Choreographing a World of Dynamic Objects

Yanzhe Lyu[1,*,†] Chen Geng[1,*] Karthik Dharmarajan[1]
Yunzhi Zhang[1] Hadi Alzayer[1,3] Shangzhe Wu[2] Jiajun Wu[1]

[1]Stanford University [2]University of Cambridge [3]University of Maryland

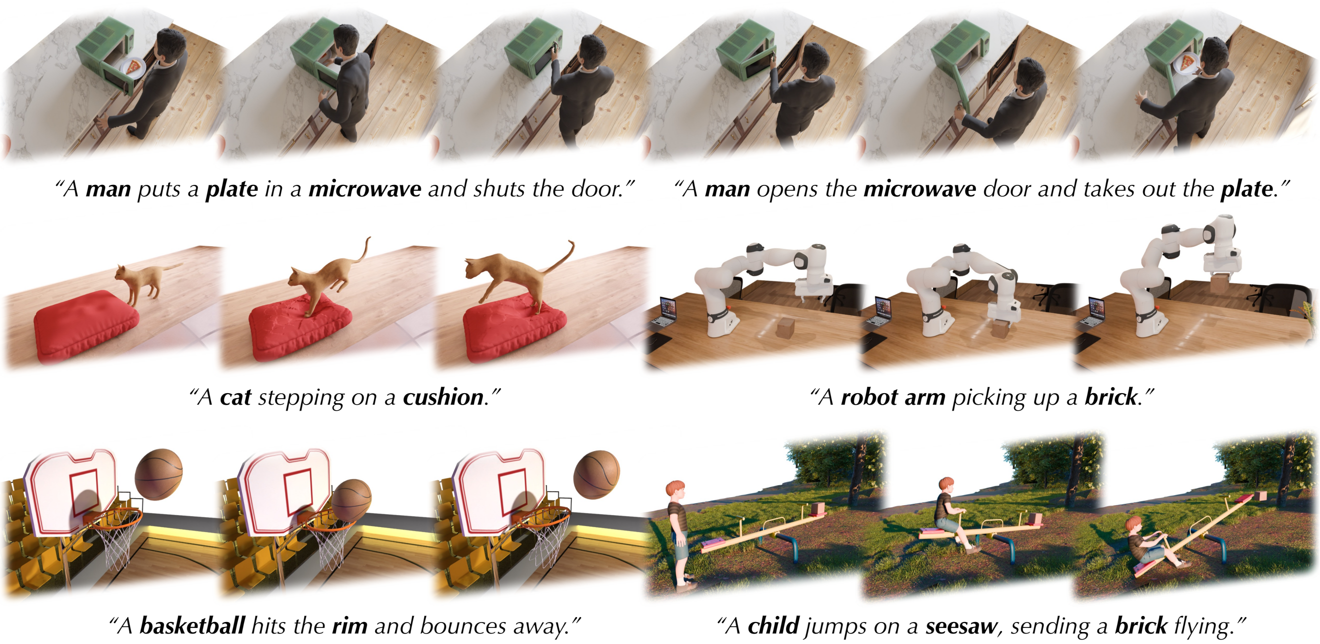

Figure 1. **4D scene motion generated by our method.** We present CHORD, a universal generative pipeline capable of animating scenes with **multiple** objects that interact with each other. Project page: https://yanzhelyu.github.io/chord

Abstract

Dynamic objects in our physical 4D (3D + time) world are constantly evolving, deforming, and interacting with other objects, leading to diverse 4D scene dynamics. In this paper, we present a universal generative pipeline, CHORD, *for* ***CHOR****eographing* ***D****ynamic objects and scenes and synthesizing this type of phenomena. Traditional rule-based graphics pipelines to create these dynamics are based on category-specific heuristics, yet are labor-intensive and not scalable. Recent learning-based methods typically demand large-scale datasets, which may not cover all object categories in interest. Our approach instead inherits the universality from the video generative models by proposing a distillation-based pipeline to extract the rich Lagrangian motion information hidden in the Eulerian representations of 2D videos. Our method is universal, versatile, and category-agnostic. We demonstrate its effectiveness by conducting experiments to generate a diverse range of multi-body 4D dynamics, show its advantage compared to existing methods, and demonstrate its applicability in generating robotics manipulation policies.*

1. Introduction

Humans and other embodied agents live in a 4D (3D + time) world, a world composed of a diverse range of dynamic objects, *i.e.*, objects that can evolve, deform, or interact with other objects. Creating 4D motions for both object deformations and interactions is crucial when building 3D world models for robotics [41, 68] and embodied AI [29].

Traditionally, it has been challenging to generate such motions for a scene composed of dynamic objects in their static snapshots because it requires extensive manual mod-

*Equal contribution. †Work was done when Y. Lyu was a visiting student at Stanford University. Y. Lyu is currently with the University of Science and Technology of China.

eling and expert labor. Recent approaches [67] have attempted to learn such 4D generators purely from data in an end-to-end manner. However, most existing datasets [9] focus on the internal deformations and evolutions of an individual object with little to no coverage on their interactions, and 4D data describing both deformations of objects and object interactions is extremely rare. This scarcity on scene-level 4D dynamics has rendered existing data-driven approaches into only being capable of generating dynamics of a single object.

Inspired by the recent success of general-purpose video generative models, we use a different approach to tackle this problem: distilling these scene motions from video generative models. At a high level, we iteratively optimize the low-level Lagrangian deformations of each object. At each optimization step, we deform the 3D scene and render it from certain viewpoints, and let video generative models judge whether the deformation is plausible. Through this process, we essentially leverage video models as a high-level "choreographer" to plan the motions of individual objects and make them consistent with each other.

Despite the promise of this distillation-based paradigm, getting plausible results with it has been challenging. Existing methods [2, 24, 58] mainly operate at the object level and often show noticeable artifacts in the generated motion. Two major obstacles hinder these approaches from working effectively in our setting: (1) 4D deformations are both spatially high-dimensional and temporally ill-regularized, and (2) the non-conventional architecture designs of modern video generative models are not compatible with existing distillation algorithms [46].

We address the first challenge by analyzing the inherent locality of 4D deformations: temporal deformation fields should be locally smooth in both space and time. To this end, we design a coarse-to-fine 4D motion representation that injects hierarchical structures to both the spatial and the temporal domain. Spatially, we adopt a bi-level control point-based representation that disentangles fine-grained motion details from coarse transformations. Temporally, inspired by a time-honored data structure in theoretical algorithm design, i.e. the Fenwick tree [11, 27], we store deformations in a cumulative, range-based structure that implicitly enforces temporal coherence and improves the learnability of long-horizon motion. With these two innovations, our novel 4D representation is robust, stable, and supports generating a diverse range of motions.

The second challenge stems from modern video generative models being based on flow-based models [37]. These models are incompatible with the traditional distillation algorithms. Therefore, we propose a novel strategy for distillation from modern rectified flow-based video generative models. We derive a novel Score Distillation Sampling (SDS) [46] target for flow-based video diffusion models and analyze their noise pattern, thus enabling video models to effectively provide guidance to our 4D representation.

