Drag4D: Align Your Motion with Text-Driven 3D Scene Generation

Minjun Kang[1*]   Inkyu Shin[2*†]   Taeyeop Lee[1]   In So Kweon[1]   Kuk-Jin Yoon[1]

[1]KAIST   [2]ByteDance Seed

A smoky grey kitchen with modern accents, small garden-facing windows, Bauhaus furniture, high ceilings, and a pastel beige-blue-salmon palette. Cozy atmosphere with a basket of produce, water bottle, magazine-style wall decor, and wooden parquet flooring.

An elegant dining room featuring a long polished table set with fine china, crystal glasses, and a chandelier, with ornate chairs and side tables adorned with decorative items.

Figure 1. We propose Drag4D, a comprehensive user-interactive framework for 4D-controllable video generation, designed to achieve spatial and temporal alignment of a target instance within a text-driven 3D background. For example, this framework allows users to create a high-quality 3D scene from a text description (middle section), seamlessly integrate target instances (left section), and precisely control motion following a user-defined 3D trajectory (right section).

Abstract

We introduce **Drag4D**, an interactive framework that integrates object motion control within text-driven 3D scene generation. This framework enables user to define 3D trajectory for the 360° objects generated from a single image, seamlessly integrating them into a high-quality 3D background. Our Drag4D pipeline consists of three stages. First, we enhance text-to-3D background generation by applying 2D Gaussian Splatting with panoramic images and inpainted novel views, resulting in dense and visually complete 3D reconstructions. In the second stage, given a reference image of the target object, we introduce a 3D copy-and-paste approach: the target instance is extracted in a full 360° representation using an off-

---

* Equal contribution.

† This work was partially conducted while at KAIST.

the-shelf image-to-3D model and seamlessly composited into the generated 3D scene. The object mesh is then positioned within the 3D scene via our physics-aware object position learning, ensuring precise spatial alignment. Lastly, the spatially aligned object is temporally animated along a user-defined 3D trajectory. To mitigate motion hallucination and ensure view-consistent temporal alignment, we develop a part-augmented, motion-conditioned video diffusion model that processes multiview image pairs together with their projected 2D trajectories. We demonstrate the effectiveness of our unified architecture through evaluations at each stage and in the final results, showcasing the harmonized alignment of user-controlled object motion within high-quality 3D background.

1. Introduction

The importance of user experience in computer vision has grown significantly. It has been driven by the rapid advancement of applications in various domains, specifically in VR/AR [33]. Enhancing user experience lies the ability to control and manipulate digital environments intuitively and interactively. One area where this control is particularly transformative is in video content, where users seek to generate their own videos with text descriptions [3, 17, 21, 29]. Due to the ambiguity of user-intention in text prompt, video model with user-given trajectory [16, 48] has recently emerged to allow users to effectively direct and modify the objects within videos. The another area is 3D content generation [11, 20, 30, 39, 57, 58]. It can construct 3D space, enabling users to engage with generating objects [30, 39] or/and scenes [11, 20, 57] in a fully immersive manner with unconstrained multiple camera views. The module designed for controllable video generation with trajectory and text-to-3D generation are not only evolving over time, but also becoming very distinct. Consequently, it is infeasible to easily adapt temporal controllability to text-to-3D (and vice versa). Particularly, current 2D trajectory-based video generation [16, 48] lacks the capability to scale up to multi-views, while text-to-3D method [5, 20, 58] can suffer from misalign-

Table 1. Our approach, Drag4D, seamlessly unifies three key components required for 4D controllable video generation: (1) generating a 3D scene from a text prompt, (2) composing objects into the generated 3D background, and (3) conditioning video motion to manipulate an object's trajectory naturally within the 3D space. No prior work has achieved this level of integration.

Method	3D Scene Generation	Object Composition	Motion Conditioned Video
LucidDreamer [5]	O	X	X
SceneDreamer360 [20]	O	X	X
Layout3D [58]	X	O	X
DragAnything [48]	X	X	O (2D Video)
Drag4D	O	O	O (4D Video)

ment of user-given moving objects with generated 3D scenes as shown in Tab. 1. The need for this scenario-specific design results in degrading the quality of user experience. A natural question thus emerges: *Is it feasible to develop a unified framework capable of achieving controllable 4D environments, where user can manipulate the trajectory of 360° object while synthesizing 3D background?*

To answer the question, we present a unified user-interactive framework, **Drag4D**, which aims to align your motions with text-to-3D generation. Our pipeline incorporates three key stages. First, in order to construct a basis for 3D background scene, we employ off-the-shelf text-driven panoramic image generation model [53]. Then, we extract depth and normal information from the panoramic image using a depth estimator [14], providing essential priors (point cloud) for 3D scene reconstruction. In contrast to SceneDreamer360 [20, 39], which directly augments training set with novel view images projected from point cloud, our approach integrates image inpainting model [6] to refine these novel views like HoloDreamer [57]. This inpainting process effectively addresses occlusions from different camera views, seamlessly filling in missing regions to enhance the visual coherence in the reconstructed 3D scene. Our framework departs from HoloDreamer [57] by employing efficient 2D Gaussian Splatting [15] to be optimized jointly with panoramic image set and inpainted novel view images. Specifically, we enhance the joint learning by proposing a pixel-level adaptive weighting mechanism using depth-normal similarity, which can mitigate the influence of noisy areas in aug-

mented views. Secondly, to extract the instance from user-provided reference image and composite it into the generated 3D background, we propose a 3D copy-and-paste approach. We employ an image-to-3D model [49] to scale up the area of foreground mask to 360°object ("3D copy"). Then, it is spatially aligned with the surrounding 3D background through physics-aware object position learning ("3D paste"), which can be implemented via collision and gravity loss. In the final stage, we input multiple views of the spatially composited scene-object and the corresponding trajectories (derived from a user-provided 3D path) into motion-conditioned video diffusion model. Unlike a previous model [48], which controls the motion of an entire instance with a single global feature, we introduce a part-augmented motion-conditioned video generator where local feature is co-utilized with global feature. We coin this approach, **L**ocal-**G**lobal DragAnything. We observe that this approach effectively prevents local motion hallucinations within the target instance, which is essential for precise motion alignment. We rigorously validate our design choices and methodology through comprehensive experiments presented in this paper. Given the limited availability of datasets specifically targeting moving objects within 3D scenes, we introduce a custom dataset, Drag4D-30, to showcase the enhanced performance of our approach compared to baseline methods.

