Gaussians-to-Life: Text-Driven Animation of 3D Gaussian Splatting Scenes

Thomas Wimmer[1,2] Michael Oechsle[3] Michael Niemeyer[3] Federico Tombari[1,3]
[1]Technical University of Munich [2]Max Planck Institute for Informatics [3]Google
https://wimmerth.github.io/gaussians2life.html

Figure 1. Our proposed method Gaussians2Life preserves high visual quality of scenes while animating them according to a text prompt. It significantly outperforms a baseline method crafted from DreamGaussian4D [38] and creates more realistic movements.

Abstract

State-of-the-art novel view synthesis methods achieve impressive results for multi-view captures of static 3D scenes. However, the reconstructed scenes still lack "liveliness," a key component for creating engaging 3D experiences. Recently, novel video diffusion models generate realistic videos with complex motion and enable animations of 2D images, however they cannot naively be used to animate 3D scenes as they lack multi-view consistency. To breathe life into the static world, we propose Gaussians2Life*, a method for animating parts of high-quality 3D scenes in a Gaussian Splatting representation. Our key idea is to leverage powerful video diffusion models as the generative component of our model and to combine these with a robust technique to lift 2D videos into meaningful 3D motion. We find that, in contrast to prior work, this enables realistic animations of complex, pre-existing 3D scenes and further enables the animation of a large variety of object classes, while related work is mostly focused on prior-based character animation, or single 3D objects. Our model enables the creation of consistent, immersive 3D experiences for arbitrary scenes.*

1. Introduction

Recent advances in 3D representation, like NeRF and 3D Gaussian Splatting (3DGS) [23, 30], have emerged as powerful tools for highly accurate and fast novel view rendering of static scenes. While these approaches enable immersive experiences with impressive quality, their static nature can result in a lack of dynamism and engagement. Breathing life into scenes and automatically generating animation of previously static 3D objects holds strong potential to create more engaging and realistic experiences.

In the 2D domain, video diffusion models have demonstrated significant capabilities in generating realistic animations in videos given input images or text prompts [7, 8, 16, 43]. However, recent advances to leverage such models for the creation of dynamic 3D content [3, 27, 38, 39, 44, 60] still lag behind the generative capabilities of video diffusion models, particularly animating existing 3D scenes appears to be underexplored and methods are limited to animate single assets [19, 38]. To this end, we investigate how realistic outputs from existing video diffusion models can help animating objects in static 3D scenes. More specifically, we aim to animate 3DGS scenes following a user-defined text prompt and a bounding box containing the target object. We identified two key challenges to solve this task. The first challenge is about how to generate multi-view consistent video guidance with a VDM for a static scene. Given valid video guidance, the remaining challenge is how to lift generated 2D videos into realistic and consistent 3D motions of the 3DGS primitives in the static scene without degrading the visual quality of the input scene.

In this work, we provide a method capable of addressing these challenges. By leveraging multi-view information in the video diffusion step, we can generate approximately multi-view consistent video guidance without the need for expensive fine-tuning of the diffusion model. Further, we provide an in-depth analysis of lifting 2D motion to 3D and propose a robust framework for animating 3DGS scenes by using video diffusion guidance in a pipeline that combines depth estimation and point tracking to generate 3D anchor trajectories. These are used to animate the static 3DGS scene in a multi-view consistent manner. In summary, our contributions are as follows:

- We introduce a novel method for animating 3DGS scenes given a text prompt and an object bounding box.
- Our approach interfaces existing open-source video diffusion models to generate multi-view consistent video guidance for a static scene and lifts the 2D video guidance to a realistic 3D motion for Gaussian primitives in the 3DGS scene.
- We provide an experimental evaluation on real-world scenes from the MipNeRF360 [6] and Instruct-NeRF2NeRF [13] datasets, where we utilize an adaptation of DreamGaussian4D [38] as a relevant baseline. Further, we provide an in-depth ablation study indicating the effectiveness of our architectural choices.

2. Related Work

Text-to-Video Generation The recent success of text-to-image diffusion models [15, 40] has increased interest in generative models for other data types, including videos. Common paradigms in the creation of video diffusion models are to build upon pre-trained 2D image generative models and to train additional components for modeling temporal relationships between generated video frames [7, 8, 16, 43]. Recent works have also proposed additional conditioning signals besides text, e.g., images or sparse manually defined motion [7, 51, 52, 58]. We employ video generative models in our approach to optimize dynamics in a given 3D scene. The resulting 4D scenes are naturally 3D-consistent and can be rendered in real-time from any viewpoint, a major advantage over 4D generative models in many use cases. Recent, concurrent works explored explicit camera control or multi-view generation for video diffusion models [4, 14, 24, 57]. We note that multi-view consistent outputs are not guaranteed for these models, as they are not based on an explicit 3D representation. Additionally, rendering novel views of the dynamic scene requires querying the diffusion model, which is costly and cannot be performed in real-time.

