Comp4D: Compositional 4D Scene Generation

Anonymous authors
Paper under double-blind review

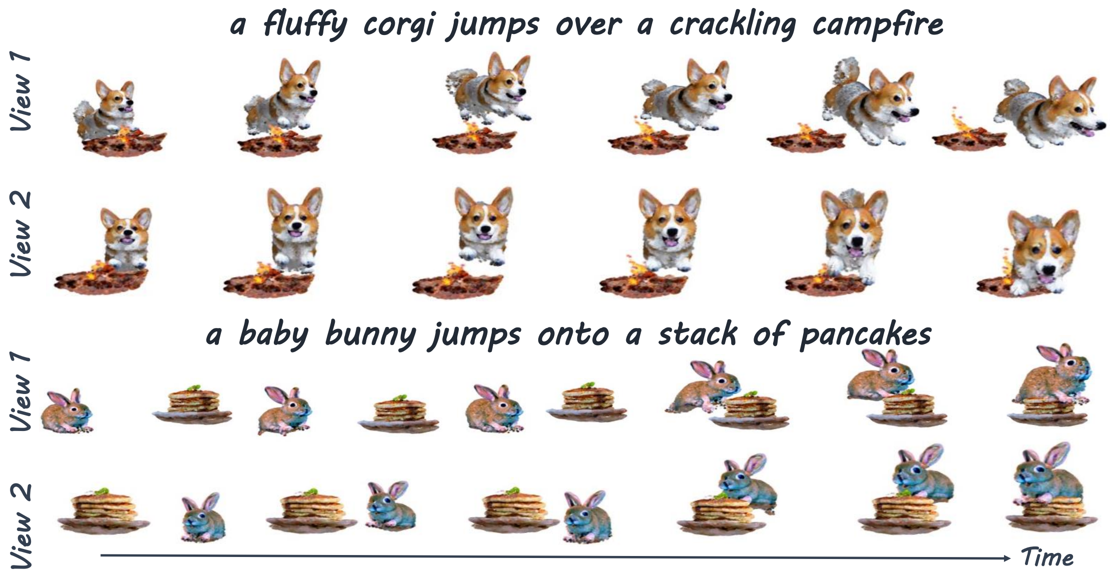

Figure 1: We present **Comp4D**: compositional 4D scene generation from text input. Our model can render realistic images from generated 4D assets at various viewpoints and different timestamps.

Abstract

Recent advancements in diffusion models for 2D and 3D content creation have sparked a surge of interest in generating 4D content. However, the scarcity of 3D scene datasets constrains current methodologies to primarily object-centric generation. To overcome this limitation, we present Comp4D, a novel framework for compositional 4D scene generation. Unlike conventional methods that generate a singular 4D representation of the entire scene, Comp4D innovatively employs a decompose-then-recompose strategy, constructing each 4D component within the scene separately. The framework first decomposes a textual input prompt into multiple object components and delineates their moving trajectories. After initializing the static 3D objects, we construct the compositional 4D scene by accurately positioning these objects along their designated paths. To refine the scene and motion, our method proposes a novel compositional score distillation technique involving trajectory-guided and object-centric sampling, utilizing pre-trained diffusion models across text-to-image, text-to-video, and text-to-3D domains for optimization. Extensive experiments demonstrate our superior 4D content creation capability compared to prior arts, showcasing superior visual quality, motion fidelity, and enhanced object interactions.

1 Introduction

Recent advances in text-to-image diffusion models Saharia et al. (2022); Nichol et al. (2021); Ramesh et al. (2022); Rombach et al. (2022) have revolutionized generative AI, simplifying digital content creation. Traditional pipelines, often cumbersome and reliant on domain expertise, are being replaced by these generative

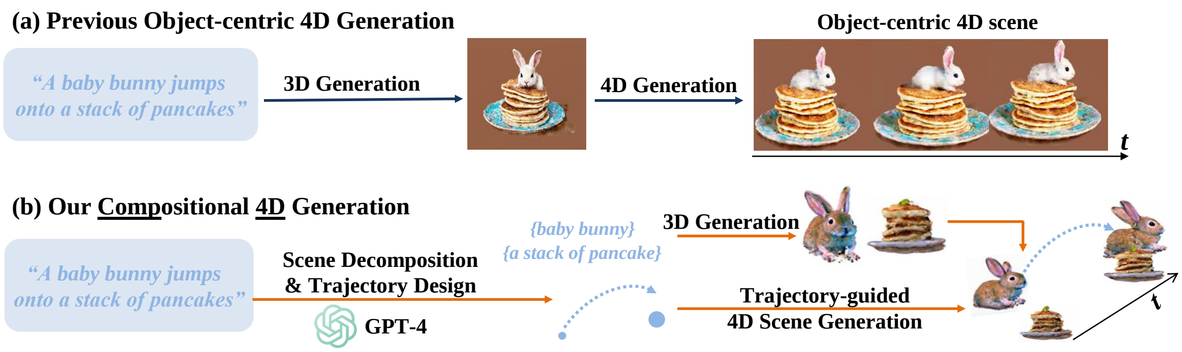

Figure 2: Compared with previous object-centric 4D generation pipelines, our Compositional 4D Generation (**Comp4D**) framework proposes decompose-then-recompose strategy, fulfilling larger-scale movements and more realistic object interactions.

models that bring complex ideas to life from simple text prompts. This innovation extends to the domain of 3D content creation, where score distillation techniques Poole et al. (2022); Xu et al. (2022); Wang et al. (2023a); Shi et al. (2023b); Wang et al. (2023b); Liu et al. (2023) leverage 2D diffusion models to generate 3D content. Meanwhile, image diffusion models have also made significant strides in video generation, prompting further exploration into adapting these models for 4D content creation. The 4D synthesis works often rely on partial or joint supervision signals from text prompts Ling et al. (2023); Singer et al. (2023); Zheng et al. (2023); Yin et al. (2023); Bahmani et al. (2023), images Yin et al. (2023); Ren et al. (2023); Zhao et al. (2023); Liang et al. (2024), 3D models Yin et al. (2023); Zheng et al. (2023), or monocular videos Yin et al. (2023); Jiang et al. (2023); Ren et al. (2023), to guide the generation process.