2. Method

The meta architecture of Drag4D aims to design 4D controllable video generation, which align and manipulate object within 3D scene background generation. This process unfolds across three seamlessly integrated stages. The first stage, detailed in [Sec. 2.3], generates a 3D background scene by transforming given text prompt into high-quality panoramic image. In the second stage, described in [Sec. 2.4], a user-defined object is extracted from a reference view in a 360°manner and spatially composited into the generated 3D background. Finally, the third stage, covered in [Sec. 2.5], temporally aligns the user-specified 3D trajectory with the object, maintaining seamless composition with the surrounding background.

2.1. Preliminaries

Diffusion Models Diffusion probabilistic models (DPMs), first introduced by [41] and further refined by [10], constitute a type of generative model that reconstructs a target data distribution, denoted as q, through a staged denoising process. The process begins with an image x_T that is initially Gaussian-distributed as $x_T \sim \mathcal{N}(0, I)$. Containing independent and identically distributed noise. The diffusion model, represented by ϵ_θ , then progressively reduces this noise, transforming the image step-by-step until it arrives at a clean version, x_0 drawn from the target distribution q.

3D Gaussian Splatting (3D-GS) Kerbl et al. [18] introduce a method for representing 3D scenes using 3D Gaussian primitives and rendering images through differentiable volume splatting. In this approach, 3D-GS explicitly defines Gaussian primitives by specifying their 3D covariance matrix Σ and spatial location p_k:

$$\mathcal{G}(\mathbf{p}) = \exp\left(-\frac{1}{2}(\mathbf{p} - \mathbf{p}_k)^\top \Sigma^{-1}(\mathbf{p} - \mathbf{p}_k)\right) \quad (1)$$

Here, the covariance matrix Σ is decomposed into a scaling matrix S and a rotation matrix R, such that $\Sigma = RSS^\top R^\top$. To render an image, the 3D Gaussian is transformed to the camera's coordinate system using a world-to-camera transformation matrix W, and then projected onto the image plane via a local affine transformation. This results in a modified covariance matrix:

$$\Sigma' = JW\Sigma W^\top J^\top \quad (2)$$

2.2. Problem Setting

Our pipeline utilizes multiple user prompts across different stages to fully capture and reflect user's intentions. In the first stage, a detailed and long text prompt t is provided to generate high-quality 3D background scene. In the subsequent stage, a single reference image, x_r with an foreground mask (m_r) is supplied to extract the target object,
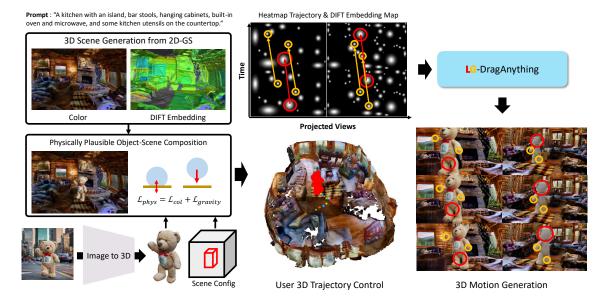
Prompt : "A kitchen with an island, bar stools, hanging cabinets, built-in oven and microwave, and some kitchen utensils on the countertop."

3D Scene Generation from 2D-GS

Color

DIFT Embedding

Physically Plausible Object-Scene Composition

$$\mathcal{L}_{phys} = \mathcal{L}_{col} + \mathcal{L}_{gravity}$$

Image to 3D

Scene Config

Heatmap Trajectory & DIFT Embedding Map

Time

Projected Views

LG-DragAnything

User 3D Trajectory Control

3D Motion Generation

Figure 2. The proposed Drag4D comprises three key stages. First, we conduct text-to-3D scene generation. Here, we use 2D Gaussian Splatting to process panoramic and inpainted augmented view images, generating high-quality 3D scenes with diffuion features (DIFT). Next, given reference image with target instance, a 360°object mesh is extracted from a reference image using an Image-to-3D model and composited into the 3D background based on scene configuration (e.g., 3D bounding boxes). A physics-aware object-scene composition method including collision loss and gravity loss ensures accurate spatial alignment of the target instance. In final stage, with the composited 3D scene and a user-defined 3D trajectory, the LG-DragAnything motion-conditioned video model enables view-consistent multi-view video generation, achieving high-quality 3D motion alignment.

which is then transformed into 360°object. Afterwards, it is scaled and positioned within generated 3D background scene according to a user-defined 3D bounding box configuration, B_{3D} , which specifies center coordinates, dimensions and rotation. Finally, user can manipulate the object within the 3D space using a sequence of 3D trajectory points, $(p_x, p_y, p_z)_{i=1...N}$, where N denotes the number of points defining the trajectory path.

2.3. 1st Stage: Generating Text-to-3D Scene

Panoramic Image Generation for 3D Scene The objective of the panoramic image generator is to construct high-quality 360-degree scenes guided by text input t, serving as a critical prior for reconstructing 3D backgrounds. To this end, we employ a diffusion model [53] for generat-

ing panoramic images. To effectively handle extended text prompts and generate high-resolution panoramic images, we incorporate LoRA [13] fine-tuning and super-resolution [46] techniques, following SceneDreamer360 [20]. Then, we can obtain the panoramic image I_p along with its corresponding depth D_p using a pre-trained metric depth estimator, Metric3D [51]. Given panoramic image I_p with corresponding depth D_p, we can obtain point cloud, P, utilizing [1]inverse equirectangular projection, E^{-1} as below:

$$P = E^{-1}(I_p, D_p) \tag{3}$$

, which serves as the initialization for reconstructing a dense 3D background. To obtain a series

---

[1]Mapping function to transform 2D pixel coordinates of panorama image into 3D coordinates.

4

of perspective images used as supervision for 3D reconstruction, we first can derive a high-quality set of base images by projecting from point cloud P using cameras positioned at the center of the panoramic sphere. Specifically, using shared intrinsic K and multiple center-positioned extrinsics E_i, base images can be obtained with following equation:

$$I_i = \Phi(P, K, E_i) \quad (4)$$

Let Φ denote the projection function from 3D point cloud to the corresponding 2D pixel coordinate.

Image Inpainting for Novel Views However, 3D reconstruction with the supervision of the base images I_i may result in poor rendering quality due to the limited range of camera poses. In our Drag4D scenario, where a dense 3D background is essential, this limitation can lead to the emergence of significant visual artifacts. To address this, we augment extrinsics (E_i) of the base images, adjusting camera positions away from the center of the panoramic sphere as follows:

$$\text{Aug}(I_i) = [\Phi(P, K, E_{ij})]_{j=1...T} \quad (5)$$

, which corresponds T number of augmented views from ith base image. Yet, we observe significant artifacts near object boundaries in the augmented views due to depth instability. Thus, we filter out these areas based on a depth gradient threshold and fill them in using pretrained stable diffusion model, SD [6]. The process of image inpainting for augmented views are as follows:

$$\text{PaintAug}(I_i)_j = SD(\Phi(P, K, E_{ij}), M_{ij}) \quad (6)$$

Here, M_{ij} stems from mask filtered out from depth gradient threshold. We can then obtain a pair of base images and inpainting augmented images, $[I_i, \text{PaintAug}(I_i)_j]$.