Dynamic Gaussian Splatting Kerbl et al. [23] proposed 3D Gaussian Splatting (3DGS), a method for novel view synthesis that bridges the gap between volumetric-rendering based implicit representations [30] and explicit 3D representations, like 3D meshes, by representing a 3D scene as a set of 3D Gaussians. The explicit nature of 3DGS allows for fast rasterization and more controllability in modeling.

Naturally, follow-up works also explored the reconstruction of dynamic 3D scenes [10, 18, 26, 29, 54, 59]. The most common strategy in this domain is the optimization of a canonical 3DGS representation alongside a neural deformation field that maps a 3D coordinate x and time t to attribute changes at the respective moment t for the 3D Gaussians at x in the canonical representation. Various regularization terms have been proposed to steer the optimized motion to be physically or geometrically plausible, i.e., to preserve local rigidity [18, 29], isometry [29], or momentum [11]. Gao et al. [12] further proposed using an off-the-shelf 2D flow estimation model as additional supervision signal for deformations. Recent concurrent works proposed methods for monocular dynamic scene reconstruction that make use of different pre-trained 2D models to obtain more information on the 3D structure [25, 45]. While we also make use of pre-trained 2D models to lift motion into 3D, we generate dynamics instead of reconstructing them. For this, we repeatedly generate new guidance videos and lift motion from different viewpoints.

4D Generative Models Poole et al. [35] proposed Score Distillation Sampling (SDS) to leverage 2D diffusion models as powerful priors for 3D generative tasks, which was followed by several technical improvements [50, 53, 61]. In SDS, 3D scenes are rendered at every optimization step, 2D views are noised, and one de-noising step with the 2D diffusion model is performed, which provides a gradient signal that can be used for back-propagation to the parameterized scene. An alternative to SDS is multi-step denoising [55, 63], which follows the same idea as SDS, but instead of doing one, multiple de-noising steps, as well as decoding from latent to pixel space are performed, similarly to the standard inference of a diffusion model. Losses are subsequently computed directly in pixel space.

Recently, several works started exploring the use of SDS for 4D generation [3, 27, 38, 39, 44, 60]. While some methods directly perform text-to-4d generation, others require image or video inputs which first need to be generated using off-the-shelf 2D diffusion models. Bahmani et al. [2] proposed trajectory-conditioned 4D generation, where coarse trajectories of objects are given as an additional input, which resolves the problem of limited motion in other 4D generation works. Except for 4Real [60], a concurrent work, all methods are restricted to single animated 3D objects, often lacking photo-realism. The main reason for this focus on object-centric generation is the usage of multi-view image diffusion models that are trained on datasets of single 3D objects and do not generalize to more complex 3D scenes [28, 42]. Instead of generating dynamic 3DGS scenes from only text, our goal is to generate dynamics for a given 3DGS scene and text as control. Another concurrent work, Animate3D [19], is the only work that we are aware of that aims at animating given 3D scenes. Contrary to our work, this method focuses on the animation of single 3D assets, while we aim at generating realistic deformations in the contexts of larger 3D scenes. Finally, another recent direction for animation of 3D scenes is the optimization of physical material fields [17, 62], which can then be used to perform physically realistic animations based on manual inputs and the PhysGaussian method [56]. We note that such methods usually require manipulation by external agents or external forces, e.g., using deformation handles, and are limited to a handful of simulation types [56]. In contrast, our proposed method is able to synthesize any type of deformation using only text as a control signal.

3. Method

A successful method for animating 3D scenes requires two main components: A powerful driving signal for motion generation and an effective way to distill this motion into the 3D scene while keeping the scene appearance and generating motion realistic. The diffusion model guidance should stay closely aligned with the given 3D scene, as well as be as multi-view consistent as possible, while the distillation of this signal into the 3D world should be as efficient as possible. In an optimal scenario, these two components will improve each other to achieve the best possible results. Our method offers improvements over standard SDS- and optimization-based solutions by introducing a training-free approach for generating approximately multi-view consistent outputs of a video diffusion model and a technique to directly lift 2D motion into 3D, leveraging several pretrained 2D models to align information between generated videos and the given 3D scene.

3.1. Basic Setup

We formalize our problem as follows: We are given a captured 3DGS scene as input, as well as a text prompt that describes the desired motion within the scene. Each Gaussian in the initial scene has a mean position $\mu \in \mathbb{R}^3$, an anisotropic covariance matrix, factorized into a scaling vector $s \in \mathbb{R}^3$ and a quaternion rotation $q \in \mathbb{R}^4$, as well as an opacity level α and view-dependent colors represented using spherical harmonics. In our work, we aim to add a temporal dimension to the given 3D scene. While we do not change the opacity of Gaussians, the position μ_t, scaling s_t, and rotation q_t should be time-dependent attributes, see Sec. 3.3.

For more user control, we allow for a user-defined selection of scene elements that should be animated. Such selection can stem from binary labeling of 3D Gaussians, which can be automated using open-world 3D segmentation methods [36] and is often assumed given in related works [29, 62], or from simple 3D bounding boxes.