Despite notable advancements, current 4D content creation predominantly focuses on object-centric generation. This limitation is mainly attributed to the scarcity of comprehensive scene-level dynamic 3D datasets. MVDream Shi et al. (2023b) and Zero-123 Liu et al. (2023); Shi et al. (2023a) trained on Objaverse Deitke et al. (2023b) are widely adopted in 4D content creation pipelines Ling et al. (2023); Singer et al. (2023); Zheng et al. (2023); Yin et al. (2023); Ren et al. (2023); Zhao et al. (2023); Jiang et al. (2023); Bahmani et al. (2023), which provide direct supervision on the multi-view renderings with geometry awareness. Compared to 2D diffusion models, these 3D-aware diffusion models greatly improve the 3D geometry quality Liu et al. (2023); Shi et al. (2023b). However, their focus on object-centric generation persists, attributed to the reliance on the training data Deitke et al. (2023b;a), which is comprised mostly of synthetic objects positioned at the world origin. Notably, in these datasets, the object's global movement consistently synchronizes with the camera, exhibiting only local deformation. However, in a complex 4D scene with multiple objects, we expect to observe not only individual local deformations but also inter-object global displacement.

As illustrated in Fig. 2, to address the aforementioned challenges, we propose **Comp4D**, the first text-to-4D scene generation work that extends the previous boundaries in object-centric 4D generation to the demanding task of 4D scene construction. To overcome the prevalent object-centric constraint, our approach disentangles the **compositional 4D scene** generation into two stages: scene decomposition for constructing individual static 3D assets, and scene re-composition with motion modeling. The motion modeling is further factorized into global displacements and local deformations. We manually or utilize a Large Language Model (LLM) to delineate the movement trajectories of each object to guide global displacements. This alleviates the computational load on deformation modules by narrowing their focus on local deformations. Formulating each object as disjoint 3D Gaussians, we introduce a novel compositional score distillation sampling mechanism in the re-composition stage. We selectively render the whole scene or partial objects for motion optimization. This strategy acts as a powerful augmentation to enhance the motion fidelity of each object, especially in scenarios where object occlusion becomes prevalent as objects move.

The generation of our 4D scene is conducted through the following steps. Given an input text description, we first leverage an LLM to decompose the scene by extracting entities and determining their attributes, such as scale. Following this, static 3D objects are individually constructed using pre-trained 3D-aware diffusion models. Meanwhile, we manually or take advantage of the LLM to design kinematics-based trajectory functions to guide object global displacement. Subsequently, we re-compose the 4D scene with comprehensive motion learning. Each object's deformation field is optimized via a novel compositional score distillation mechanism, with objects moving along the pre-defined trajectories.

Our key contributions can be summarized as follows:

- We introduce **Comp4D**, the first framework that achieves Compositional 4D scene generation conditioned on text. By formulating 4D scene generation as the construction of individual 4D objects and their interactions, we overcome the object-centric constraint in previous methods.
- We propose to factorize the object motions into global displacements and local deformations. The global displacement, implemented via kinematics-based trajectories, offloads the burden on 4D representation and enables it to concentrate solely on local deformations.
- Comp4D re-composes the scene with a novel compositional score distillation sampling mechanism, incorporating trajectory-guided and object-centric optimization. This design enables flexible switching between whole-scene and partial-objects renderings, facilitating stable optimization of object motion even in the presence of entity occlusions.
- Extensive experiments compared to existing baselines demonstrate the superiority of our model in compositional 4D scene generation in terms of visual quality, motion realism, and object interaction.

2 Related Works

2.1 4D Content Creation

Text-guided diffusion models have significantly advanced image and video generation. However, the scarcity of large-scale annotated 3D datasets constrains progress in 3D generative learning. To address this limitation, the score distillation sampling (SDS) Poole et al. (2022) is proposed for optimization-based text-to-3D generation. Presently, some researchers have extended SDS to dynamic 4D scene generation. MAV3D Singer et al. (2023) is a pioneering work generating dynamic 4D scenes from text prompts. It uses NeRFs with HexPlane features for 4D representation. 4DFY Bahmani et al. (2023) leverages a NeRF-based representation with a multi-resolution feature grid, combining supervision signals from images, videos, and 3D-aware diffusion models for text-to-4D synthesis. Consistent4D Jiang et al. (2023) tackles the task of video-to-4D generation with the help of RIFE Huang et al. (2022) and a 2D super-resolution module. With the advances in 3D Gaussians, AYG Ling et al. (2023) proposes generating 4D scenes using dynamic 3D Gaussians, disentangling the 4D representation into static 3D Gaussian Splatting and a deformation field for modeling dynamics. 4DGen Yin et al. (2023) introduces a driving video to ground 4D content creation from text or image, providing added controllability in motion generation through reconstruction loss alongside score distillation. DreamGaussian4D Ren et al. (2023) adopts mesh texture refining through video diffusion models to improve texture quality. However, existing works in 4D content creation focus on object-centric generation due to the underlying constraints of the 3D-aware diffusion model. In contrast, this work is the first attempt to tackle the challenging compositional 4D scene generation task by decomposing the scene into object components.