One-stage 2D-GS Optimization For improved 3D reconstruction, we replace 3D Gaussian primitives with 2D primitives [15]. It has been demonstrated that 2D-GS provides faster and more consistent multi-view consistency evaluations than 3D-GS, a crucial for efficient and accurate 3D reconstruction. It is easily implemented by skipping the third row

and column of Σ' and deriving normal primitive from orthogonal of two tangential vectors. Following the training procedure of 2D-GS [15], we can optimize our model from an initial sparse point cloud P using our panoramic base image I_i with following objective:

$$L_{\text{base}} = L(G_\theta(P, K, E_i), I_i) \quad (7)$$

Here, L represents an integration of the reconstruction loss [18], two regularizers [15] (e.g., depth distortion loss and depth-normal consistency loss). G_θ denotes Gaussian model with θ parameter. Furthermore, to fully leverage the inpainted augmented views $\text{PaintAug}(I_i)_j$ while preventing the model from being constrained by noisy regions, we apply depth-normal similarity as a weighting factor for the inpainted areas of $\text{PaintAug}(I_i)_j$. We simplify the corresponding equation using rendered image of augmented view $R_{\text{aug}} = G_\theta(P, K, E_{ij})$ as below:

$$L_{\text{aug}} = L((1 - M_{ij})R_{\text{aug}} + C_{ij}M_{ij}R_{\text{aug}}, \text{PaintAug}(I_i)_j) \quad (8)$$

C_{ij} denotes depth-normal similarity value.

Finally, we embed semantic features including DINO [28] and DIFT [43] in 3D geometry to be utilized as prior for semantic-level motion-based video generation in Stage3 of Sec. 2.5. Inspired by 3DitScene [56] and LangSplat [31], we apply feature distillation loss, L_{distill}, between rasterized features of the Gaussian Splats and the semantic features constrained by the SAM2 [34] mask, both on base and augmented images. Therefore, total objective loss is expressed as following:

$$L_{\text{scene}} = L_{\text{base}} + L_{\text{aug}} + L_{\text{distill}} \quad (9)$$

We skip the summation of loss for simplicity. This approach allows for joint optimization of 2D-GS on base images and augmented images with learning 3D geometry, which contrasts with Holo-Dreamer [57], where complex multi-stage 3DGS optimization process was introduced. We show that our method can reconstruct an accurate 3D scene in Fig. 4.
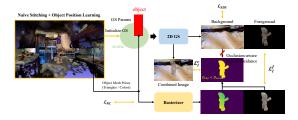
Figure 3. Our object-scene composition pipeline. It begins with naive stitching between 3D background and the position-learned object mesh, which serves as an initialization for 2D-GS optimization. To refine the composition, we apply a photometric reconstruction loss to learn the opacity and spherical harmonic (SH) coefficients of the foreground object, while the background is optimized using the SDS loss [30].

2.4. 2nd Stage: Spatial Alignment of Object

The second stage begins with a user-provided reference image. It extracts the target instance in 3D from this image and integrates it with the generated scene from 1st stage using our proposed 3D copy-and-paste approach. For 3D copy, we first extract the target instance from a reference image with a foreground mask. Then, we leverage off-the-shelf instance-to-3D model [49] to generate a full 3D object of the target instance. Specifically, this model use a multi-view diffusion model [38] to synthesize six novel views at fixed camera poses. These generated multi-view images are then fed into a transformer-based sparse-view reconstruction model to create a high-quality 3D mesh. Next, to perform 3D paste accurately, we take two sequential learning stages. First, we find the floor plane of the scene and roughly locate our object on the plane. We can observe that this naive way of stitching often results in unreliable poses of object as shown in the first column of Fig. 5. To achieve accurate placement of composited objects, we design a physics-aware object-scene composition framework with two regularizers: 1) collision loss, which minimize the collision area between object and scene, and 2) gravity loss, which enforce the object to be grounded within the scene. Our

loss term for this process is as follows:

$$
\begin{aligned}
L_{\text{physics}} &= L_{\text{collision}} + L_{\text{gravity}} \\
&= \sum_{p \in O} \sum_{q \in S} (1 - n_p n_q) \left(\|p - q\|_2 < 1e^{-3} \right) \\
&\quad + \sum_{p \in O} \frac{1}{2} g m_p \left(p - y_{\text{floor}} \right)
\end{aligned}
\tag{10}
$$

Here, O is the point cloud of the object, S is the point cloud of the scene, g is gravitational acceleration, and m_p is the vertex mass of the object. Gravity loss penalized the gravity positional energy relative to the closest floor plane of the scene. Collision loss enforces normal consistency between points that come into contact within a distance below the threshold. This optimization stage yields well-aligned object–scene composited point clouds.

In the next stage, we initialize 2D-GS with the well-aligned object–scene composited point clouds and jointly optimize the integrated object–scene representation, as illustrated in Fig. 3. Consistent with the first stage, semantic features are also embedded into 2D-GS. Specifically, we reuse the semantic features of the background scene and introduce new features for the foreground object. For the latter, DINO features are employed to provide a part-level semantic prior of the target instance, thereby enabling part-level control in stage 3. While optimizing 2D-GS of the object and the scene together, we use SDS loss [30] L_{SDS}, to seemingly generate the occluded area of the background. Our total loss to optimize the scene-object composited 2D-GS is as follows:

$$
L_{\text{scene-object}} = L_{\text{base}} + L_{\text{distill}} + \lambda L_{\text{SDS}}
\tag{11}
$$

Here, we use λ as 0.01.

2.5. 3rd Stage: Temporal Alignment for 4D

Motion Conditioned Video Generation At this stage, we aim to manipulate spatially aligned objects based on a user-defined 3D trajectory path of $(p_x, p_y, p_z)_{i=1...N}$. We begin by generating multi-view images projected from P_{comp} by setting different azimuth and elevation angles for the camera views. For example, we define azimuth

Table 2. Comparison of Methods on Various Quality Metrics on *Drag4D-30* dataset.