Our method is agnostic to the specific 3DGS implementation used for capturing as long as the output follows the standard 3DGS conventions. During optimization, we only use the 0-degree spherical harmonics to save computational resources but note that at inference time, higher-degree spherical harmonics can be added back again while being rotated similarly as the corresponding 3D Gaussians, see Xie et al. [56].

3.2. Diffusion Guidance

In this section, we describe how we interface a recent video diffusion model for the purpose of generating dynamics for a given 3DGS scene.

Image-Conditioned Generation The use of an image-conditioned diffusion model [9] has proven to be beneficial for 3D scene editing [13]. As outputs are more aligned with the given 3D scenes through such conditioning, the noise level in SDS can be increased, resulting in larger amounts of motion compared to outputs of solely text-conditioned diffusion guidance, where the noise level needs to be reduced to stay aligned with the 3D scene. We thus employ a text- and image-conditioned video diffusion model, DynamiCrafter [58], as guidance in our method.

Multi-Step Denoising Score Distillation Sampling suffers from several technical issues, as pointed out by previous works [50, 53, 61]. Proposed solutions often require fine-tuning or training a second diffusion model, which is impractical for video diffusion models. Besides that, SDS imposes a computational burden where the diffusion model must be queried at each optimization step, and the loss computed in latent space needs to be back-propagated to the scene parameterization.

Instead, we propose using multi-step denoising (Sec. 2), which decouples generation from optimization, as pixel-level outputs (videos) can be stored and reused. This approach also enables the computation of additional supervision signals, like optical flow or depth, which is not possible with SDS. Moreover, pixel-level outputs improve user control during optimization, addressing the instability of current text-to-4D methods in public video diffusion models.

Multi-View Consistent Video Generation A remaining problem of current video diffusion models is their lack of output consistency. Especially when generating videos from different viewpoints, generated motion in the videos will often be inconsistent and thus hinder the optimization of 3D dynamics. For static 3D objects, fine-tuning image diffusion models for generating multi-view outputs is a standard solution for improving this multi-view consistency. This fine-tuning, however, is even more costly and difficult with video diffusion models, where multi-view data

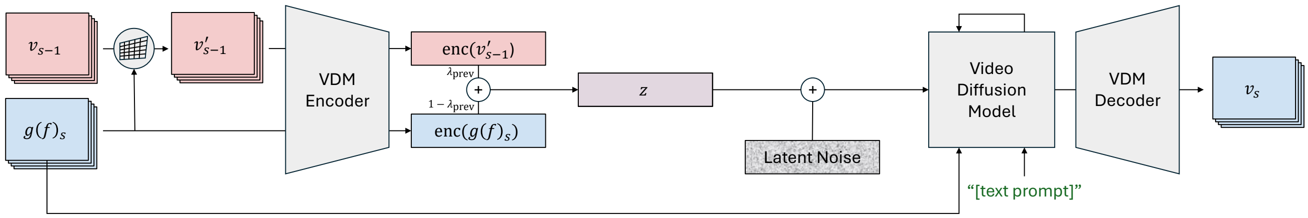


Figure 2. Improvement of multi-view consistency of generated videos through latent interpolation. In addition to the rendering of the dynamic scene f using the rendering function g from the current viewpoint $g(f)_s$, we compute the latent embedding of the warped video output v_{s-1} of the previous optimization step $s-1$ (from a different viewpoint). We linearly interpolate the latents before passing them through the video diffusion model (VDM), which is additionally conditioned on the static scene view from the current viewpoint. The resulting output is finally decoded to a new video output v_s.

is limited and fine-tuning on synthetic datasets, e.g., of single animated 3D objects, limits the generalizability of the video diffusion models [57].

Instead, we leverage the 3D information given through the static scene initialization. While using rendered views from the current viewpoint promotes 3D awareness of the diffusion model, this signal is only static. To steer generations to also contain consistent motion, we propose a new method of latent interpolation, where we encode the previously generated guidance video v_{s-1} and fuse its latent, which encodes the desired motion, with the latent of the current video rendering $g(f)_s$, as shown in Fig. 2:

$$z = \lambda_{\text{prev}}\, \text{enc}(v_{s-1}) + (1 - \lambda_{\text{prev}})\, \text{enc}(g(f)_s), \quad (1)$$

where z denotes the fused latent, $g(\cdot)$ the rendering function, f the 4D Gaussian Splatting scene, and λ_{prev} is a hyperparameter that is gradually decreased with increasing number of steps s. To resemble motion from the new viewpoint as realistically as possible, we make use of an off-the-shelf optical flow estimation model to warp the video frames of v_{s-1} as detailed in Sec. S1.2. While this warping can, of course, not give us a truly realistic projection of the motion to the current viewpoint, it helps in adapting the video to resemble the view from the new viewpoint when using small viewpoint changes. We adapt the view sampling procedure to reflect this assumption of small baseline changes, as described in Sec. S1.1.