By proposing these two innovations and the framework to choreograph object motion, we arrive at a simple yet elegant solution to the challenging problem of generating 4D-consistent motion of dynamic objects in a scene. We name this pipeline "CHORD", for **CHOR**eographing **D**ynamic objects and scenes. CHORD is universal, versatile, and applicable across a wide range of dynamic phenomena. We evaluate our framework on diverse dynamic objects and compare it against prior art and show clear advantages.

Beyond visual generation, our pipeline also enables the robot manipulation in the physical world by generating physically-grounded Lagrangian deformation trajectories of real-world objects. We demonstrate this by leveraging the generated 3D trajectories to plan the motion of a real robot and showing that they can guide zero-shot manipulation of diverse dynamic objects.

In summary, our contributions are as follows:

1. A 4D motion representation that combines a Fenwick tree–inspired cumulative temporal structure with a hierarchical low-to-high DoF parameterization, making it well-suited for distillation-based 4D generation.
2. A distillation strategy for modern flow-based video generative models to make SDS algorithms effective on generating 4D motions from 2D video generative models.
3. A robust framework to generate physically-grounded 4D motions for diverse dynamic objects that are applied to learning real-world robotic manipulation policies.

2. Related Work

Object-Level 4D Generation. Generating 4D consistent object deformations has been a long-standing challenge in the community. Traditional approaches first determine category-specific kinematic models (*i.e.*, rigging representations) [4, 5, 17, 36, 38, 44, 65, 81] and then generate motion based on them [15, 22, 35, 40, 45, 51, 53, 57, 71], which inherently limits these methods to constrained categories. Some methods [67, 75, 79] attempt to learn end-to-end 4D generators from existing 4D object datasets [8–10], but they struggle to generalize beyond humanoid characters since most existing datasets are dominated by animated human-like models. Other approaches [2, 14, 16, 23, 31, 33, 39, 42, 48, 49, 54, 55, 58, 61, 69, 72, 73, 76, 78] avoid supervised learning by performing 4D reconstruction or distillation from video generative models, yet they typically yield minor and unrealistic motion due to the difficulty of optimizing high-dimensional 4D motion and the noise in the guidance signals. Our framework addresses these limitations by assuming neither category-specific kinematic structure nor large-scale 4D datasets, and generates realistic 4D motion for arbitrary objects.

Scene-Level 4D Generation. Scene-level 4D generation extends beyond the object-centric setting, introducing substantially more complexity and greater challenges. It must not only produce plausible object-level motion but also maintain motion consistency across multiple interacting objects. Therefore, existing methods often simplify the problem by restricting it to specific categories (*e.g.*, human-object interaction [19, 30, 62, 74]), enforcing physical constraints [6, 32, 34, 80], or conditioning on symbolic structures [1, 50]. Some approaches attempt to produce 4D scenes by reconstructing them from videos [7, 28, 60, 64] generated by video models, yet the resulting representation remains largely 2.5D and does not support full 360° view synthesis. Our approach is the first to tackle the challenging setting of generating scene-level 4D motion of objects without relying on any category-specific inductive bias.

4D Representations. A key component in 4D generation pipelines is the selection of the underlying 4D representation. Early works use high-dimensional deformation fields to represent 4D scenes [14, 43, 47, 63]. They work well for reconstruction targets with dense inputs, but are not suitable for generative tasks with noisy supervision signals. Recent works explore reducing the dimensionality of 4D representations in the spatial domain [16, 17, 60, 66]. Our hierarchical 4D representation strengthens this idea by injecting low-dimensionalities and hierarchies in both spatial and temporal domains, which serves as a backbone representation in our 4D generation framework.

3. Method

Given a 3D scene containing multiple dynamic objects represented by their static 3D snapshots, along with a text prompt describing how the scene should change over time (e.g., a man facing a lamp with the prompt "*the man lowers the head of the lamp with his hand*"), our goal is to generate a sequence of temporal deformations that drive the objects so that the resulting 3D animations aligns with the prompt.

Figure 2 shows an overview of our method. We iteratively optimize a 4D scene motion representation using guidance signals distilled from a video generative model. In the following section, we detail the three main components in this framework: a strategy for distillation from modern rectified flow-based video generative models (Sec. 3.2), a robust and general 4D scene motion representation (Sec. 3.3), and regularization terms to ensure stable optimization (Sec. 3.4).

3.1. Preliminary: Score Distillation Sampling

The Score Distillation Sampling method [46] was introduced to distill 3D assets from image diffusion models [18]. At each iteration, an image $\mathbf{z}$ is rendered from the 3D asset parameterized by θ. Gaussian noise ϵ is then added to produce a noisy image $\mathbf{z}_\tau$, where the noise level τ is uniformly sampled from $(0, 1)$. The noisy image $\mathbf{z}_\tau$ is subsequently fed into a image diffusion model, which predicts noise $\hat{\epsilon}$. SDS updates θ with the following gradient:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\theta; \mathbf{z}, \mathbf{y}) = \mathbb{E}_{\tau,\epsilon}\left[w(\tau)\left(\hat{\epsilon}\left(\mathbf{z}_\tau; \tau, \mathbf{y}\right) - \epsilon\right)\frac{\partial \mathbf{z}}{\partial \theta}\right], \tag{1}$$

where $w(\tau)$ is a weighting function.

Extending this idea to 4D generation follows the same principle: at each iteration, a video is rendered from the 4D asset, blended with noise, and then passed through the diffusion model, which provides gradients to update the 4D representation.

3.2. Distilling from Rectified Flow Models

The above-mentioned 4D SDS algorithm is conceptually simple, yet it is non-trivial to apply them to distill from modern video generative models. The major obstacle is the gap between the diffusion architecture used in the original SDS target and the Rectified Flow (RF)-based model architecture in modern video generative models, such as Wan 2.2 [59] used in our paper.

To mitigate this architectural gap, we derive a novel SDS target for RF models. Similar to the derivation of SDS gradients for diffusion models [46], we align the optimization objective with the model's training loss and express the SDS update rule for RF models as:

$$\begin{aligned}&\nabla_\theta \mathcal{L}_{\text{RFSDS}}(\theta; z, \mathbf{y}) = \\ &\mathbb{E}_{\tau,\epsilon}\left[w(\tau)\left(\hat{v}\left(\mathbf{z}_\tau; \tau, \mathbf{y}\right) - \epsilon + \mathbf{z}\right)\frac{\partial \mathbf{z}}{\partial \theta}\right],\end{aligned} \tag{2}$$

where τ is the noise level uniformly sampled from $(0, 1)$, $w(\tau)$ is the corresponding weight in the training schedule, ϵ is the added noise, $\mathbf{z}_\tau = (1-\tau)\mathbf{z} + \tau\epsilon$ denotes the noisy video, and $\hat{v}\left(\mathbf{z}_\tau; \tau, \mathbf{y}\right)$ is the predicted velocity.