Method	Image Quality (Novel View)				Render Quality	
	CLIP-Score ↑	Sharp ↑	Colorful ↑	Quality ↑	PSNR ↑	SSIM ↑
LucidDreamer [5]	0.656	0.961	0.603	0.704	-	-
SceneDreamer360 [20]	0.773	0.970	**0.760**	0.736	24.59	0.857
Ours	**0.782**	**0.973**	0.740	**0.747**	**25.74**	**0.885**

range as [0°, 90°, 180°, 270°] and elevation range as [0°, 30°], offering users unconstrained multi-views. The 3D trajectory is then reprojected for each view, yielding total 8 pairs of multi-view images and corresponding 2D trajectories. We represent those pairs as $(V_i, T_i)_{i=1...8}$, where V_i denotes each view image and T_i its corresponding 2D trajectory. To generate video from image V following a specified trajectory T, we opt DragAnything [48] as a motion-conditioned video model. We train the model on large-scale video dataset, VIPSeg [23], using ControlNet [54] to condition trajectory following DIFT features. Specifically, the DIFT features of the video's first frame are pooled based on each mask, allowing us to obtain an entity representation $\hat{E} \in \mathbb{R}^{H \times W \times C}$ along with a gaussian heatmap h. They are mapped to the following frames according to the trajectory path. The objective to condition the motion into video diffusion model can be simplified to:

$$\mathcal{L}_\theta = \sum_{i=1}^{L} \left\| \epsilon - \epsilon_\theta \left(z, \mathcal{E}_\theta(\hat{\mathbf{E}}_i), \mathcal{E}_\theta(h_i) \right) \right\|_2^2 \quad (12)$$

where latent feature of first frame (z) and encoded trajectory features $(\mathcal{E}_\theta(\hat{\mathbf{E}}_i), \mathcal{E}_\theta(h_i))$ with encoder $\mathcal{E}$ are added to the denoised features in diffusion model. However, as shown in the first row of the second box in Fig. 6, we observe motion hallucination in a local part of the moving object. To address this, we augment the current DIFT features from the instance mask with a part-level DIFT feature derived from the part mask of each instance. For VIPSeg, consistent with the previous stage, we use DINOv2 [28] and apply k-means clustering within the instance mask to obtain the part mask. This enables us to extract part features, which are then concatenated with the global feature and their re-

Table 3. Ablation Studies of Object Scene Composition. There is trade-off between geometric quality and image quality depending on usage of SDS loss and joint learning of background.

Method	SDS	Normal. C	w/o BG	Rasterize	CLIP-Score ↑	Sharp ↑	Align ↓
(1)	✗	✓	✗	✓	0.643	0.971	**0.134**
(2)	✓	✓	✗	✗	0.661	0.973	0.265
(3)	✓	✗	✓	✓	0.753	**0.977**	0.545
(4)	✓	✓	✗	✓	**0.775**	0.976	0.529
(5)	✓	✓	✓	✓	**0.775**	0.976	0.528

spective trajectories, represented as $E_i^{\hat{\text{part}}}$ and h_i^{part}. Consequently, Eq. (12) can be modified as follows:

$$\mathcal{L}_\theta = \sum_{i=1}^{L} \left\| \epsilon - \epsilon_\theta \left(z, \mathcal{E}_\theta(\hat{\mathbf{E}}_i), \mathcal{E}_\theta(h_i), \mathcal{E}_\theta(\mathbf{E}^{\hat{\text{part}}}_i), \mathcal{E}_\theta(h_i^{\text{part}}) \right) \right\|_2^2 \quad (13)$$

This modified objective serves as the overall training goal for our part-augmented, motion-conditioned video generation. As the model aims to align motion by considering both local and global features of instances, we refer to it as **L**ocal-**G**lobal DragAnything. Given the pretrained LG-DragAnything and desired trajectory from users, we generate both global- and part-level trajectory features from (V_i, T_i, F_i), enabling the synthesis of eight motion-conditioned videos with a single model. Here, F_i represents the DIFT feature map, partitioned at the part and instance levels. From the optimized object–scene composited 2D-GS, we rasterize (V_i, T_i, F_i) for each view and leverage these priors to produce spatially and temporally coherent 4D videos with LG-DragAnything.

3. Experiments

In this section, we evaluate Drag4D across three key tasks: Text-to-3D Generation (Sec. 3.1),

Object-Scene Composition (Sec. 3.2), and Motion-Conditioned Video Generation (Sec. 3.3), each aligning with one of Drag4D's distinct stages. We provide the details of baselines used for those three tasks, main quantitative and qualitative results, and ablation studies.

Datasets Due to the absence of publicly available datasets for validating 4D environments with motion guidance, we created our own dataset, named *Drag4D-30*. The dataset comprises 30 complex and extended text prompts designed to synthesize corresponding 3D scenes, demonstrating the effectiveness of Drag4D compared to other baselines in 1st stage. Additionally, it includes 4 object-centric images, each containing a target instance. These images are used for evaluating the 2nd stage and 3rd stage, where they are paired with the 30 text prompts to assess the spatial and motion alignment of the object within the 30 3D scenes. We will provide the details of *Drag4D-30* in supplementary material.

3.1. Text-to-3D Generation

Baselines As 1st stage of our Drag4D aims to reconstruct 3D scene from text prompt, we compare our approach with two recent methods. 1) LucidDreamer [5], employs a technique where outpainted RGBD images are mapped onto a point cloud, which is then used to guide the optimization of 3DGS by projecting various images derived from this point cloud. However, as LucidDreamer lacks the capability to directly produce 3D scenes from textual prompts, we address this limitation by leveraging a diffusion model to create conditional images, enabling the generation of 3D scenes based on text input. 2) SceneDreamer360° [20], utilizes a text-driven panoramic image generation model, fine-tuned with a three-stage enhancement process, to produce high-resolution panoramas. These panoramas are integrated into 3D space using 3D-GS, ensuring multi-view consistency.

Main Results To assess the fidelity of the generated 3D scenes to the text prompts, we calculate the CLIP-Score [9]. Additionally, we use CLIP-IQA [45] to evaluate visual sharpness, colorfulness, and quality. Both metrics leverage the pre-trained

Table 4. Performance comparison of motion-conditioned video generation on VIPSeg validation set. Our proposed LG-DragAnything with part augmentation surpasses the baseline across both image-based metrics (FID, PSNR, SSIM) and video-based metrics (FVD and ObjMC). Notably, higher values indicate better performance for PSNR and SSIM, while lower values are preferable for FID, FVD, and ObjMC. Results marked with ∗ indicates that we reproduce better score from baseline, DragAnything [48].

Method	FID ↓	FVD ↓	PSNR ↑	SSIM ↑	ObjMC ↓
∗DragAnything [48]	34.45	288.68	18.41	0.57	19.9
LG-DragAnything	32.79	272.02	19.02	0.59	17.6

Table 5. Comparison between the DragAnything [48] and our LG-DragAnything on *Drag4D-30* dataset.