We note that this proposed method puts a stronger emphasis on the initial diffusion video that is created for the first viewpoint. This fits well for the case where the user selects this first guidance video but can potentially cause problems when the pipeline is run without such a selection and the first generated video does not represent realistic motion. In such cases, the next generated outputs can still make up for any previous mistakes, but there is no guarantee of improvements. We include an exemplary multi-view generation of our proposed method in Fig. S6.

3.3. Lifting 2D Motion to 3D

Given valid video guidance, we now investigate the second core question: how can we lift 2D guidance signals efficiently to 3D? We first point out that optimization-based solutions, i.e., SDS or rendering-based optimizations, are slow in convergence and sensitive to the still-existing small inconsistencies in the generated guidance videos. Results are, therefore, often noisy and do either yield divergent motion or almost static outputs (see Ablation studies, Fig. 5). To evade these problems, we propose to instead leverage the power of several pre-trained 2D models to lift motion from 2D to 3D. More specifically, we combine 2D point tracking and depth estimation to obtain information on the depicted 3D motion from the videos, similar to concurrent works for monocular dynamic reconstruction [25, 45]. In our case, however, we use the 2D model outputs to efficiently bring dynamics into the 3D scene in just one step, compared to multiple steps necessary in optimization-based solutions [27, 38]). We repeat this procedure from other viewpoints to get more reliable and more 3D-aware estimations. We explain our method that is schematically visualized in Fig. 3 in the following.

We propose to track a sparse set of points throughout the generated video that is sampled from the video frame depicting the static scene rendering[1] (which we refer to as t_0). Recent works [22] have made remarkable progress in this task, being able to reliably track a set of points and even giving accurate estimations and recover points that are occluded. Further, we use an off-the-shelf metric depth estimation model [34] to obtain dense per-pixel depth estimations for every frame.

Tracking Correction Using 2D point tracks $(u, v)_t$ and per-frame dense depth maps, we compute the depth d_t for all tracked points at each timestep. Despite the robustness of the point-tracking method, errors can oc-

[1]The video diffusion model used in our method [58] was trained to always contain the image condition at one frame in the video output, which is often but not necessarily the first frame.

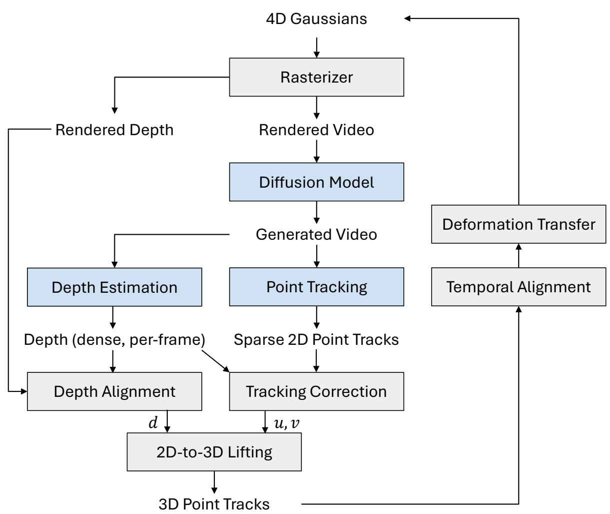


Figure 3. Pipeline for lifting 2D dynamics into 3D. Pre-trained models are shown in blue. We detect 2D point tracks and use aligned estimated depth values to lift them into 3D.

cur when the tracker "loses" a point and begins tracking another in the background or foreground. To address this, we use a correction method that detects errors by thresholding the depth value ratio between consecutive frames, where the threshold value is chosen manually to be $\max\{d_t, d_{t+1}\}/\min\{d_t, d_{t+1}\} < 1.2$ in our experiments. If the relative difference is large, we assume a tracking error and correct it by checking the local pixel neighborhood for a better tracking point, i.e., with a smaller depth ratio. If none is found, we discard the respective trajectory. We estimate depth values for points at frames where they are not visible with a cubic spline interpolation from known depth values. When extrapolation is needed, we use linear extrapolation based on the interpolation gradients at the last visible point.

Depth Alignment For every tracked point p_i, we compare the estimated depth d_i at the video frame at t_0 with the ground-truth depth value d_i^{GT} for the given scene and compute the ratio between them, which we then apply to the respective estimated depth values in other frames:

$$d'_{i,t} = d_{i,t}\frac{d_{i,t_0}}{d_i^{\text{GT}}}. \tag{2}$$

We note that as we sample the tracking points at the same frame, all tracked points are always visible in it.

2D-to-3D Lifting As camera extrinsics R, T and intrinsics K of the rasterization camera are known, we can project point trajectories given by pixel coordinates $(u, v)_t$ and estimated depth d_t back into the 3D world space X_t:

$$X_t = \begin{bmatrix} R\,T \\ 0\,1 \end{bmatrix}^{-1} \begin{bmatrix} K^{-1}\begin{bmatrix} u_t\ v_t\ d_t \end{bmatrix}^T\ 1 \end{bmatrix}^T \tag{3}$$

To reduce memory and computational demands, we only store the trajectories that lie within the 3D bounding boxes of the animated scene elements at t_0.

