work can inspire future research in generating labeled data for robotics applications.

Our approach has several limitations that point to future work. The visual quality of our generated assets is still suboptimal; improving the visual quality can help reduce the gap between rendered outputs and the realistic video distributions encoded in video models. The diversity of motions produced by our method largely depends on the video model’s ability to synthesize desired motions; improved video priors in the future could enhance this diversity. Our method focuses on the articulated motion of objects, and may not be suitable for other types of deformation such as soft-body dynamics. Additionally, our pipeline assumes a manually rigged skeleton. To scale up our method, future work can leverage automatic rigging techniques such as RigNet [55] to generalize to diverse character types, and can use more efficient sampling techniques for video models to accelerate convergence.

References

- [1] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7996–8006, 2024. 1, 2, 6
- [2] Sherwin Bahmani, Xian Liu, Wang Yifan, Ivan Skorokhodov, Victor Rong, Ziwei Liu, Xihui Liu, Jeong Joon Park, Sergey Tulyakov, Gordon Wetzstein, et al. Tc4d: Trajectory-conditioned text-to-4d generation. In *European Conference on Computer Vision*, pages 53–72. Springer, 2025. 1, 2, 4, 6
- [3] Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. *arXiv preprint arXiv:2406.03520*, 2024. 6
- [4] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG)*, 26(3):72–es, 2007. 4
- [5] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, 2024. 5
- [6] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024. 1
- [7] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. 4
- [8] Jacques Denavit and Richard S Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. 1955. 4
- [9] Yutao Feng, Yintong Shang, Xuan Li, Tianjia Shao, Chenfanfu Jiang, and Yin Yang. Pie-nerf: Physics-based interactive elastodynamics with nerf. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4450–4461, 2024. 2
- [10] Simon Giebenhain, Tobias Kirschstein, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Npga: Neural parametric gaussian avatars. *arXiv preprint arXiv:2405.19331*, 2024. 2
- [11] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14783–14794, 2023. 2
- [12] Chun Gu, Zeyu Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Tetrahedron splatting for 3d generation. In *NeurIPS*, 2024. 5
- [13] 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. 5
- [14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3
- [15] Shoukang Hu, Tao Hu, and Ziwei Liu. Gauhuman: Articulated gaussian splatting from monocular human videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20418–20431, 2024. 2
- [16] Tomas Jakab, Ruining Li, Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Farm3d: Learning articulated 3d animals by distilling 2d diffusion. In *2024 International Conference on 3D Vision (3DV)*, pages 852–861. IEEE, 2024. 2
- [17] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023. 2
- [18] Yanqin Jiang, Li Zhang, Jin Gao, Weimin Hu, and Yao Yao. Consistent4d: Consistent 360 $\{\backslash\deg\}$ dynamic object generation from monocular video. *arXiv preprint arXiv:2311.02848*, 2023. 2
- [19] Yanqin Jiang, Chaohui Yu, Chenjie Cao, Fan Wang, Weiming Hu, and Jin Gao. Animate3d: Animating any 3d model with multi-view video diffusion. *arXiv preprint arXiv:2407.11398*, 2024. 2
- [20] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018. 2
- [21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3, 4, 5
- [22] Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. Hugs: Human gaussian splats. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 505–515, 2024. 2
- [23] Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19876–19887, 2024. 2
- [24] Zizhang Li, Dor Litvak, Ruining Li, Yunzhi Zhang, Tomas Jakab, Christian Rupprecht, Shangzhe Wu, Andrea Vedaldi, and Jiajun Wu. Learning the 3d fauna of the web. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9752–9762, 2024. 2
- [25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2
- [26] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8576–8588, 2024. 2