Method	CLIP-Score ↑	Quality ↑	Colorful ↑
∗DragAnything [48]	0.805	0.48	0.71
LG-DragAnything	0.814	0.51	0.75

CLIP-B/32 model [32]. Furthermore, PSNR and SSIM are employed to measure rendering quality. As presented in Tab. 2, Drag4D surpasses the baselines in both image quality and rendering quality. The qualitative results in Fig. 4 demonstrate that Drag4D produces visually complete and less distorted 3D scenes, attributed to our joint training with base images and inpainting-augmented views.

3.2. Object-Scene Composition

Baselines and Main Results Since there is previous baseline in object-scene composition, we construct our self-baseline as summarized in Tab. 3. We can observe that using SDS loss and normal consistency loss help to increase both CLIP-Score and Sharpness. Additionally, according to Fig. 5, we can easily find out that using both collision loss and gravity loss help to position the object accurately.

3.3. Motion-Conditioned Video Generation

Baselines We chose DragAnything [48] as a representative baseline to evaluate motion-conditioned video generation. It proposes a framework for controllable video generation that uses entity representationhs for motion control of any object. It

Figure 4. Qualitative Results on the 1st Stage with Drag4D dataset. We show rendered color images in novel viewpoint and mesh reconstructed from (a) Lucid-Dreamer [5] (b) SceneDreamer360 [20], and (c) ours. Our method can effectively handle unseen viewpoints due to our adaptive inpainting strategy. It is best viewed in color and high resolution; please zoom in .

Figure 5. Ablation study of physics-aware position learning used in object-scene composition

enables trajectory-based interaction, removing the need for additional guidance signals like masks or depth maps. Our proposed Drag4D introduces part augmentation strategy on top of DragAnything. Both methods are trained with VIPSeg [23] training datasets.

Main Results We evaluate the motion alignment of our part augmentation method compared to DragAnything in two scenarios: 1) motion alignment in 2D videos: Using the VIPSeg validation set, we assess performance based on image metrics (e.g., FID, PSNR, and SSIM) and video metrics (e.g., FVD and ObjMC). As shown in Tab. 4, part augmentation demonstrates improved performance across these metrics. 2) motion alignment in 4D

Figure 6. Qualitative Results on the 3rd Stage with VIPSeg dataset [23]. We adapt the VIPSeg dataset by annotating it with part segmentation, achieved through feature clustering from DINOv2 [28]. This modified dataset is then used to train a part-augmented, motion-conditioned video model as described in Eq. (13). Our results show that LG-DragAnything with part augmentation effectively reduces motion hallucination by accounting for motion at both global and part levels. Best viewed in color and high resolution; please zoom in for finer details.

videos: We measure CLIP-Score to evaluate text fidelity and utilize CLIP-IQA metrics, including quality and colorfulness, to assess the quality of multi-view videos on *Drag4D-30* dataset. We summarize the quantitative result in Tab. 5. Fig. 6 shows visual impact of LG-DragAnything in the first scenario, while Fig. 7 and Fig. 8 correspond to the second scenario. The results from these two scenarios demonstrate that our proposed part-augmentation in motion-conditioned video generation effectively reduces local hallucinations while improving fidelity to the text prompt.

4. Conclusion

We introduce **Drag4D**, a comprehensive interactive pipeline designed to align user-defined 3D object motion with text-driven 3D background scene generation. In the first stage, Drag4D generates a high-fidelity 3D scene by optimizing 2D Gaussian representations on panoramic images and their
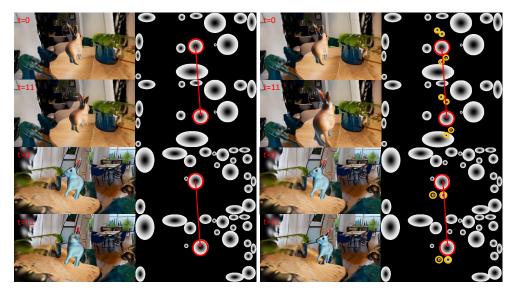
Figure 7. Qualitative Results on the 3rd Stage, which is our multi-view generated video result. Left is from DragAnything and right is from our LG-DragAnything. Part guidance leads to clear and intended results.
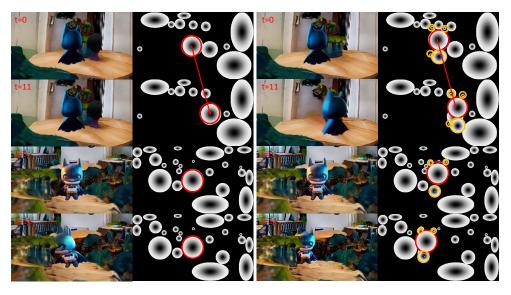

Figure 8. Qualitative Results on the 3rd Stage, which is our multi-view generated video result. Left is from DragAnything and right is from our LG-DragAnything. Part guidance leads to clear and intended results.

augmented views, surpassing previous state-of-the-art models in 3D scene generation. In the second stage, Drag4D extracts the target instance from a user-provided reference image, transforming it into a full 360° object that is spatially aligned with the generated 3D scene using our proposed 3D Copy-and-Paste method. The final stage further enhances user experience, allowing temporal manip-

ulation of the 3D object within the 3D scene using a part-augmented motion-conditioned video generator and 4D Gaussian representations. We anticipate significant societal benefits from Drag4D, as it performs robustly across diverse user prompts, offering potential applications in fields such as entertainment, video synthesis and AR/VR.

References

[1] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, pages I–I. IEEE, 2001. 16

[2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 17

[3] 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. 2

[4] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021. 15

[5] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023. 2, 7, 8, 9, 15

[6] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 2, 5

[7] Gege Gao, Weiyang Liu, Anpei Chen, Andreas Geiger, and Bernhard Schölkopf. Graphdreamer: Compositional 3d scene synthesis from scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21295–21304, 2024. 15

[8] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024. 15

[9] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 8

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 3

[11] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7909–7920, 2023. 2

[12] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7909–7920, 2023. 15

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 4

[14] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024. 2

[15] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2, 5, 17

[16] Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. Peekaboo: Interactive video generation via masked-diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8079–8088, 2024. 2

[17] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024. 2

[18] 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. 3, 5

[19] Haoran Li, Haolin Shi, Wenli Zhang, Wenjun Wu, Yong Liao, Lin Wang, Lik-hang Lee, and Pengyuan Zhou. Dreamscene: 3d gaussian-based text-to-3d scene generation via formation pattern sampling. *arXiv preprint arXiv:2404.03575*, 2024. 15

[20] Wenrui Li, Yapeng Mi, Fucheng Cai, Zhe Yang, Wangmeng Zuo, Xingtao Wang, and Xiaopeng Fan. Scenedreamer360: Text-driven 3d-consistent scene generation with panoramic gaussian splatting. *arXiv preprint arXiv:2408.13711*, 2024. 2, 4, 7, 8, 9, 15, 16

[21] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024. 2

[22] Yikun Ma, Dandan Zhan, and Zhi Jin. Fastscene: Text-driven fast 3d indoor scene generation via panoramic gaussian splatting. *arXiv preprint arXiv:2405.05768*, 2024. 15

[23] Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-scale video panoptic segmentation in the wild: A benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 7, 9

[24] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019. 15

[25] Thu H Nguyen-Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. *Advances in neural information processing systems*, 33:6767–6778, 2020.