2.2 4D Scene Representation

Building 4D scene representation allows for rendering novel views of objects under rigid and non-rigid motions. Recently, 3D Gaussian Splatting (3D-GS) Kerbl et al. (2023) has shown advantages in both effectiveness and efficiency for 4D representation, leading to multiple directions to model temporal dynamics. Katsumata *et al.* Katsumata et al. (2023) and 4DGS Wu et al. (2023a) define scales, positions, and rotations as functions of time while leaving other time-invariant properties of the static 3D Gaussians unchanged. Another direction involves directly extending 3D Gaussians to 4D with temporal slicing Yang et al. (2023b); Duan et al. (2024). There are also works leveraging a separate function to model the dynamic distribution of attributes' deformation for 3D Gaussians Lin et al. (2023); Li et al. (2023b). In this work, we adopt 3D Gaussians for our 3D content representation and use an additional Multi-layer Perceptron (MLP) to deform each set of 3D Gaussians. This disentangled 4D representation allows us to construct the static scene first and then focus on modeling the object's deformation.

2.3 Grounding and Reasoning from Large Language Models

LLMs have emerged as a natural tool for performing reasoning tasks and enabling implementation in the real world Li et al. (2022); Lu et al. (2023); Ichter et al. (2022); Rajani et al. (2019). A popular approach to improving the reasoning capabilities of LLMs is to fine-tune models on domain-specific tasks Yang et al. (2023a). Moreover, recent studies have explored techniques for incorporating multimodal information, such as images and videos, to enhance contextual understanding and improve the robustness of language models, paving the way for more effective applications in various domains Seff et al. (2023); Tikayat Ray et al. (2023). Specifically, LLMs have recently been used for generating trajectories in robotics applications. For instance, in Kwon et al. (2023); Bucker et al. (2023), dense trajectories were generated for a manipulator by an LLM in a zero-shot manner. The demonstration confirms the potential of LLMs to act as trajectory generators. In this work, LLMs are used to help generate trajectories of objects for 4D scene construction.

3 Method

In this section, we illustrate the components of our proposed method in detail (Fig. 3). We start by introducing some preliminaries (Sec. 3.1) on 3D Gaussians and score distillation sampling. Then we introduce our decompose-then-recompose strategy and compositional 4D scene representation (Sec. 3.2). We later illustrate the compositional score distillation involving multiple diffusion models (Sec. 3.3). Finally, we discuss how we leverage LLMs for scene decomposition including scale assignment and trajectory design (Sec. 3.4).

3.1 Preliminaries

3D Gaussian Splatting 3D Gaussian Splatting (3D-GS) Kerbl et al. (2023) parameterizes a 3D scene as a set of 3D Gaussians. Each Gaussian is defined with a center position μ, covariance Σ, opacity α, and color c modeled by spherical harmonics. Unlike implicit representation methods such as NeRF Mildenhall et al. (2021), which renders images based on volumetric rendering, 3D-GS renders images through a tile-based rasterization operation and achieves real-time rendering speed. Starting from a set of points randomly initialized in the unit sphere, each point is designated a 3D Gaussian, which can be queried as follows:

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}, \tag{1}$$

where x is an arbitrary position in the 3D scene. During the rendering process, the 3D Gaussians $G(x)$ are first transformed to 2D Gaussians $G^{'}(x)$ on the image plane. Then a tile-based rasterizer is designed to efficiently sort the 2D Gaussians and employ α-blending:

$$C(r) = \sum_{i \in N} c_i \sigma_i \prod_{j=1}^{i-1}(1-\sigma_j), \quad \sigma_i = \alpha_i G'(r), \tag{2}$$

where r is the queried pixel position, N denotes the number of sorted 2D Gaussians associated with the queried pixel, c_i and α_i denote the color and opacity of the i-th Gaussian. In our experiments, we empirically simplify the color of Gaussians to diffuse color for the sake of efficient training.

Score Distillation Sampling Current methodologies for text-to-3D or 4D generation typically involve iterative optimization of a scene representation with supervisory signals from pre-trained diffusion models Poole et al. (2022); Wang et al. (2023b). Initially, rendering of the 3D or 4D scene is acquired in the form of an image or sequence of images. Random noise is added to the rendered images, and a pre-trained diffusion model is employed to de-noise the images. The estimated gradient from this process is utilized to update the 3D or 4D representations. Specifically, employing a 3D representation parameterized by θ and a rendering method g, the rendered images are generated as $x = g(\theta)$. To align the rendered image x