- [27] Xinqi Liu, Chenming Wu, Jialun Liu, Xing Liu, Chen Zhao, Haocheng Feng, Errui Ding, and Jingdong Wang. Gva: Reconstructing vivid 3d gaussian avatars from monocular videos. *CoRR*, 2024. 2
- [28] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023. 2
- [29] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. <https://github.com/nvidia/warp>, 2022. NVIDIA GPU Technology Conference (GTC). 5
- [30] Thalmann Magnenat, Richard Laperrière, and Daniel Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings of Graphics Interface '88*, pages 26–33. Canadian Inf. Process. Soc, 1988. 4
- [31] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 2
- [32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2
- [33] Arthur Moreau, Jifei Song, Helisa Dharmo, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Human gaussian splatting: Real-time rendering of animatable avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 788–798, 2024. 2
- [34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 2
- [35] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2
- [36] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. 3
- [37] 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. 2
- [38] Yi-Ling Qiao, Alexander Gao, Yiran Xu, Yue Feng, Jia-Bin Huang, and Ming C Lin. Dynamic mesh-aware radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 385–396, 2023. 2
- [39] Sigal Raab, Inbal Leibovitch, Guy Tevet, Moab Arar, Amit Haim Bermano, and Daniel Cohen-Or. Single motion diffusion. In *ICLR*, 2024. 2
- [40] 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. 2
- [41] Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, et al. L4gm: Large 4d gaussian reconstruction model. *arXiv preprint arXiv:2406.10324*, 2024. 2
- [42] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 4
- [43] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1606–1616, 2024. 2
- [44] 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. 2
- [45] Keqiang Sun, Dor Litvak, Yunzhi Zhang, Hongsheng Li, Jijun Wu, and Shangzhe Wu. Ponymation: Learning articulated 3d animal motions from unlabeled online videos. In *European Conference on Computer Vision*, pages 100–119. Springer, 2025. 2
- [46] Jeff Tan, Donglai Xiang, Shubham Tulsiani, Deva Ramanan, and Gengshan Yang. Dressrecon: Freeform 4d human reconstruction from monocular video. *arXiv preprint arXiv:2409.20563*, 2024. 2
- [47] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. 2
- [48] Lukas Uzolas, Elmar Eisemann, and Petr Kellnhofer. Motiondreamer: Exploring semantic video diffusion features for zero-shot 3d mesh animation, 2024. 2
- [49] Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. Magicpony: Learning articulated 3d animals in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8792–8802, 2023. 2
- [50] Zijie Wu, Chaohui Yu, Yanqin Jiang, Chenjie Cao, Fan Wang, and Xiang Bai. Sc4d: Sparse-controlled video-to-4d generation and motion transfer. In *European Conference on Computer Vision*, pages 361–379. Springer, 2025. 2
- [51] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10965–10974, 2019. 2
- [52] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024. 2, 3

- [53] Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency. *arXiv preprint arXiv:2407.17470*, 2024. [2](#)
- [54] Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2024. [2](#)
- [55] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: Neural rigging for articulated characters. *arXiv preprint arXiv:2005.00559*, 2020. [8](#)
- [56] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2863–2873, 2022. [2](#)
- [57] Gengshan Yang, Chaoyang Wang, N Dinesh Reddy, and Deva Ramanan. Reconstructing animatable categories from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16995–17005, 2023.
- [58] Gengshan Yang, Shuo Yang, John Z Zhang, Zachary Manchester, and Deva Ramanan. Ppr: Physically plausible reconstruction from monocular videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3914–3924, 2023. [2](#)
- [59] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. [1](#), [4](#), [5](#)
- [60] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Lassie: Learning articulated shapes from sparse image ensemble via 3d part discovery. *Advances in Neural Information Processing Systems*, 35:15296–15308, 2022. [2](#)
- [61] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Hi-lassie: High-fidelity articulated shape and skeleton discovery from sparse image ensemble. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4853–4862, 2023. [2](#)
- [62] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024. [2](#)
- [63] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16010–16021, 2023. [2](#)
- [64] 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. In *European Conference on Computer Vision*, pages 163–179. Springer, 2025. [2](#)
- [65] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snively, Jiajun Wu, and William T Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. In *European Conference on Computer Vision*, pages 388–406. Springer, 2025. [3](#)
- [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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7300–7309, 2024. [2](#), [4](#), [6](#)
- [67] Yang Zheng, Qingqing Zhao, Guandao Yang, Wang Yifan, Donglai Xiang, Florian Dubost, Dmitry Lagun, Thabo Beeler, Federico Tombari, Leonidas Guibas, et al. Physavatar: Learning the physics of dressed 3d avatars from visual observations. *arXiv preprint arXiv:2404.04421*, 2024. [2](#)