[26] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 15

[27] OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, 2024. Accessed: 2024-11-21. 17

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 5, 7, 9

[29] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 2

[30] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 2, 6, 15

[31] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 5

[32] 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. 8

[33] Jason Rambach, Gergana Lilligreen, Alexander Schäfer, Ramya Bankanal, Alexander Wiebel, and Didier Stricker. A survey on applications of augmented, mixed and virtual reality for nature and environment. In *International Conference on Human-Computer Interaction*, pages 653–675. Springer, 2021. 2

[34] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and

Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 5

[35] Manuel Rey-Area, Mingze Yuan, and Christian Richardt. 360monodepth: High-resolution 360deg monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3762–3772, 2022. 16

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 15

[37] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 15

[38] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 6

[39] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 2

[40] Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. Realmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion. *arXiv preprint arXiv:2404.07199*, 2024. 15

[41] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 3

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 15

[43] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023. 5

[44] Alexander Vilesov, Pradyumna Chari, and Achuta Kadambi. Cg3d: Compositional generation for text-to-3d via gaussian splatting. *arXiv preprint arXiv:2311.17907*, 2023. 15

[45] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2555–2563, 2023. 8

[46] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 4

[47] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 15

[48] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2025. 2, 3, 7, 8, 15

[49] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 3, 6, 17

[50] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023. 15

[51] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9043–9053, 2023. 4, 16

[52] Cheng Zhang, Qianyi Wu, Camilo Cruz Gambardella, Xiaoshui Huang, Dinh Phung, Wanli Ouyang, and Jianfei Cai. Taming stable diffusion for text to 360∘ panorama image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 17

[53] Cheng Zhang, Qianyi Wu, Camilo Cruz Gambardella, Xiaoshui Huang, Dinh Phung, Wanli Ouyang, and Jianfei Cai. Taming stable diffusion for text to 360 {\deg} panorama image generation. *arXiv preprint arXiv:2404.07949*, 2024. 2, 4

[54] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala.

13

Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 7

[55] Qihang Zhang, Chaoyang Wang, Aliaksandr Siarohin, Peiye Zhuang, Yinghao Xu, Ceyuan Yang, Dahua Lin, Bolei Zhou, Sergey Tulyakov, and Hsin-Ying Lee. Towards text-guided 3d scene composition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6829–6838, 2024. 15

[56] Qihang Zhang, Yinghao Xu, Chaoyang Wang, Hsin-Ying Lee, Gordon Wetzstein, Bolei Zhou, and Ceyuan Yang. 3ditscene: Editing any scene via language-guided disentangled gaussian splatting. *arXiv preprint arXiv:2405.18424*, 2024. 5, 15

[57] Haiyang Zhou, Xinhua Cheng, Wangbo Yu, Yonghong Tian, and Li Yuan. Holodreamer: Holistic 3d panoramic world generation from text descriptions. *arXiv preprint arXiv:2407.15187*, 2024. 2, 5, 15, 16

[58] Junwei Zhou, Xueting Li, Lu Qi, and Ming-Hsuan Yang. Layout-your-3d: Controllable and precise 3d generation with 2d blueprint. *arXiv preprint arXiv:2410.15391*, 2024. 2, 15

[59] Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suya You, Zhangyang Wang, and Achuta Kadambi. Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In *European Conference on Computer Vision*, pages 324–342. Springer, 2024. 15

14

Supplementary Material

A. Related Works

3D Scene Generation. 3D Scene generation has been actively studied due to the rapid development of image generation. Early studies [4, 24–26, 37] utilize Generative Adversarial Networks (GANs) and implicit neural networks to represent 3D objects with texture. However, these methods have limited ability to generate diverse categories of objects and scenes due to GAN's inherent difficulty of learning and limited 3D representation. Advanced recent studies [5, 12, 30, 56] generate large 3D scenes from either text prompts or a single image from the user by incorporating advanced diffusion-based image generation techniques [36, 42]. These methods use Neural Radiance Fields (NeRFs) [12, 30] to represent 3D scenes or 3D Gaussian Splatting [5, 40, 56] for creating high-fidelity results and efficient generation. To generate 3D scene with a large field of view, these methods use diffusion prior to progressively outpaint the unseen part of the scene and integrate it to get the full 3D scene. The combined 3D scene from these outpainting steps often suffers from multi-view inconsistencies, as diffusion priors struggle to maintain coherence across different camera viewpoints. To alleviate this limitation, some recent methods [20, 22, 57, 59] generate panoramic images from text prompt and learn to reconstruct 3D scene using 3D Gaussian Splatting. Compared to these recent methods, our method utilizes 2D Surfel Gaussian, takes advantage of high-quality geometry reconstruction, and also inpaints unseen parts of augmented viewpoints in the training stage so that our method shows robustness under unseen novel viewpoints as well.

Object Scene Composition. Given the desired 3D layout positions of multiple objects, there have been several attempts to generate these objects together within a cohesive scene. CG3D [44] enables physically realistic composition and generation of multiple objects by using physics-inspired losses. GraphDreamer [7] employs scene graphs to represent relationships between multiple objects, ensuring the generated scene follows these relational constraints. More recent studies [19, 55] have explored compositing scenes and objects derived from text-to-3D models. However, the textures produced by these methods often lack realism due to their heavy dependence on diffusion models or CLIP priors. Moreover, we employ two-stage optimization inspired by [58], first optimizing the object's position using physics prior and jointly training positioned objects with a pre-trained scene for the natural composition.