Temporal Alignment Knowing the timestep t_0, where the video shows the static scene, allows us to temporally align trajectories at this point, as shown in Fig. S1. During optimization, we use a rendered video as noised input to the diffusion model, which is fixed at generating n frames. As we wish to avoid interpolation, we select the sequence with the most overlaps to include maximum information from sampled viewpoints. If multiple sequences have the same "support," we select the last one, containing t_0 as early as possible. During testing, we can make use of linear interpolation between discrete timesteps to extend the frame count in the generated scene renderings.

Deformation Transfer Given the projected point trajectories, referred to as anchor trajectories in the following, we still need to transfer the motion onto single 3D Gaussians. At this point, the explicit nature of 3DGS comes in handy, as we can use deformation estimation methods that are inspired by techniques from traditional geometry processing (see Fig. 4). As we can directly infer motion that is, e.g., as rigid as possible while closely aligned with the anchor trajectories, this also is a decisive advantage over optimization-based solutions using such terms as regularization, where such terms hinder the amount of motion being distilled, as no motion is the most rigid motion possible.

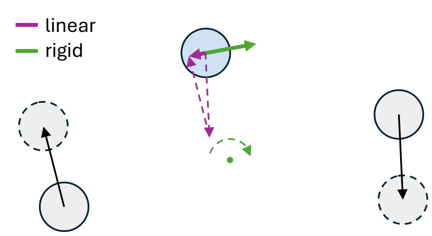


Figure 4. Comparison of linear and rigid motion estimation. The rigid motion estimation finds a fitting rotation for the source displacements and estimates the displacement for the target point accordingly.

Firstly, we propose to use a weighted linear motion estimation (cf. Linear Blend Skinning). To compute the weights, first, the K nearest anchor trajectories at t_0 and their distance d to the center of the Gaussian are determined for each 3D Gaussian. The displacement is then calculated as a weighted average of the neighboring anchor trajectory displacements $y_j - x_j$, where the weight depends on the distance d:

$$t_i = \sum_{j \in k\text{-NN}(i,K)} w_{i,j}(y_j - x_j), \tag{4}$$

$$\text{with} \quad w_{i,j} = \frac{\exp(-\tau d_{i,j})}{\sum_{j' \in k\text{-NN}(i,K)} \exp(-\tau d_{i,j})} \tag{5}$$

and a temperature parameter τ. To estimate rotation and scaling changes of 3D Gaussians, one can use a similar method as the subsequently proposed rigid motion estimation with fixed t.

A second, more sophisticated technique is the estimation of rigid body movements from the computed displacements of the anchor points. To do so, we make use of the Kabsch algorithm [21, 49] to estimate the optimal rotation, isotropic scaling and translation that align the point clouds of neighboring anchor trajectories at two subsequent timesteps, where we choose the weights $w_{i,j}$ as in Eq. 4:

$$\min_{R_i,t_i,s_i} \sum_{j \in k\text{-NN}(i,K)} w_{i,j} \left\| s_i R_i x_j + t_i - y_j \right\|^2 . \tag{6}$$

We note that though always considering the K nearest neighbors for every 3D Gaussian, motion is being refined when adding information from more guidance videos from different viewpoints.

4. Experiments

In this section, we provide experimental results of our method and qualitatively compare it to an adapted version of DreamGaussian4D [38]. Further, we demonstrate the effectiveness of our methodological choices in an in-depth ablation study. In our experiments, we focus on qualitative analysis due to the lack of established metrics for generative 3D dynamics (without ground truth deformations), as well as the lack of competing methods. However, we provide a section on possible metrics (and their shortcomings) as well as quantitative results for our ablation study in Sec. S2.1.

4.1. Setup

Dataset We select a number of scenes from the Mip-NeRF 360 [6] dataset, where we use RadSplat [32] for 3D reconstruction, as well as the bear scene from the Instruct-NeRF2NeRF [13] dataset, which we reconstruct with standard Gaussian Splatting [23].

Baseline Given that no prior work tackled the exact task of animating objects in the context of a full 3D scene, we carefully compose a baseline by adapting DreamGaussian4D [38] for the considered setting[2]. As this method was developed for single-object video-to-4D generation, we can also use it to demonstrate the strengths and importance of animating objects within a larger scene context. DreamGaussian4D works by first creating a static 3DGS model that is subsequently deformed following a given 2D video, as well as using SDS with a multi-view diffusion model [28]. For a fair comparison, we use the bounding boxes or masks that our method takes as input to mask out background elements for [38]. Further, we use the same initial diffusion guidance videos, where the background is automatically removed for the input video in DreamGaussian4D. We provide more details on the baseline, as well as the used hyperparameters in Secs. S1.4 and S1.5.