A domain-specific noise sampling strategy is critical for this target to work well on our objective of optimizing scene deformations. We observed that the deformations are prone to be generated at higher noise levels τ, as significant changes only happen when substantial noise is added. Based on this observation and the properties of $w(\tau)$, instead of sampling τ uniformly, we perform sampling according to a probability density function $\hat{w}(\tau) = \frac{1}{\int_{-\infty}^{\infty} w(\tau)\,\mathrm{d}\tau} w(\tau)$, which is the normalized form of $w(\tau)$.

With this modification in sampling strategy, the weighted RFSDS update rule becomes:

$$\begin{aligned}&\nabla_\theta \mathcal{L}_{\text{W-RFSDS}}(\theta; z, \mathbf{y}) = \\ &\mathbb{E}_{\tau\sim\hat{w}(\tau),\epsilon}\left[\left(\hat{v}\left(\mathbf{z}_\tau; \tau, \mathbf{y}\right) - \epsilon + \mathbf{z}\right)\frac{\partial \mathbf{z}}{\partial \theta}\right],\end{aligned} \tag{3}$$

where the weighting term in RFSDS gradients defined in Eq. (2) is eliminated to ensures the invariance of the expec-

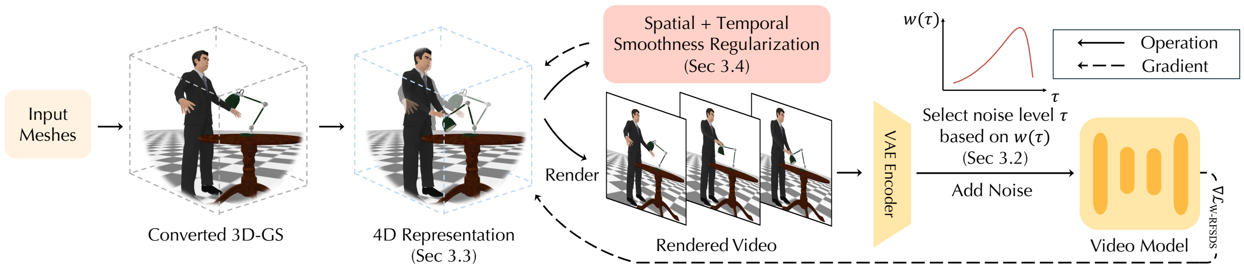

Figure 2. **Overview.** For the input meshes of a given scene, we first convert them into 3D-GS representations to enable smooth gradient computation. The converted 3D-GS models are then used to initialize a 4D representation (Sec. 3.3). We iteratively refine this 4D representation by sampling camera poses at each iteration, rendering the corresponding videos, and passing them to the video generation model to obtain optimization gradients (Sec. 3.2). Additionally, we compute regularization terms (Sec. 3.4) to enforce spatial and temporal smoothness during the optimization process.

tation of gradients. Empirically, this yields more realistic generated motion, as shown in Sec. 4.3.

Practically, this noise sampling strategy is implemented with an annealing noise schedule [20, 56] during the optimization. At each optimization step i out of entire I iterations, we set τ to be a fixed noise level τ_i, which is obtained by solving:

$$h(\tau_i) = 1 - \frac{i}{I+1}, \tag{4}$$

where $h(\tau) = \int_{-\infty}^{\tau} \hat{w}(t)\,\mathrm{d}t$ is the cumulative distribution function (CDF) of $\hat{w}(\tau)$. This creates an annealing schedule in which τ gradually decreases over training, enabling coarse motion to form early and allowing fine deformations to be refined in later iterations.

3.3. Hierarchical 4D Representation

Most existing 4D representations are highly unstable to optimize with the W-RFSDS target described above. Therefore, we introduce a hierarchical 4D representation that leverages natural locality of deformations in both spatial and temporal domain to stabilize the optimization process.

Our representation is composed of two components: a geometric representation of canonical shapes and a 4D motion representation that deforms the canonical geometry in different frames. The canonical shape of our 4D representation is represented with 3D-GS [25]. Specifically, given N mesh inputs, we convert them into 3D-GS models $\mathcal{S} = \{\mathcal{G}_i\}_{i=1}^{N}$ by optimizing directly on multi-view images rendered from the meshes, where each $\mathcal{G}_i$ denotes a converted 3D-GS model.

At time t, the canonical 3D geometry is deformed with a set of deformation fields to represent the 4D motion of the 3D-GS scene $\mathcal{S}$. The set of deformation fields at time t is denoted by $\mathcal{D}^t = \{\mathcal{T}_i^t\}_{i=1}^{N}$, where $\mathcal{T}_i^t$ denotes the deformation for object i at time t.

The 4D scene motion $\mathcal{D}$ is represented with a novel representation that injects hierarchical structures in both spatial and temporal domain, as detailed below.

Spatial Hierarchy with Control Points. The deformation fields $\mathcal{T}_i^t$ are spatially high-dimensional, and we reduce the dimensionality of this representation with a hierarchical control point-based representation.

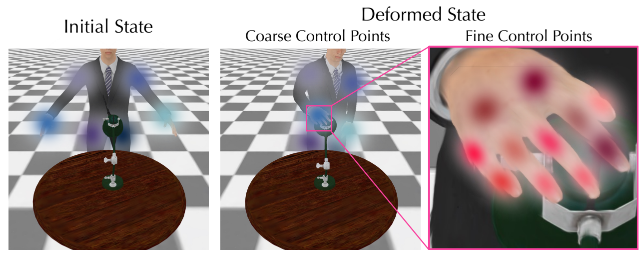

Figure 3. **Illustration of the hierarchical control point representation.** We represent the deformation using a spatial hierarchical structure. Coarse control points capture large-scale deformations, while fine control points refine local details.

Inspired by SC-GS [21], we represent $\mathcal{T}_i^t$ with a coarse level and a fine level of control points — a sparse set of spatially-grounded blobs that controls a local spatial region of deformations. The coarse level of control points roughly dictates how an object will deform, and the fine level adds more details to the deformation.