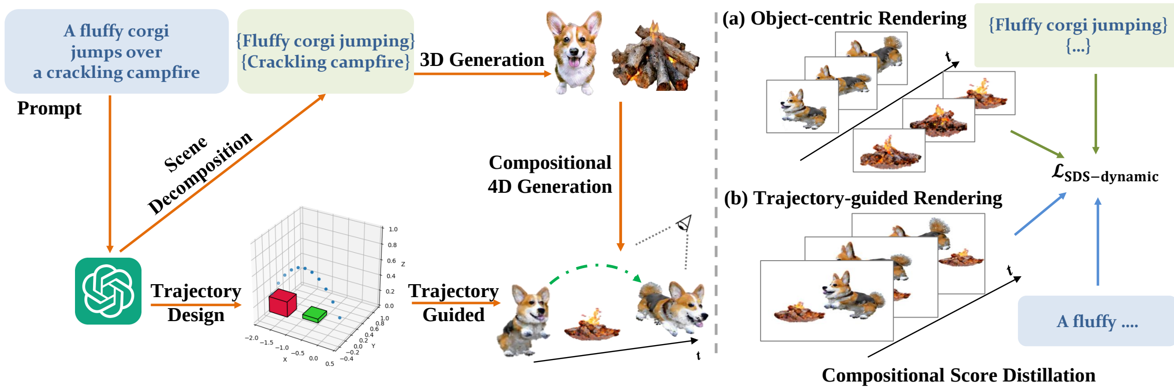

Figure 3: An overview of our proposed Comp4D method. Given an input text description, we first perform scene decomposition and obtain multiple individual 3D components. We also design the object trajectories which guide the global displacements of objects in compositional 4D scene. Thanks to the 4D representation based on 3D Gaussians, at each training iteration, we propose a compositional score distillation that switches between object-centric rendering and trajectory-guided scene rendering flexibly.

with samples obtained from the diffusion model ϕ, the diffusion model employs a score function $\hat{\epsilon}_\phi(x_t; y, t)$ to predict a noise map $\hat{\epsilon}$, given the noise level t, noisy input x_t and text embeddings y. By evaluating the difference between the Gaussian noise ϵ added to the rendered images x and the predicted noise $\hat{\epsilon}$, this score function updates the parameter θ with gradient formulated as:

$$\nabla_\theta \mathcal{L}_{SDS}(\phi, x = g(\theta)) = w(t)(\hat{\epsilon}_\phi(x_t; y, t) - \epsilon)\frac{\partial x}{\partial \theta}, \tag{3}$$

where $w(t)$ is a weighting function. Using SDS for 4D generation requires coordinated guidance to achieve realistic outcomes in terms of appearance, 3D structure, and motion Bahmani et al. (2023). This often involves the utilization of hybrid SDS, which combines both image-based and video-based diffusion models Ling et al. (2023). For our compositional 4D scene generation task, we develop a compositional SDS technique that is applied to a varying number of assets in the scene.

3.2 Compositional 4D Representation

We develop a decompose-then-recompose strategy to build compositional 4D scenes. Given a text description, we first decompose the description into multiple assets that make up the scene. Each asset is assigned a scale and a moving trajectory, either manually or through LLM models. The 4D scene is then constructed by recomposing these individual objects. In Fig. 3, we use two objects for illustration. Our framework is easily applicable to more objects.

For each object, we utilize a set of static 3D Gaussians along with an MLP-based deformation network. The MLP network takes in (x, y, z, t) coordinates as input and outputs the 3D deformation of point locations. Following previous works Tancik et al. (2020); Mildenhall et al. (2021), the input coordinates are processed with positional encoding as a 32-dimensional vector to enable high-frequency feature learning. This architecture design supports decoupled learning of the static attributes of an object (*e.g.* geometry and texture) and the local motion information. We start our training stage by optimizing the static 3D Gaussian attributes. Once they converge, we introduce the deformation field and freeze partial 3D Gaussian attributes (i.e. covariance, opacity, and color) to stabilize the training process. However, naively optimizing the deformation field leads to unpleasant results. This is primarily because the MLP modulates each point location individually, ignoring the overall rigidity of the object. Similar to AYG Ling et al. (2023), we adopt rigidity constraints to ensure that the deformation of each Gaussian is consistent with its k-nearest neighbors,

$$\mathcal{L}_{\text{rigidity}}(x) = \frac{1}{k}\sum_{i=1}^{k} ||\Delta_x - \Delta_{x_{NN_i}}||. \tag{4}$$

Moreover, to avoid flickering motion, we introduce additional regularization loss components that penalize sudden changes in the acceleration of each 3D Gaussian,

$$\mathcal{L}_{\text{acc}}(x,t) = ||\Delta_{x,t} + \Delta_{x,t+2} - 2\Delta_{x,t+1}||. \tag{5}$$

For the whole scene optimization, thanks to the explicit nature of 3D Gaussians, at rendering time, we can selectively render a single object or multiple objects, and perform compositional SDS. This enables direct and better supervision over the motion learning of each object as well as their interactions. Meanwhile, since the objects are separately represented as 3D Gaussians, we need explicit constraints to prevent the objects from intersecting with each other. If the objects have overlapping parts, the rendered image will show collapsed shapes, resulting in unstable gradients from score distillation. To this end, we draw inspiration from CG3D Vilesov et al. (2023) to incorporate a physics-based contact loss that avoids the collision of multiple objects. For one object, we ensure the contact angle θ_j for each 3D Gaussian with mean μ_j to be acute:

$$\begin{aligned} \theta_j &= (c - \mu_i) \cdot (\mu_j - \mu_i), \\ \mathcal{L}_{\text{contact}} &= -\theta_j[\theta_j < 0], \end{aligned} \tag{6}$$

where μ_i refers to the mean of the Gaussian in other objects that is closest to μ_j, and c denotes the center of the current object.

3.3 4D Scene Optimization via Compositional Score Distillation

We start the 4D scene generation by constructing each static 3D component. In the subsequent whole scene optimization, we propose compositional SDS, which involves trajectory-guided scene optimization and object-centric motion learning. We illustrate these parts in detail in the following sections.