4.2. Qualitative Results

We show qualitative results[3] of our method applied to the different scenes along with the corresponding text prompts in Fig. 7. As can be seen, our method is able to generate compelling deformation while preserving the visual quality of the initial 3D scenes. We additionally show the optical flow between different frames, where we employ the same color coding as Baker et al. [5].

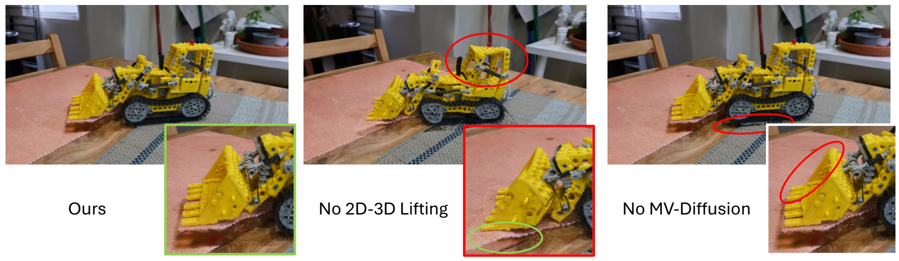
4.3. Baseline Comparison

We show exemplary comparisons of our method against the DreamGaussian4D baseline on real-world scenes in Fig. 1. We notice two things: First, the visual quality of the 3D scene is much more preserved by our method. This can be attributed to several factors. Our method is based on deforming the existing objects by inferring 3D motion directly from generated videos. The detected anchor trajectories are then used to estimate the transformation of the single 3D Gaussians. As the nearest neighboring anchor trajectories for 3D Gaussians that lie close to each other have significant overlap, shapes are automatically smoothly deformed. On the opposite, the baseline deforms the 3D Gaussians independently from each other, resulting in less coherent motion. The appearance-based optimization is also more prone to artifacts in the diffusion model guidance and, as the employed diffusion model is not image-conditioned, can also exhibit problems like the Janus problem which is not explicitly solved by the multi-view diffusion model, see the bear from the back side in Fig. 1.

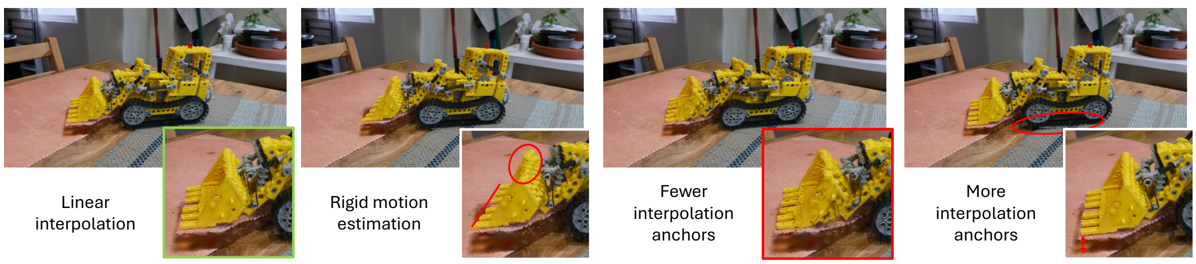
Next, we notice that our method is able to generate more realistic movements of the animated objects *within* the context of the larger scenes. As diffusion guidance is generated for the objects within the initial, larger scene instead of for only the single object, contact points and realistic motion within the scene are implicitly taken care of. The baseline that uses a multi-view diffusion model that was trained on single 3D assets is not able to model the motion within this scene context, leading to discontinuities with the background elements, e.g., the feet of the bear in Fig. 1.

[2] The concurrent method Animate3D [19] for text-driven animation of single objects has no public code at the time of submission.

[3] We strongly recommend the reader to check the project website for the videos corresponding to the respective figures.



(a) Comparison against ablated versions.



(b) Motion transfer from anchor trajectories to 3D Gaussians.

Figure 5. Qualitative comparison against ablations on the LEGO bulldozer scene for the prompt "toy bulldozer lifting its shovel."

4.4. Ablation studies

In our ablation study, we first perform a comparison against an optimization-based approach, a commonly used technique for modeling dynamic scenes where a neural field is trained to model the deformations of the scene [27, 38], guided by our proposed diffusion guidance. More details on this baseline model are provided in Sec. S1.3. To show the effects of our proposed approximately multi-view consistent diffusion, we further ablate a version of our method using standard novel-view video generation without the proposed latent interpolation. Finally, we compare rigid and linear motion estimation and analyze the effect of the number of anchor trajectories considered in motion transfer. The qualitative results of this analysis can be found in Fig. 5.