Specifically, each control point is defined by a mean $\boldsymbol{p}$ and a covariance matrix $\boldsymbol{\Sigma}$, which together determine its radius of influence. In addition, each control point maintains a sequence of deformations $(\mathbf{R}^t, \mathbf{T}^t)$ in $SE(3)$. The deformation of a Gaussian is obtained by blending transformations from neighboring control points using linear blend skinning. For a Gaussian $(\boldsymbol{\mu}, \boldsymbol{q}, \boldsymbol{S}, \mathcal{C}, o) \in \mathcal{G}_i$, we denote its K nearest neighboring control points as $\mathcal{N}$. The deformed Gaussian at time t is then computed as:

$$\mu^t = \sum_{k \in \mathcal{N}} \beta_k \left(R_k^t(\mu - \mathbf{p}_k) + \mathbf{p}_k + T_k^t \right), \tag{5}$$

$$\mathbf{q}^t = (\sum_{k \in \mathcal{N}} \beta_k r_k^t) \otimes \mathbf{q}, \tag{6}$$

where $r_k^t \in \mathbb{R}^4$ are the quaternion representations of rotation on control point k, and $\otimes$ is the production of quaternions. Furthermore, β_k in the formula denotes the blending

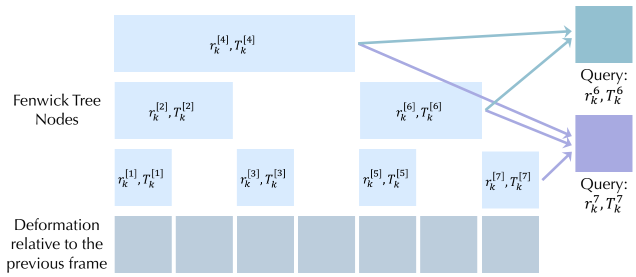

Figure 4. **Illustration of the Fenwick Tree representation.** Each node stores the cumulative deformation over a temporal range, allowing nearby frames to share parameters and enforcing temporal coherence. For example, $(r_k^{[6]}, T_k^{[6]})$ encodes the accumulated deformation from frames 5–6. Queries for frames 6 and 7 then compose their deformations from a small, overlapping set of nodes.

weight of control point k, which is calculated through:

$$\beta_k = \frac{\hat{\beta}_k}{\sum_{l \in \mathcal{N}} \hat{\beta}_l}, \; \hat{\beta}_k = \exp\left(-\frac{1}{2}\left[(\mu - p_k)\, \Sigma_k^{-1}\, (\mu - p_k)^T\right]\right). \tag{7}$$

We optimize the bi-level sets of control points in a coarse-to-fine manner, following the noise schedule defined in Eq. (4). When τ is large during the optimization process, substantial motion can be generated; however, the SDS gradients produced at such noise levels are often noisy. Conversely, when τ is annealed to a lower value, the gradients become more stable but are less capable of producing substantial deformations. To accompany with the inherent nature of this optimization process, we only optimize the coarse level of control points at earlier iterations when τ is large, and we introduce the fine control points later, once τ becomes smaller, to append their residual deformations:

$$\mu^t_{\text{final}} = \Delta\mu^t + \mu_t, \tag{8}$$

$$\mathbf{q}^t_{\text{final}} = \Delta\mathbf{q}^t \otimes \mathbf{q}^t \tag{9}$$

where $\Delta\boldsymbol{\mu}^t$ and $\Delta\mathbf{q}^t$ denote the residual deformations from the fine layer of control points, computed in the same manner as in Eq. (5) and Eq. (6).

After training, the deformation learned with Gaussians can be directly transferred to deform meshes. Concretely, we deform the mesh vertices using Eq. (5) by substituting the Gaussian means with the vertex positions.

Temporal Hierarchy with the Fenwick Tree. We further observe that deformations of later frames are challenging to learn if (R^t, T^t) of frame t are modeled independently from other frames. This can be explained by the fact that all deformations are initially initialized as zero vectors and the parameters of the first frame are kept frozen, leading to the significant deviation of deformations in later frames.

To alleviate this issue, we represent the sequence of deformations for each control point (R^t, T^t) with the Fenwick tree, a hierarchical data structure from theoretical algorithm design [13]. As illustrated in Figure 4, for each control point k, we maintain nodes $\mathcal{F}_k = \{(r_k^{[j]}, T_k^{[j]})\}_{j=1}^T$, where each node encodes the accumulated deformation over a specific range of frames. This range-based decomposition allows deformations at different frames to share parameters through overlapping intervals, greatly improving temporal coherence and enable the learning of long-horizon motion.

The final deformation at frame t is obtained by composing all relevant nodes:

$$T_k^t = \sum_{j \in \text{BIT}(t)} T_k^{[j]}, \tag{10}$$

$$r_k^t = \text{norm}(\sum_{j \in \text{BIT}(t)} r_k^{[j]}), \tag{11}$$

where $\text{BIT}(t)$ denotes the set of active nodes returned by the Fenwick query operation, and $\text{norm}(\cdot)$ ensures that the summed result forms a valid quaternion.

3.4. Regularization

We introduce two regularization terms to further stabilize the optimization process: a temporal regularization loss to enforce smoothness over time and a spatial regularization loss to encourage local spatial consistency.

Temporal Regularization. When rendering the RGB video for computing the SDS gradients, we additionally render a 3D flow map video $\mathbf{F}$ from the same viewpoint, which is used for temporal regularization. To produce the flow map at frame t, we replace the color attribute of Gaussians in the 3D-GS rendering equation with $\boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t+1}$, where $\boldsymbol{\mu}_i^t$ denotes the mean of Gaussian i at time t. After obtaining $\mathbf{F}$, the temporal regularization loss is defined as:

$$\mathcal{L}_{\text{temp}} = \sum_t \sum_{\mathbf{p}} ||F^t_{\mathbf{p}}||_2^2, \tag{12}$$

where the inner summation is over all pixels $\mathbf{p}$, and $\mathbf{F}^t_{\mathbf{p}}$ represents the rendered 3D flow at pixel $\mathbf{p}$ and time t.

Spatial Regularization. To ensure spatially uniform regularization, we generate a uniformly distributed point cloud near the surface of each object i, deform it using the learned motion, and compute an As-Rigid-As-Possible (ARAP) loss [52] over the resulting sequence of deformed point clouds. Specifically, we first compute a signed distance field (SDF) $\phi_i(\mathbf{x})$ from the mesh of object i. We then extract voxel centers near the surface as $\mathcal{S}_i = \{\mathbf{x} \mid |\phi_i(\mathbf{x})| \le \tau, \mathbf{x} \in V_s\}$, where V_s is the set of voxel centers on a grid with voxel size s, and τ is a predefined threshold. At each iteration, for every $\mathbf{x} \in \mathcal{S}_i$ and timestamp t, we compute its deformed position $\mathbf{x}^t$ using Eq. (5) (with μ replaced by $\mathbf{x}$), thereby producing the deformed point set $\mathcal{S}_i^t = \{\mathbf{x}^t \mid \mathbf{x} \in \mathcal{S}_i\}$. ARAP loss is then calculated as:

$$\mathcal{L}_{\text{ARAP}} = \sum_{i,t,\mathbf{x} \in \mathcal{S}_i, \mathbf{y} \in \mathcal{N}_{\mathbf{x}}} ||\mathbf{x} - \mathbf{y} - \hat{R}_{\mathbf{x}}(\mathbf{x}^t - \mathbf{y}^t)||_2^2, \tag{13}$$

where $\mathcal{N}_{\mathbf{x}}$ denotes the set of the 10 nearest neighbors of $\mathbf{x}$ in $\mathcal{S}_i$, and $\hat{R}_{\mathbf{x}}$ is the estimated local rotation matrix at $\mathbf{x}$.

4. Experiments

We evaluate our proposed method on a diverse dynamic scenes featuring multiple interacting objects. We compare our approach with several state-of-the-art baselines, each representing a distinct category of methods.