Controllable Video Generation. With advancements in diffusion models significantly improving video generation performance, there has been a growing interest in controllable video generation. While numerous existing studies focus on generation conditioned on text, images, depth, or skeletal data, our work is specifically aligned with video generation conditioned by either camera movement or motion trajectories. DragNUWA [50] proposes a multi-scale trajectory encoding approach that integrates trajectory conditioning into a video diffusion model, along with adaptive training to effectively learn from dense optical flow to more intuitive, user-friendly motion. DragAnything [48] introduces an entity representation to enable instance-level motion guidance, effectively mitigating distortions and undesired deformations often associated with point-based trajectories. Building on these approaches, CameraCtrl [8] use Plücker coordinates to precisely control camera trajectories, while MotionCtrl [47] goes further by decomposed control of both camera and object trajectories. Our work builds upon DragAnything [48], extending it to allow part-based instance control. We observed that controlling only at the instance level is insufficient for articulated objects and often leads to hallucinations in the generated result.

B. Implementation Details

In this section, we aim to elaborate on the details of Stage 1 and Stage 2, which were not fully covered in the manuscript.
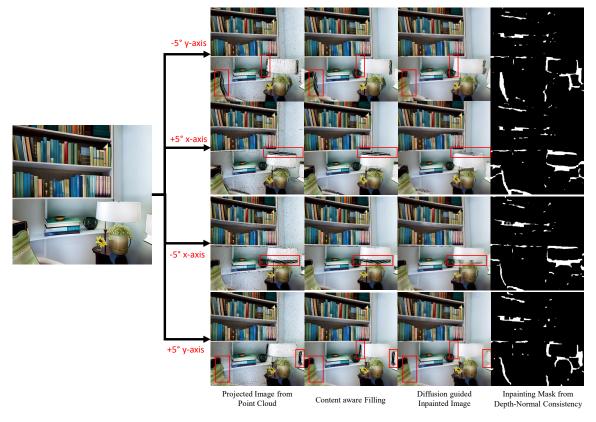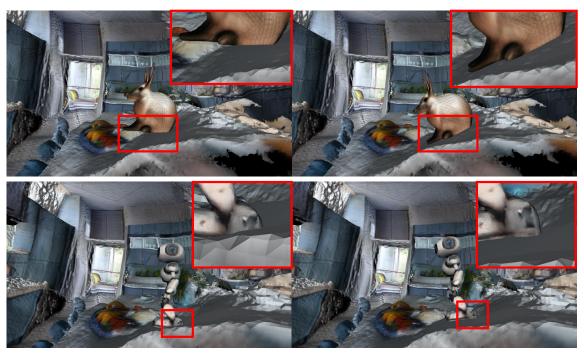
Figure 9. **Viewpoint Augmentation in Stage 1**: We describe our viewpoint augmentation strategy adapted in stage1 to reconstruct 3D scene from given panorama image. First, the point cloud is projected onto four augmented viewpoints, where the consistency of the projected depth and normals is evaluated. A geometric uncertainty map derived from this evaluation guides the inpainting process, addressing unseen and distorted pixels.

Stage 1. 3D Scene Generation. Given the panoramic image generated from the text, we follow our baselines [20, 57] to augment viewpoints additional to the sphere projected images from the panorama. This is because the projected images only offer very limited camera viewpoints, leading the reconstructed scene to be overfitted to that viewpoint and significantly distorted when viewed from novel perspectives. Unlike previous works [20, 57], our approach directly generates images at augmented viewpoints through a combination of view projection and inpainting steps depicted in Fig. 9. Specifically, we begin by projecting the globally aligned point cloud, obtained using methods from [35, 51] and following the projection process in [57], to create a projected image with holes. Next, we use content-aware filling [1] to fill these holes. To evaluate consistency, we calculate the cosine similarity between the projected depth and normals derived from the point cloud. Finally, we define an uncertainty mask for regions with similarity below 0.75 and in-paint pixels within these uncertain areas.

Stage 2. Object Scene Composition. We show our physics-aware object-scene composition framework's effectiveness in Fig. 10.

Hyperparameters. When we train 2D-GS in stage 1, we use the same parameters as Scene-Dreamer360 [20] and train 4000 iterations in total. We use the learning rate of object position learning

Before Object Position Learning   After Object Position Learning

Figure 10. **Object Position Learning in Stage 2**: In stage 2, object position learning is crucial in aligning the object's position to seamlessly fit within the given scene. We demonstrate the visual difference between including this learning step and not.

as 0.001.

C. Drag4D-30 Dataset

Our Drag4D-30 Dataset features 30 distinct 3D scenes, each containing four different objects: "teddy bear," "batman," "rabbit," and "robot." To obtain these object assets, we generate 3D textured meshes from single images using InstantMesh [49] and DALLE3 [2]. For constructing the 3D scenes, we first generate 30 different panoramic images and corresponding text prompts using ChatGPT-4o [27] and PanFusion [52]. Subsequently, we reconstruct 3D scene using 2D-GS [15], described in stage 1 of the manuscript, to produce complete 3D scenes from these panoramic images. Finally, we visualize some samples of our Drag4D-30 dataset including text prompt, panorama image, and a mesh of the generated scene with aligned 4 objects (result after stage 1 training is finished), in Fig. 11.

D. Additional Results

We present additional qualitative comparisons between our final 4D dragged video generated by DragAnything (baseline) and our Local-Global DragAnything (LG-DragAnything) in Figs. 12 and 13. LG-DragAnything successfully models both part-level and global motion, enabling the video diffusion model to move objects more accurately along the input trajectory while minimizing visual artifacts or hallucinations.

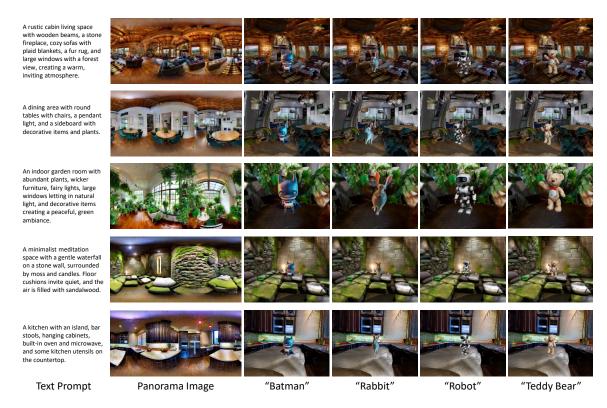| Text Prompt | Panorama Image | "Batman" | "Rabbit" | "Robot" | "Teddy Bear" |

Figure 11. **Drag4D-30 Dataset**: We visualize some of our aligned 3D scenes with objects in 3D. Our scene is generated and reconstructed from the following text prompts. Thanks to our physically plausible object position learning step, our object is well-composited with the reconstructed scene. **Please take a closer look to observe the finer details.**

We also demonstrate that our objects move naturally along the given 3D path within the scene by showing multi-view rendered moving objects in Figs. 14 to 18.