Static 3D Object Construction To ensure both photo-realism of texture and consistent geometry, we draw inspiration from Magic123 Qian et al. (2023) and 4DFY Bahmani et al. (2023) to incorporate the joint distillations of an image diffusion Rombach et al. (2022) and a 3D-aware diffusion model Shi et al. (2023b). Specifically, we adopt the weighted combination of two sets of score distillation losses. Given a batch of rendered image x and text embeddings y, the loss function is formulated as follows,

$$\mathcal{L}_{\text{static}}(x,y) = \omega_1(\epsilon_{\text{sd}}(x_{t1}; y, t_1) - \epsilon_1) + \omega_2(\epsilon_{\text{mv}}(x_{t2}; y, t_2) - \epsilon_2) \tag{7}$$

where ω_1 and ω_2 are coefficients for the score distillation loss of Stable Diffusion Rombach et al. (2022) and MV-Dream Shi et al. (2023b).

Trajectory-Guided Scene Optimization After the initial construction of static 3D assets, we focus on the object's motion learning. At the scene level, the object's motion can be decomposed into global displacement and local deformation. The global displacement, represented by the moving trajectory, can be designed manually or by LLMs. We sample uniformly from the trajectory function, $F(\cdot)$, and obtain the object locations at arbitrary timesteps t_i. Objects are rotated accordingly such that their canonical orientation faces toward the next location along the trajectory $\overrightarrow{R_i} = (F(t_{i+1}) - F(t_i))$. Thanks to MVDream Shi et al. (2023b), which generates objects in their canonical orientation, our static stage produces objects facing the same direction (*e.g.* $\overrightarrow{R_0} = (1,0,0)$), ensuring that our rotation strategy will produce objects moving towards their head direction. Given normalized head direction $A = \frac{\overrightarrow{R_0}}{||\overrightarrow{R_0}||}$ and $B = \frac{\overrightarrow{R_i}}{\overrightarrow{R_i}}$, the axis of rotation v is obtained as $v = A \times B$. The angle of rotation θ is determined by $\cos(\theta) = A \cdot B$. We then obtain the skew-symmetric matrix $\mathbf{K}$ as follows,

$$\mathbf{K} = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}, \tag{8}$$

which is then used in Rodrigues' rotation formula to obtain the final rotation matrix $\mathbf{R}$,

$$\mathbf{R} = \mathbf{I} + (\sin\theta)\mathbf{K} + (1 - \cos\theta)\mathbf{K}^2. \tag{9}$$

Thanks to the predefined trajectory, our framework supports distilling objects with long-range motion and multi-concept interactions, which is difficult to achieve using previous baselines.

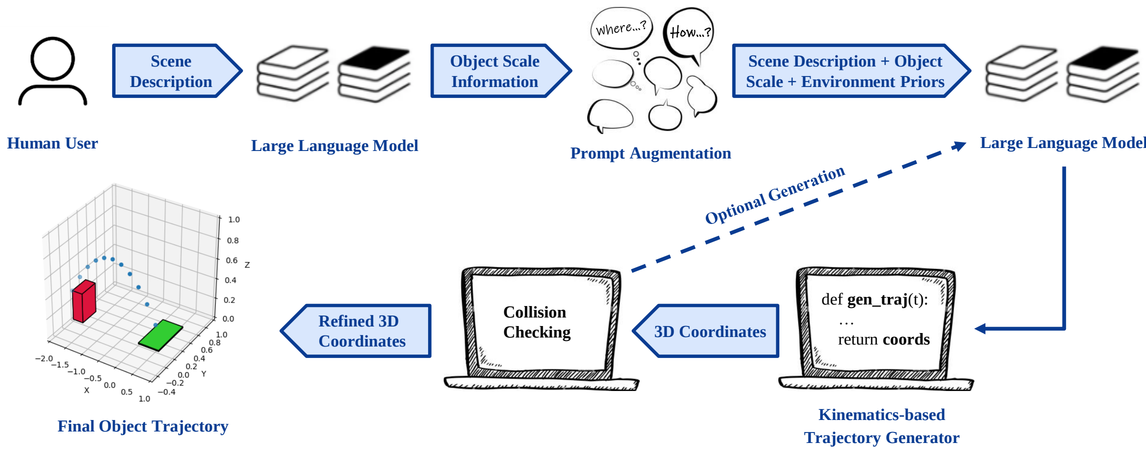

Figure 4: The pipeline for scene decomposition and trajectory design with LLMs. First, a scene description is provided by a human user as a prompt to an LLM which yields the object components as well as the relative object scales. Subsequently, the LLM is prompted with environmental constraints to return a trajectory function, which takes timestep as an input and returns the corresponding object's 3D positions. After the collection of a set of positions, collision checking is performed manually to truncate the trajectory if the collision occurs. Optionally, premature collisions can be mitigated by re-querying the LLMs for an improved trajectory function.

Besides global displacement, we utilize a deformation MLP for each set of 3D Gaussian for local motion learning. To better learn the deformation field, we leverage a text-to-video diffusion model zer (2023) to formulate the score distillation loss. Similar to distilling a static 3D object via an image diffusion model, score distillation via a video diffusion model ensures that the renderings at consecutive frames form a natural video aligned with the text prompt. As observed in previous works Bahmani et al. (2023); Ling et al. (2023), image diffusion models usually generate a more realistic appearance compared to video diffusion models. Therefore, we jointly distill the score from image diffusion on individual frames to ensure texture quality. The loss function can be formulated as follows,

$$\mathcal{L}_{\text{traj}}(x, y) = \omega_{\text{img}}(\epsilon_{\text{sd}}(x_{t1}; y, t_1) - \epsilon_1) + \omega_{\text{vid}}(\epsilon_{\text{vid}}(x_{t2}; y, t_2) - \epsilon_2), \tag{10}$$

where x is the generated image sequence of the whole scene and y is the text prompt. ω_{img} and ω_{vid} are coefficients for the score distillation loss from image- and video-based diffusion models.