4.1. 4D Scene Motion Generation

We compare our method against state-of-the-art mesh animation approaches, as well as 4D reconstructions from camera-controlled video models. Specifically, we compare our approach with four baselines: Animate3D [24], AnimateAnyMesh [67], MotionDreamer [58], and TrajectoryCrafter [77]. Animate3D generates multi-view videos using a multi-view video diffusion model and then performs 4D reconstruction on them. AnimateAnyMesh directly predicts mesh deformations using a pretrained Rectified Flow model. MotionDreamer first generates a video conditioned on the text prompt and a rendering of the given mesh, and then animates the mesh by performing diffusion feature matching with the generated video. We present results from our reimplementation using Wan 2.2, and provide results obtained with DynamiCrafter [70] which was used in its original pipeline in the supplementary materials. TrajectoryCrafter is a video generation model that redirects camera trajectories for monocular videos. We first generate a video using Wan 2.2, then produce corresponding multi-view videos with TrajectoryCrafter, and finally perform 4D reconstruction on the sampled videos.

We select six scenes spanning diverse object categories for comparison: "A man petting a dog", "A cat stepping on a cushion", "A sealion nudging a ball", "A block falling on a trampoline", "Two men shaking hands", and "A robot picking up a block". We additionally include comparisons between our method and baseline approaches for **single-object mesh animation** in the supplementary materials.

Qualitative Comparisons. Part of the qualitative results are shown in Figure 5; please refer to the supplementary materials for the complete set of results. Our method exhibits stronger prompt alignment and generates more natural motion compared to existing approaches. Animate3D and AnimateAnyMesh fail to generate results that align with the given prompts, as they have not been extensively trained on 4D data containing multiple objects. MotionDreamer suffers from severe artifacts due to errors in diffusion feature matching when fitting meshes. Although 4D reconstruction from videos sampled via TrajectoryCrafter yields motions that follow the prompts, the results suffer from strong temporal inconsistencies and unnatural dynamics due to discrepancies among videos generated under different camera trajectories. This highlights the necessity of distilling a video model in our method.

Table 1. **Quantitative comparisons with baselines.** We conduct a user study on six scene animations to evaluate the performance. Additionally, we report the Semantic Adherence (SA) and Physical Commonsense (PC) metrics computed with VideoPhy-2 [3].

	User Study		VideoPhy-2	
	Alignment ↑	Realism ↑	SA ↑	PC ↑
Animate3D	0.34%	0.51%	3.83	3.42
AnimateAnyMesh	1.01%	0.51%	3.5	**4.5**
MotionDreamer (DC)	0.51%	0.84%	3.42	4.08
MotionDreamer (Wan)	0.84%	0.34%	3.5	3.83
TrajectoryCrafter	9.60%	10.44%	<u>4.17</u>	3.83
CHORD (Ours)	**87.71%**	**87.37%**	**4.33**	<u>4.25</u>

Quantitative Comparisons. We perform a user study with 99 participants to compare the quality of our method with the baselines. Additionally, we utilize VideoPhy-2 [3] to automatically evaluate the rendered videos from two aspects: Semantic Adherence (SA) and Physical Commonsense (PC). As shown in Table 1, our method achieves the highest score in SA and the second-highest score in PC. Note that AnimateAnyMesh achieves the highest Physical Commonsense (PC) score due to its common failure mode, where objects remain static—an outcome that aligns with physical commonsense but fails to follow the given prompt.

4.2. Extensions and Applications

Beyond generating multi-object 4D motion, our framework naturally supports several extension and downstream uses.

Long-Horizon Motion Generation. By using the last frame of the generated deformation as the input state for the subsequent generation process, we can extend our method to produce longer motion sequences. In Figure 1, we show an example motion sequence consisting of four actions.

Real-world Object Animation. Since our method distills a video generative model trained extensively on real-world video data, it is robust and can be applied to animate scanned real-world objects without concern for the gap between synthetic and real-world data, as shown in Figure 6.

Robot Manipulation. We demonstrate that the dense object flow generated by our method can be utilized as guidance for manipulation of rigid, articulated, and deformable objects, as shown in Figure 7. We first use an off-the-shelf grasp planner [12] to propose a grasp on the relevant object. Then, the robot either grasps the object or moves to a pose for pushing the object, which is at an offset from the proposed grasp. Constrained by a rigid attachment forward model, where relative transformations of the end-effector also apply to the initial points on the object, a motion planner [26] solves for a sequence of end-effector poses to minimize an objective consisting of transformed points to dense flow alignment, reachability, and pose smoothness costs.