While our method effectively deforms 3D scenes by lifting 2D motion to 3D and transferring it to 3D Gaussians, rendering-based optimization of a deformation field fails to generate coherent and realistic motion. The baseline particularly struggles with the temporal consistency of the generated motion due to the lack of robustness of rendering-based optimizations to inconsistent input data (see, e.g., static 3D Gaussian Splatting reconstruction of moving objects as in Wu et al. [54], Fig. 5). Although our proposed approximately multi-view consistent video generation mitigates this issue to some extent, the remaining inconsistencies still pose a significant challenge for optimizing meaningful motion. On the other hand, appearance-based optimization is able to "fill" holes in the 3D scene by moving or scaling Gaussians accordingly, which is an advantage over our purely deformation-based approach. The use of our proposed approximately multi-view diffusion helps increase 3D consistency in the results and especially reduces diffusion artifacts, which can lead to noisy anchor trajectories.

When transferring motion from anchor trajectories to 3D Gaussians, linear motion estimation surprisingly often results in more rigid deformation than rigid motion estimation, as seen at the back ends of the shovel in Fig. 5. The rigid motion estimation struggles with unbalanced observations, e.g., in the first optimization step. Specifically, when most registered (visible) anchor trajectories are on one side of the object, while some static observed points lie behind the Gaussians in question, as simplified in Fig. 6, where the query point (blue) is influenced by two moving source points and one static point, the rigid motion estimation predicts an incorrect rotation of the object. Subsequent optimization steps from other viewpoints fail to correct this error as noise is gradually removed from scene renderings, causing the diffusion model to adapt to the faulty movement.

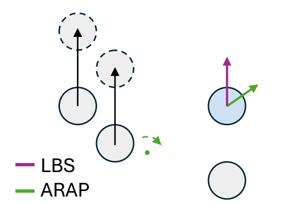


Figure 6. Linear vs. rigid motion estimation with limited observations.

5. Conclusions

We presented Gaussians2Life, a technique for text-driven animation of static 3D Gaussian Splatting scenes based on video diffusion models. We addressed shortcomings of current public video diffusion models to improve 3D consistency in the outputs and enhance motion distillation from 2D into static 3D scenes while keeping the initial scene appearance intact through geometry-aware deformation trans-

Figure 7. Qualitative results of our method on different 3D scenes with the optical flow (see Fig. S4) shown between keyframes.

fer techniques. We emphasize that our proposed method is optimization-free and thus significantly faster than a baseline using an inference-time optimization. The high-quality results of applying our method to diverse real-world captures underline its strength in the faithful animation of given 3D scenes.

Limitations The proposed method deforms an existing scene without adding or removing 3D Gaussians. Thus, it is not possible to fill holes created by moving objects (see Fig. 5) or to add and remove particles, e.g., fire, in an animation. As our method does not include any rendering-based optimization, eventual artifacts through moving Gaussians are not made up for. A possible fix to this problem is employing SDS to refine the generated motion in an appended stage, e.g., as proposed by Bahmani et al. [2]. Object intersections or collisions are not handled explicitly but implicitly by using a generalist video diffusion model that generates videos of objects moving in the full scenes. While this generally works, making up for wrong depth or tracking estimations is not possible.

Finally, we note that there exists a domain mismatch between the often inherently static 3D scenes captured using 3D reconstruction methods and the dynamic scenes used in the training of video diffusion models. While this mismatch is limiting the effectiveness of diffusion model guidance, we introduced several techniques to deal with noisy predictions in this work. Generally, we observe that with currently available "open" video diffusion models, often, a realistic *actor* is still necessary for animating scenes, which can also be invisible, such as "the wind."

Acknowledgements We thank Daniel Cremers, whose computing resources were used for some of the experiments in the context of this work. TW is supported by the Konrad Zuse School ELIZA, sponsored by the German Federal Ministry of Education and Research.

References

[1] Rameen Abdal, Wang Yifan, Zifan Shi, Yinghao Xu, Ryan Po, Zhengfei Kuang, Qifeng Chen, Dit-Yan Yeung, and Gordon Wetzstein. Gaussian shell maps for efficient 3d human generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 9441–9451. IEEE, 2024. 4

[2] Sherwin Bahmani, Xian Liu, Wang Yifan, Ivan Skorokhodov, Victor Rong, Ziwei Liu, Xihui Liu, Jeong Joon Park, Sergey Tulyakov, Gordon Wetzstein, Andrea Tagliasacchi, and David B. Lindell. *TC4D: Trajectory-Conditioned Text-to-4D Generation*, page 53–72. Springer Nature Switzerland, 2024. 2, 8

[3] 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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 7996–8006. IEEE, 2024. 1, 2

[4] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. Vd3d: Taming large video diffusion transformers for 3d camera control. *arXiv preprint arXiv:2407.12781*, 2024. 2

[5] Simon Baker, Stefan Roth, Daniel Scharstein, Michael J. Black, J.P. Lewis, and Richard Szeliski. A database and evaluation methodology for optical flow. In *2007 IEEE 11th International Conference on Computer Vision*, page 1–8. IEEE, 2007. 6, 2

[6] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022. 2, 6

[7] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1, 2

[8] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 22563–22575. IEEE, 2023. 1, 2