Object-Centric Motion Learning Thanks to our compositional design, our framework supports arbitrary rendering combinations among the objects. This means that at each training iteration, we have the flexibility to render and optimize the whole scene or partial objects. This provides us with great freedom in rendering the scene with diverse appearances. Such diversity provides rich augmentations that are essential for the stable optimization of score distillation loss. Otherwise, the occlusion of multiple objects in the same scene will make score distillation loss ineffective in ensuring reasonable motion in occluded regions. For the single-object motion learning, the text prompt is modified by removing inactive entities to avoid disturbing learning the deformation of the current object. Following the whole scene optimization, we supervise the object-centric motion learning with joint score distillation losses (Eq. 10).

3.4 LLM Guided Scene Decomposition and Trajectory Design

We take advantage of LLMs for the scene decomposition and trajectory design. It reduces the workload on the 4D representation and enables the distillation models to focus on producing realistic local deformations. We illustrate the overall pipeline for scene decomposition and object trajectory generation in Fig. 4.

Scene Decomposition Given a text description, we first prompt an LLM to decompose the scene into multiple components and give a distinct description of each component. These descriptions serve as text prompts in object-centric motion learning. Since most 3D-aware diffusion models are trained on synthetic 3D objects that are normalized to unit scale, the resulting objects are generally of similar scales. Therefore, determining the appropriate scale of each asset becomes crucial for a realistic and reasonable composition of the scene. Recent studies Li et al. (2023a); Bubeck et al. (2023) show that ChatGPT-4 demonstrates remarkable ability in reasoning with commonsense knowledge. Therefore, we directly prompt the LLM to make reasonable assumptions of the relative scale of the objects and we adaptively resize our static 3D assets to corresponding scales.

Trajectory Design Through Kinematics Templates We further leverage the reasoning capability of ChatGPT-4 to select physics-based formulas to govern the displacement of objects. To streamline and ease the task, we instruct the model to assume that one reference object is always positioned at the camera origin and solely design the trajectory of relative displacement between the objects in the coordinate system relative to the reference object. The trajectory follows kinematrics-based equations such as uniform linear motion and parabolic motions. Furthermore, ChatGPT-4 adeptly determines a sensible initial location and velocity for the moving object, ensuring that the generated trajectory aligns with the scene description. While we take advantage of LLM to obtain trajectories, our compositional 4D scene generation framework is general and it allows human involvements to further refine or use self-designed trajectories based on customized needs.

Optional Trajectory Refinement via Collision Checking Despite curated prompt engineering, we observe that the trajectories proposed by ChatGPT-4 can be imperfect occasionally. The design trajectories may contain unexpected collisions where objects overlap with each other. Therefore, we introduce an optional trajectory refinement step as a workaround with collision checking. We uniformly sample points along the trajectory which serve as the objects' centers at corresponding timestamps. At this point, we use rectangular cuboids to simulate the objects. We rotate the objects such that the canonical orientation of the object faces the next location sampled from the trajectory. After obtaining the object placement at each timestamp, we utilize Eq. 6 to determine if there exist collisions. We then truncate the trajectory at the first collision to avoid the intersection of objects during rendering. If the truncated trajectory is too short to perform reasonable object motion, we can re-generate the trajectory by prompting ChatGPT-4 again.

4 Experiments

4.1 Implementation Details

Given a text prompt, we utilize an LLM (ChatGPT-4) or manual input to decompose the scene into multiple components, assigning scales and designing motion trajectories for each asset. Some LLM prompting examples are shown in Appendix. For asset generation, inspired by approaches like Magic123 Qian et al. (2023) and 4DFY Bahmani et al. (2023), we first create static 3D objects using joint score distillation from MVDream Shi et al. (2023b) and Stable Diffusion 2.1 Rombach et al. (2022). These objects are then converted into point clouds to initialize 3D Gaussians. By default, each object is represented with 60,000 Gaussian points. In the compositional optimization stage, we randomly assign training iterations to adopt single-object rendering (with a probability of 0.2) or whole-scene rendering (with a probability of 0.8). In each iteration, we render 16 frames via uniformly sampled timesteps. We use the frozen video diffusion model, Zeroscope zer (2023) and Stable Diffusion 2.1 Rombach et al. (2022) to provide SDS supervision. We compare with two open-source baseline text-to-4D generation methods, 4DFY Bahmani et al. (2023) and Animate124 Zhao et al. (2023). More experimentation details are provided in the Appendix.

4.2 Main Results

Quantitative Comparison We evaluate our method against baseline approaches using 20 text prompts describing diverse compositional scenes with 2-4 assets. First, we carry out a user study involving 30 participants from diverse backgrounds. Participants evaluated rendered videos of compositional 4D scenes based on four key properties, following the practice in Bahmani et al. (2023): 3D Geometry Consistency (3DC), Appearance Quality (AQ), Motion Fidelity (MF), and Text Alignment (TA). For each method, we demonstrate four views ($0°$,$90°$,$180°$, $270°$) videos for preference selection. We report the percentage of user preferences overall and for each property. Also, in the absence of ground truth for unsupervised text-to-4D scene generation, we employed non-reference quality-assessment models for images and videos. Q-Align Wu et al. (2023b) is a recently proposed large multi-modal model fine-tuned from mPLUG-Owl2 Ye et al. (2023) using in-the-wild image and video quality assessment datasets. It provides quality assessment functionality for images and videos in terms of aesthetics and quality, and it has achieved state-of-the-art performance in alignment with human ratings on existing quality assessment benchmarks. The output scores are in the range of 1 (worst) to 5 (best). We report the average scores on four views ($0°$,$90°$,$180°$, $270°$) videos of our test samples. The evaluation results are shown in Tab. 1. It can be observed that our method outperforms existing methods in all metrics by a large margin.