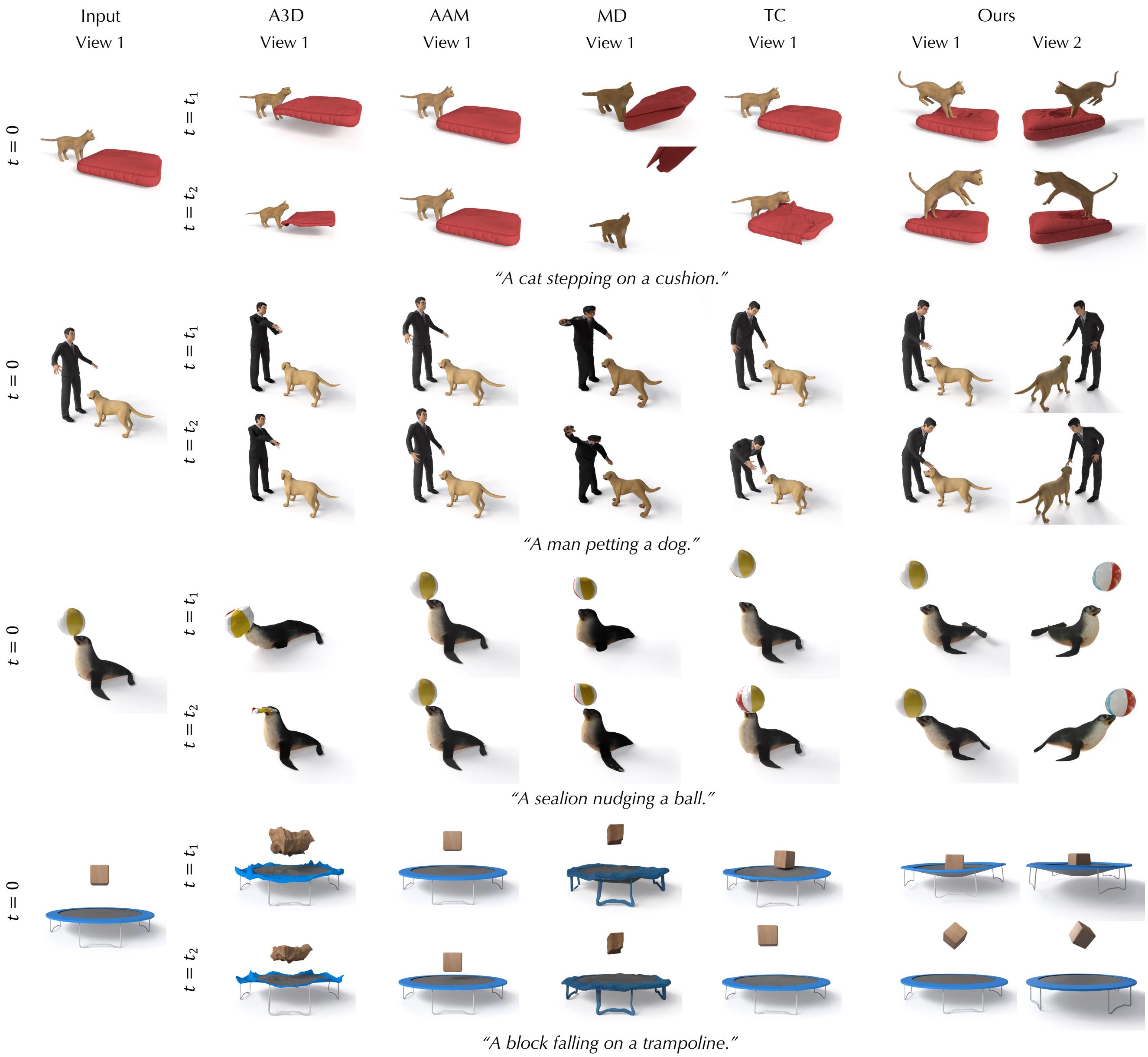

Figure 5. **Qualitative comparisons.** We compare our approach with several mesh animation methods. Our method produces results that better align with the given prompts and exhibit more natural motion. In the figure, A3D refers to Animate3D [24], AAM denotes AnimateAnyMesh [67], MD represents MotionDreamer [58], and TC corresponds to 4D reconstruction results from videos generated by TrajectoryCrafter [77]. For additional comparisons and full animation results, please refer to our supplementary website.

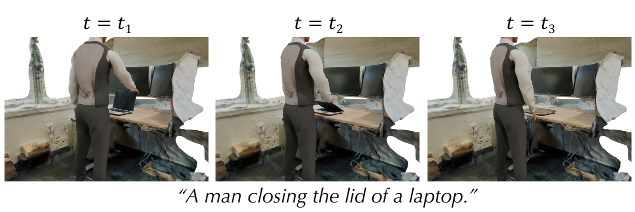

Figure 6. **Real-world object animation results.**

4.3. Ablation Studies

Noise Level Sampling Strategy We compare the effectiveness of the noise-level sampling strategy (Sec. 3.2) against uniform noise sampling with weighting. As shown in Figure 8, unrealistic results emerge under uniform sampling due to insufficient coverage of noise levels that inject motion. In this case, the laptop appears to float above the table.

4D Representation. We study two key components of our 4D representation: the Fenwick tree for modeling deformation sequences and the hierarchical control-point structure. Results are shown in Figure 9. Removing the Fenwick tree leads to noticeable artifacts, as later frames become extremely difficult to learn when each deformation is modeled independently. Removing the fine control-point layer

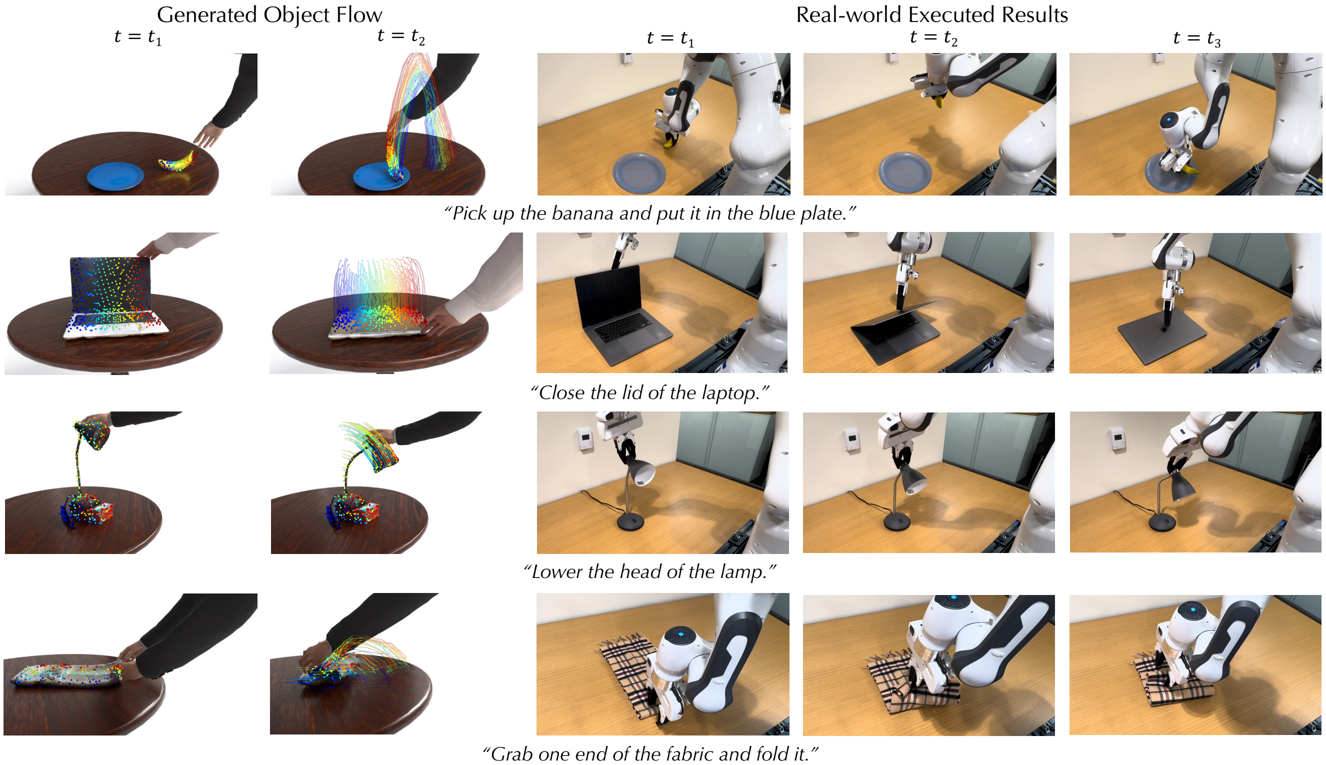

Figure 7. **Robot manipulation guided by our generated dense object flow.** Given our generated dense object flow, the robot either grasps or pushes the object of interest in a manner that matches the flow. This allows effective manipulation of rigid objects (first row), articulated objects (second row), and deformable objects (third and fourth rows).