[9] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 18392–18402. IEEE, 2023. 3

[10] Devikalyan Das, Christopher Wewer, Raza Yunus, Eddy Ilg, and Jan Eric Lenssen. Neural parametric gaussians for monocular non-rigid object reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 10715–10725. IEEE, 2024. 2

[11] Bardienus P Duisterhof, Zhao Mandi, Yunchao Yao, Jia-Wei Liu, Mike Zheng Shou, Shuran Song, and Jeffrey Ichnowski. MD-splatting: Learning metric deformation from 4d gaussians in highly deformable scenes. *arXiv preprint arXiv:2312.00583*, 2023. 2, 3

[12] 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

[13] Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 19683–19693. IEEE, 2023. 2, 3, 6

[14] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for text-to-video generation, 2024. 2

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[16] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 1, 2

[17] Tianyu Huang, Yihan Zeng, Hui Li, Wangmeng Zuo, and Rynson WH Lau. DreamPhysics: Learning physical properties of dynamic 3d gaussians with video diffusion priors. *arXiv preprint arXiv:2406.01476*, 2024. 3

[18] 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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 4220–4230. IEEE, 2024. 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. 1, 3, 6

[20] HyunJun Jung, Nikolas Brasch, Jifei Song, Eduardo Perez-Pellitero, Yiren Zhou, Zhihao Li, Nassir Navab, and Benjamin Busam. Deformable 3d gaussian splatting for animatable human avatars. *arXiv preprint arXiv:2312.15059*, 2023. 4

[21] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5): 922–923, 1976. 6

[22] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, 2024. 4

[23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4):1–14, 2023. 1, 2, 6

[24] Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video genera-

tion with camera control. *arXiv preprint arXiv:2405.17414*, 2024. 2

[25] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. MoSca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024. 2, 4

[26] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. *arXiv preprint arXiv:2312.03431*, 2023. 2

[27] 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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 8576–8588. IEEE, 2024. 1, 2, 4, 7

[28] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 9264–9275. IEEE, 2023. 2, 6

[29] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, page 800–809. IEEE, 2024. 2, 3

[30] 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*, page 405–421. Springer International Publishing, 2020. 1, 2

[31] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41 (4):1–15, 2022. 2

[32] Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Michael Oechsle, Daniel Duckworth, Rama Gosula, Keisuke Tateno, John Bates, Dominik Kaeser, and Federico Tombari. RadSplat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ FPS. *arXiv preprint arXiv:2403.13806*, 2024. 6

[33] Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. ASH: Animatable gaussian splats for efficient and photoreal human rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 4

[34] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 10106–10116. IEEE, 2024. 4

[35] 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*, 2022. 2

[36] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3d language gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 20051–20060. IEEE, 2024. 3, 4

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[38] 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. 1, 2, 4, 6, 7

[39] 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. 1, 2

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 10674–10685. IEEE, 2022. 2

[41] 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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1606–1616. IEEE, 2024. 4

[42] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. In *The Twelfth International Conference on Learning Representations*, 2023. 2

[43] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 1, 2

[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. In *Proceedings of the 40th International Conference on Machine Learning*, pages 31915–31929, 2023. 1, 2

[45] Colton Stearns, Adam Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. *arXiv preprint arXiv:2406.18717*, 2024. 2, 4

[46] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 2

[47] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024. 2

[48] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow (extended abstract). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, page 4839–4843. International Joint Conferences on Artificial Intelligence Organization, 2021. 1

[49] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. 6

[50] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 12619–12629. IEEE, 2023. 2, 3

[51] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 2

[52] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. VideoComposer: Compositional video synthesis with motion controllability. *arXiv e-prints*, pages arXiv–2306, 2023. 2

[53] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[54] 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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 20310–20320. IEEE, 2024. 2, 7

[55] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Hołyński. ReconFusion: 3d reconstruction with diffusion priors. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 21551–21561. IEEE, 2024. 2

[56] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. PhysGaussian: Physics-integrated 3d gaussians for generative dynamics. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 4389–4398. IEEE, 2024. 3

[57] 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, 4

[58] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Xintao Wang, Tien-Tsin Wong, and Ying Shan. DynamiCrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, 2024. 2, 3, 4

[59] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 20331–20341. IEEE, 2024. 2

[60] Heng Yu, Chaoyang Wang, Peiye Zhuang, Willi Menapace, Aliaksandr Siarohin, Junli Cao, Laszlo A Jeni, Sergey Tulyakov, and Hsin-Ying Lee. 4real: Towards photorealistic 4d scene generation via video diffusion models. *arXiv preprint arXiv:2406.07472*, 2024. 1, 2

[61] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. In *The Thirteenth International Conference on Learning Representations*, 2024. 2, 3

[62] 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. *arXiv preprint arXiv:2404.13026*, 2024. 3

[63] Zhizhuo Zhou and Shubham Tulsiani. SparseFusion: Distilling view-conditioned diffusion for 3d reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 12588–12597. IEEE, 2023. 2