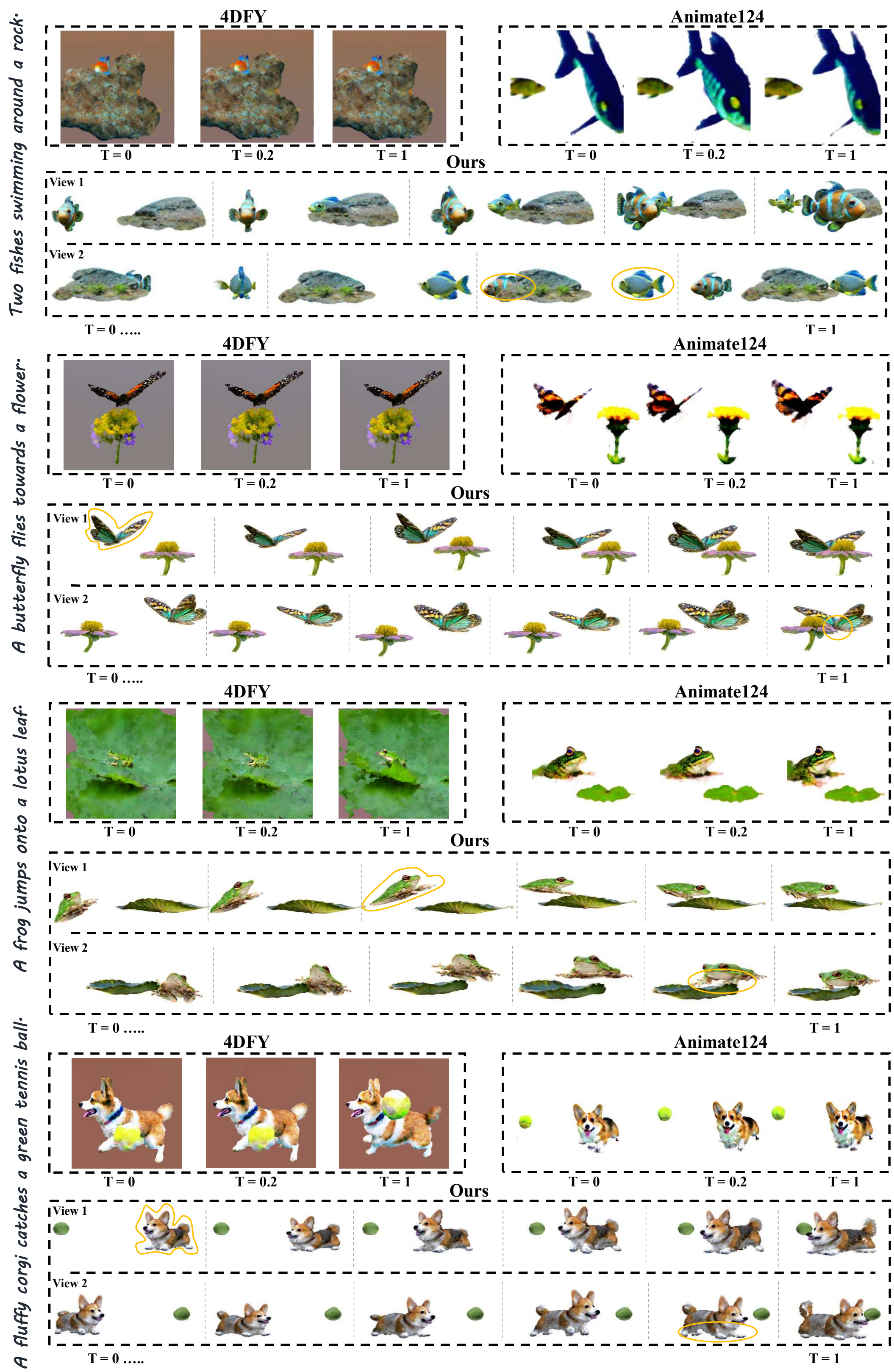

Figure 5: Comparison with previous object-centric 4D generation pipelines. Our Comp4D framework generates compositional 4D scenes involving multiple objects with more realistic motion and object interactions.

Table 1: Quantitative comparison between our method and other baselines with human preference, QAlign metrics, and rendering efficiency. The human study includes 3D geometry consistency(3DC), appearance quality(AQ), motion fidelity(MF), text alignment(TA), and overall score. QAlign metrics include quality and aesthetic evaluations on both rendered images and videos.

	Human Preference↑					*QAlign Metrics*↑				*Efficiency*↑
Method	3DC	AQ	MF	TA	Overall	Img-quality	Img-aesthetic	Vid-quality	Vid-aesthetic	Rendering FPS
4DFY Bahmani et al. (2023)	32%	28%	24%	35%	30%	2.031	1.767	2.465	1.973	4
Animate124 Zhao et al. (2023)	24%	26%	20%	26%	24%	1.434	1.484	1.948	1.654	4
Ours	**44%**	**46%**	**56%**	**39%**	**46%**	**2.931**	**2.190**	**3.367**	**2.461**	**70**

Table 2: Cumulative success rate of LLM-generated trajectory in different settings, i.e. number of objects (two, three, and four) and trajectory types (straight and curved).