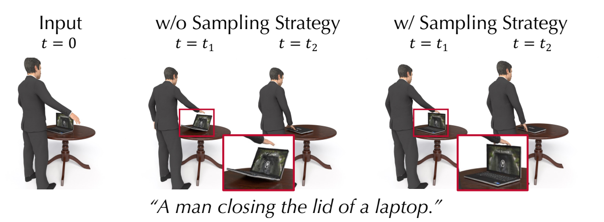

Figure 8. **Ablation on noise-level sampling strategy.** Removing our noise-level sampling strategy leads to unnatural motion, such as the laptop appearing to float.

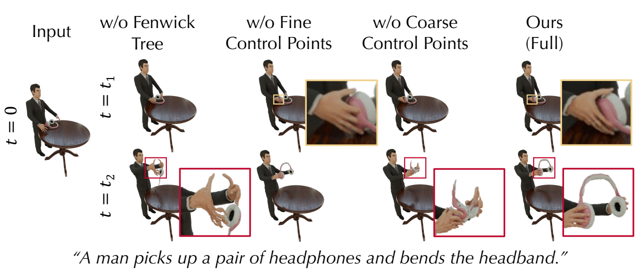

Figure 9. **Ablations on components in the 4D representation.** Removing the Fenwick Tree leads to severe artifacts in later frames; removing fine control points prevents detailed deformation; and removing coarse control points causes distortions.

prevents the model from producing detailed motions (e.g., grasping). Conversely, starting with the fine layer from the beginning also introduces artifacts, since the noise injected at early iterations cannot be effectively smoothed without an initial coarse stage.

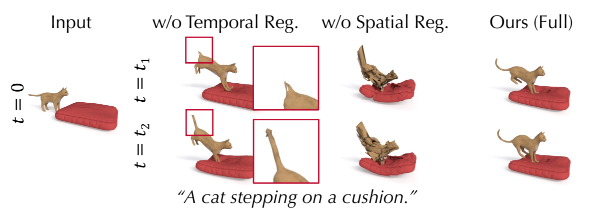

Figure 10. **Ablations on regularization losses.** Removing temporal regularization results in flickering, while removing spatial regularization results in distortions.

Regularization. We show that the regularization losses are necessary. Removing them results in temporal flickering (e.g., the tail suddenly appearing when temporal regularization is removed) and visual artifacts (when spatial regularization is removed), as shown in Figure 10.

5. Conclusion

We introduce a robust, scalable, and versatile approach to generate scene-level 4D object motion given only 3D shapes as input. Our pipeline works effectively for diverse natural phenomena and opens new possibilities of scalable 4D generation with guidance from video generative models. It also enables downstream applications as we demonstrated in the robotics manipulation.

Acknowledgments. This work is in part supported by NSF RI #2211258 and #2338203, ONR YIP N00014-24-1-2117, ONR MURI N00014-22-1-2740, the Stanford Institute for Human-Centered AI (HAI), and the Magic Grant from the Brown Institute for Media Innovation.

References

[1] 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, 2024. 3

[2] 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. 2

[3] Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. Videophy-2: A challenging action-centric physical commonsense evaluation in video generation. *arXiv preprint arXiv:2503.06800*, 2025. 6

[4] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074, 2003. 2

[5] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5543–5552, 2016. 2

[6] Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. Physgen3d: Crafting a miniature interactive world from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6178–6189, 2025. 3

[7] Wen-Hsuan Chu, Lei Ke, and Katerina Fragkiadaki. Dreamscene4d: Dynamic multi-object scene generation from monocular videos. *Advances in Neural Information Processing Systems*, 37:96181–96206, 2024. 3

[8] 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*, 36:35799–35813, 2023. 2

[9] 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. 2

[10] Yufan Deng, Yuhao Zhang, Chen Geng, Shangzhe Wu, and Jiajun Wu. Anymate: A dataset and baselines for learning 3d object rigging. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–10, 2025. 2

[11] E Knuth Donald et al. The art of computer programming. *Sorting and searching*, 3(426-458):4, 1999. 2

[12] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics (T-RO)*, 2023. 6

[13] Peter M Fenwick. A new data structure for cumulative frequency tables. *Software: Practice and experience*, 24(3): 327–336, 1994. 5

[14] 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. 2, 3

[15] Chen Geng, Sida Peng, Zhen Xu, Hujun Bao, and Xiaowei Zhou. Learning neural volumetric representations of dynamic humans in minutes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8770, 2023. 2

[16] Chen Geng, Yunzhi Zhang, Shangzhe Wu, and Jiajun Wu. Birth and death of a rose. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26102–26113, 2025. 2, 3

[17] Guangzhao He, Chen Geng, Shangzhe Wu, and Jiajun Wu. Category-agnostic neural object rigging. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22078–22088, 2025. 2, 3

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[19] Di Huang, Xiaopeng Ji, Xingyi He, Jiaming Sun, Tong He, Qing Shuai, Wanli Ouyang, and Xiaowei Zhou. Reconstructing hand-held objects from monocular video. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 3

[20] Yukun Huang, Jianan Wang, Yukai Shi, Boshi Tang, Xianbiao Qi, and Lei Zhang. Dreamtime: An improved optimization strategy for diffusion-guided 3d generation. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. 4

[21] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4220–4230, 2024. 4

[22] Zehuan Huang, Haoran Feng, Yangtian Sun, Yuanchen Guo, Yanpei Cao, and Lu Sheng. Animax: Animating the inanimate in 3d with joint video-pose diffusion models. *arXiv preprint arXiv:2506.19851*, 2025. 2

[23] Yanqin Jiang, Li Zhang, Jin Gao, Weimin Hu, and Yao Yao. Consistent4d: Consistent 360 {\deg} dynamic object generation from monocular video. *arXiv preprint arXiv:2311.02848*, 2023. 2

[24] Yanqin Jiang, Chaohui Yu, Chenjie Cao, Fan Wang, Weiming Hu, and Jin Gao. Animate3d: Animating any 3d model

with multi-view video diffusion. *Advances in Neural Information Processing Systems*, 37:125879–125906, 2024. 2, 6, 7

[25] 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. 4

[26] Chung Min Kim*, Brent Yi*, Hongsuk Choi, Yi Ma, Ken Goldberg, and Angjoo Kanazawa. Pyroki: A modular toolkit for robot kinematic optimization. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025. 6

[27] Sumith Kulal, Jiayuan Mao, Alex Aiken, and Jiajun Wu. Hierarchical motion understanding via motion programs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6568–6576, 2021. 2

[28] Jiahui Lei, Yijia Weng, Adam W Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6165–6177, 2025. 3

[29] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023. 1

[30] Hongjie Li, Hong-Xing Yu, Jiaman Li, and Jiajun Wu. Zerohsi: Zero-shot 4d human-scene interaction by video generation. *arXiv preprint arXiv:2412.18600*, 2024. 3

[31] Zhiqi Li, Yiming Chen, and Peidong Liu. Dreammesh4d: Video-to-4d generation with sparse-controlled gaussian-mesh hybrid representation. *Advances in Neural Information Processing Systems*, 37:21377–21400, 2024. 2