E. Discussions and Limitations

Our research focuses on 3D object motion control within a scene, therefore, modeling the object's texture under natural scene lighting is beyond the scope of our paper. However, leveraging the provided Drag4D-30 dataset to model realistic object textures within a generated environment under non-Lambertian assumptions presents an interesting direction for future work. In our current work, we identify two interesting failure cases, as illustrated in Fig. 19. First, due to the inherent reliance on 2D trajectories as a condition for our video generation model, motion parallel to the camera view introduces ambiguity (depicted in case 1). For future research, we aim to resolve this issue by introducing a new 3D representation for trajectory conditions. Second, our model faces challenges in handling fast, drastic movements (depicted in case 2). This limitation, commonly observed in recent trajectory-conditioned video generation methods, represents a promising future research direction.

F. Asset License

The licenses of the assets used in the experiments are denoted as follows:

Datasets:
- **VIP-Seg** (Miao et al., 2022): `https://github.com/VIPSeg-Dataset/VIPSeg-Dataset`

Codes:
- **2D Gaussian Splatting** (Huang et al., 2024): `https://github.com/hbb1/2d-gaussian-splatting`
- **InstantMesh** (Xu et al., 2024): `https://github.com/TencentARC/InstantMesh`
- **Stable Video Diffusion** (Blattmann et al., 2023): `https://huggingface.co/stabilityai/stable-video-diffusion-img2vid-xt`
- **ThreeStudio** (Guo et al., 2023): `https://github.com/threestudio-project/threestudio`
- **PanoFusion** (Zhang et al., 2024): `https://github.com/chengzhag/PanFusion`

Figure 12. Further qualitative comparisons between our baseline and LG-DragAnything are provided. We denote **part motion** and **global motion** for the reader's understanding. Our method effectively guides both global and part movements, ensuring strict compliance with the given trajectory without hallucination and distortion.
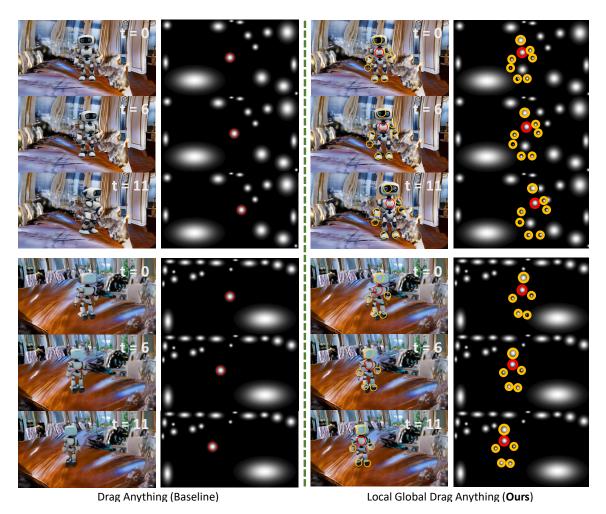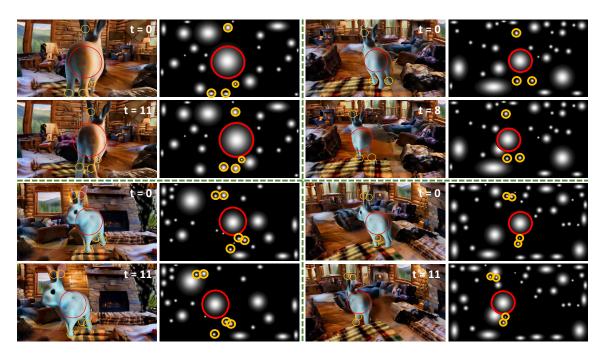
Figure 13. Further qualitative comparisons between our baseline and LG-DragAnything are provided. We denote **part motion** and **global motion** for the reader's understanding. Our method effectively guides both global and part movements, ensuring strict compliance with the given trajectory without hallucination and distortion.

Figure 14. A multi-view generated sequence of a moving rabbit moving in the 'cabin space' scene.
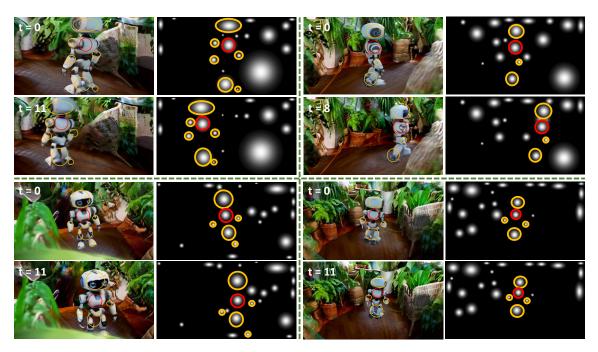

Figure 15. A multi-view generated sequence of a moving robot moving in the 'garden room' scene.

Figure 16. A multi-view generated sequence of four different objects moving in the 'greenhouse' scene.
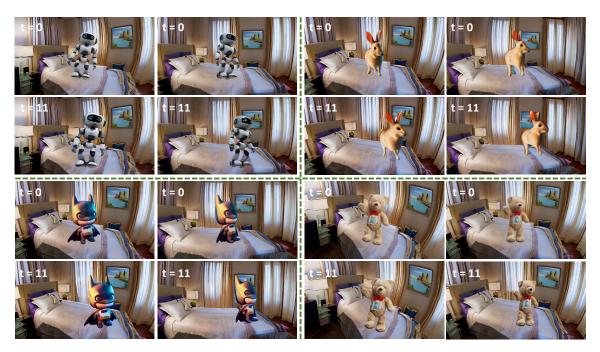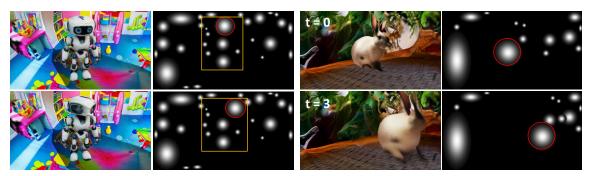

Figure 17. A multi-view generated sequence of four different objects jumping in the 'room' scene.

23

Figure 18. A multi-view generated sequence of four different objects moving in the 'playroom' scene.

Failure Case 1: Parallel Motion to Viewpoint   Failure Case 2: Drastic Motion Control

Figure 19. **Failure Cases**: Our method struggles with the motion moving parallel to the viewpoint because of the projected 2D trajectory's ambiguity (Case 1) and drastic motion control (Case 2) which is a common unsolved problem for trajectory-conditioned video generation studies.