[32] Zizhang Li, Hong-Xing Yu, Wei Liu, Yin Yang, Charles Herrmann, Gordon Wetzstein, and Jiajun Wu. Wonderplay: Dynamic 3d scene generation from a single image and actions. *arXiv preprint arXiv:2505.18151*, 2025. 3

[33] Hanwen Liang, Yuyang Yin, Dejia Xu, Hanxue Liang, Zhangyang Wang, Konstantinos N Plataniotis, Yao Zhao, and Yunchao Wei. Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models. *arXiv preprint arXiv:2405.16645*, 2024. 2

[34] Yuchen Lin, Chenguo Lin, Jianjin Xu, and Yadong Mu. Omniphysgs: 3d constitutive gaussians for general physics-based dynamics generation. *arXiv preprint arXiv:2501.18982*, 2025. 3

[35] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM transactions on graphics (TOG)*, 40(6):1–16, 2021. 2

[36] Ruoshi Liu, Alper Canberk, Shuran Song, and Carl Vondrick. Differentiable robot rendering. *arXiv preprint arXiv:2410.13851*, 2024. 2

[37] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 2

[38] 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

[39] Ziqiao Ma, Xuweiyi Chen, Shoubin Yu, Sai Bi, Kai Zhang, Chen Ziwen, Sihan Xu, Jianing Yang, Zexiang Xu, Kalyan Sunkavalli, et al. 4d-lrm: Large space-time reconstruction model from and to any view at any time. *arXiv preprint arXiv:2506.18890*, 2025. 2

[40] 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

[41] Zhao Mandi, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. Dexmachina: Functional retargeting for bimanual dexterous manipulation. *arXiv preprint arXiv:2505.24853*, 2025. 1

[42] Linzhan Mou, Jiahui Lei, Chen Wang, Lingjie Liu, and Kostas Daniilidis. Dimo: Diverse 3d motion generation for arbitrary objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14357–14368, 2025. 2

[43] 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. 3

[44] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 2

[45] Sida Peng, Chen Geng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Implicit neural representations with structured latent codes for human body modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9895–9907, 2023. 2

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

[47] 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. 3

[48] Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023. 2

[49] Jiawei Ren, Cheng Xie, Ashkan Mirzaei, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, Huan Ling, et al. L4gm: Large 4d gaussian reconstruction model. *Advances in Neural Information Processing Systems*, 37:56828–56858, 2024. 2

[50] Qing Shuai, Chen Geng, Qi Fang, Sida Peng, Wenhao Shen, Xiaowei Zhou, and Hujun Bao. Novel view synthesis of human interactions from sparse multi-view videos. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 3

[51] Chaoyue Song, Xiu Li, Fan Yang, Zhongcong Xu, Jiacheng Wei, Fayao Liu, Jiashi Feng, Guosheng Lin, and Jianfeng Zhang. Puppeteer: Rig and animate your 3d models. *arXiv preprint arXiv:2508.10898*, 2025. 2

[52] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, pages 109–116, 2007. 5

[53] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *Advances in neural information processing systems*, 34:12278–12291, 2021. 2

[54] Qi Sun, Zhiyang Guo, Ziyu Wan, Jing Nathan Yan, Shengming Yin, Wengang Zhou, Jing Liao, and Houqiang Li. Eg4d: Explicit generation of 4d object without score distillation. *arXiv preprint arXiv:2405.18132*, 2024. 2

[55] Qi Sun, Can Wang, Jiaxiang Shang, Wensen Feng, and Jing Liao. Animus3d: Text-driven 3d animation via motion score distillation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–11, 2025. 2

[56] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *The Twelfth International Conference on Learning Representations(ICLR)*, 2024. 4

[57] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 2

[58] Lukas Uzolas, Elmar Eisemann, and Petr Kellnhofer. Motiondreamer: Exploring semantic video diffusion features for zero-shot 3d mesh animation. In *2025 International Conference on 3D Vision (3DV)*, pages 893–904. IEEE, 2025. 2, 6, 7

[59] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 3

[60] Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9660–9672, 2025. 3

[61] Yikai Wang, Xinzhou Wang, Zilong Chen, Zhengyi Wang, Fuchun Sun, and Jun Zhu. Vidu4d: Single generated video to high-fidelity 4d reconstruction with dynamic gaussian surfels. *Advances in Neural Information Processing Systems*, 37:131316–131343, 2024. 2

[62] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 606–617, 2023. 3

[63] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20310–20320, 2024. 3

[64] Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26057–26068, 2025. 3

[65] 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

[66] 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, 2024. 3

[67] Zijie Wu, Chaohui Yu, Fan Wang, and Xiang Bai. Animateanymesh: A feed-forward 4d foundation model for text-driven universal mesh animation. pages 13557–13568, 2025. 2, 6, 7

[68] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020. 1

[69] 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

[70] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pages 399–417. Springer, 2024. 6

[71] Zhen Xu, Sida Peng, Chen Geng, Linzhan Mou, Zihan Yan, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. Relightable and animatable neural avatar from sparse-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 990–1000, 2024. 2

[72] Zeyu Yang, Zijie Pan, Chun Gu, and Li Zhang. Diffusion^2: Dynamic 3d content generation via score composition of

video and multi-view diffusion models. *arXiv preprint arXiv:2404.02148*, 2024. 2

[73] Chun-Han Yao, Yiming Xie, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d 2.0: Enhancing spatio-temporal consistency in multi-view video diffusion for high-quality 4d generation. pages 13248–13258, 2025. 2

[74] Yufei Ye, Abhinav Gupta, Kris Kitani, and Shubham Tulsiani. G-hop: Generative hand-object prior for interaction reconstruction and grasp synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1911–1920, 2024. 3

[75] Jiraphon Yenphraphai, Ashkan Mirzaei, Jianqi Chen, Jiaxu Zou, Sergey Tulyakov, Raymond A Yeh, Peter Wonka, and Chaoyang Wang. Shapegen4d: Towards high quality 4d shape generation from videos. *arXiv preprint arXiv:2510.06208*, 2025. 2

[76] Heng Yu, Chaoyang Wang, Peiye Zhuang, Willi Menapace, Aliaksandr Siarohin, Junli Cao, László Jeni, Sergey Tulyakov, and Hsin-Ying Lee. 4real: Towards photorealistic 4d scene generation via video diffusion models. *Advances in Neural Information Processing Systems*, 37:45256–45280, 2024. 2

[77] Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 100–111, 2025. 6, 7

[78] Yu-Jie Yuan, Leif Kobbelt, Jiwen Liu, Yuan Zhang, Pengfei Wan, Yu-Kun Lai, and Lin Gao. 4dynamic: Text-to-4d generation with hybrid priors. *arXiv preprint arXiv:2407.12684*, 2024. 2

[79] Bowen Zhang, Sicheng Xu, Chuxin Wang, Jiaolong Yang, Feng Zhao, Dong Chen, and Baining Guo. Gaussian variation field diffusion for high-fidelity video-to-4d synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12502–12513, 2025. 2

[80] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, 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, 2024. 3

[81] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2