Succeed at:	1st trial	2nd-3rd trial	4th-6th trial	avg # of trials
2-object-straight	100%	-	-	1.0
2-object-curved	80%	100%	-	1.3
3-object-straight	70%	90%	100%	1.6
3-object-curved	40%	70%	100%	2.5
4-object-straight	30%	70%	100%	3.2
4-object-curved	20%	40%	100%	4.4

Qualitative Comparison In Fig. 5, we provide a detailed visualization of generated scenes with multiple assets at different timesteps from various views. With the same text prompt, we compare our method with 4DFY Bahmani et al. (2023) and Animate124 Zhao et al. (2023). For these prior works, we show the scenes from one view at timestamps of 0, 0.2, and 1s. For our method, we show two views with uniformly sampled timestamps from 0 to 1s. As shown in the image, our framework excels in generating lifelike single objects with expansive motions while enhancing fidelity in object interactions. As indicated by the yellow contours in Fig. 5, we can observe the distinct flapping of butterfly wings, the changes in body shape as the frog jumps, and variations in body contours as the corgi runs. The three-object sample of two fish swimming around a rock shows large global displacements. Comparatively, the object motion in baseline methods is minimal. Going through the timeline, we can find that the objects move according to the pre-generated trajectory and display more frequent and realistic interactions. As illustrated in the yellow circles in the first two examples, the butterfly settles on the petal, and the frog stretches out its legs on the lotus leaf. In comparison, baseline methods tend to generate objects staying at the origin, with texture flickering to simulate the movement of the objects.

Resolution and Speed 4DFY Bahmani et al. (2023) conducts the video score distillation stage at a resolution of 160×288. Similarly, Animate124 Zhao et al. (2023) performs score distillation at a resolution of 80×144 due to NeRF's expensive rendering cost. Contrarily, our method can render video at a resolution of 320×576 during score distillation which aligns with the training resolution of video diffusion zer (2023) and facilitates superior motion generation. At inference time, thanks to the efficient Gaussian representation, our 4D scene representation renders at 70 FPS at 320×576. In comparison, 4DFY and Animate124 render at around 4 FPS.

Robustness of LLM model in trajectory design Our framework incorporates collision checking, trajectory truncation, and re-generation, to enhance the robustness of the generating trajectory by LLM. We evaluate the success rate of trajectory generation using an LLM model on 10 two-object scenes, 10 three-object scenes, and 10 four-object scenes. For each scene, ChatGPT-4 is prompted to generate one straight and one curved path. A trajectory is deemed successful if it avoids collisions. The cumulative success rate for each case w.r.t. the number of trials is shown in Tab. 2. For simpler scenarios with two objects moving along straight paths, the framework achieves a 100% success rate on the first trial. For more complex cases, such as three objects with curved trajectories, the success rate is 40% on the first trial, requiring an average of 2.5 trials to generate valid paths. In the most challenging scenario involving four objects with curved paths, an average of 4.4 trials is needed for successful trajectory generation. Overall, the LLM demonstrates strong performance in generating collision-free trajectories, even for complex multi-object scenarios. At this point, we emphasize that our method is flexible and can seamlessly incorporate human involvement during the initial trajectory design stage, ensuring success even for challenging prompts where the LLM may fall short.

Table 3: Ablation studies on our proposed components. We employ QAlign metrics including quality and aesthetic evaluations on both rendered images and videos.

Settings	QAlign-Img-quality↑	QAlign-Img-aesthetic↑	QAlign-Vid-quality↑	QAlign-Vid-aesthetic↑
w/o Single	1.8252	1.6455	2.4082	1.9062
w/o Joint	1.9893	1.8789	2.4102	1.9512
w/o Image	1.8613	1.7715	2.3926	1.9014
Fewer GS	1.9131	1.8301	2.7285	2.0039
Full	**2.4785**	**1.9004**	**2.9023**	**2.1621**

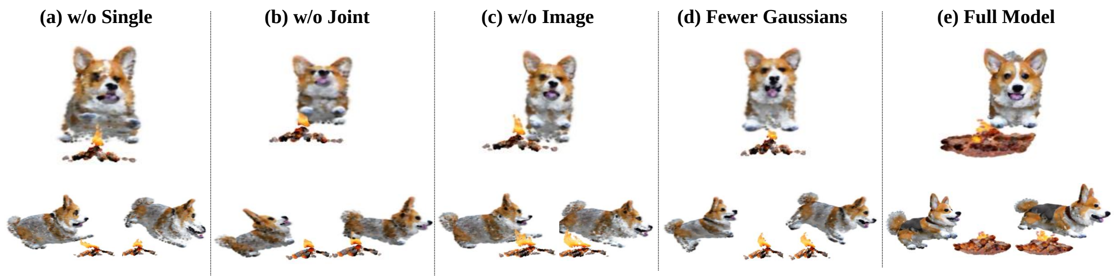

Figure 6: Ablation studies on the proposed components. The first row shows the front view. The second row shows the side view. Note that (a)-(d) are conducted using fewer number of Gaussians.

4.3 Ablation Studies

We evaluate the effectiveness of all the proposed components in Fig. 6 and Tab. 3. To save computation costs, we utilize 3D Gaussians containing 20,000 points to represent each object during the ablation study. In Fig. 6(a), "w/o Single" refers to the variant without object-centric rendering. We observe the worst object geometry possibly due to the occlusions occurring in the optimization process. "w/o Joint" in Fig. 6(b) denotes that we only perform object-centric rendering without rendering two objects altogether in the same scene. The final results exhibit decreased motion and reduced interactions. In Fig. 6(c), we remove the SDS loss from image diffusion and only distill with video diffusion. Consequently, we observe that objects appear to have poor textures compared to (d) where image diffusion SDS loss is included. In Fig. 6(d), the model training and losses are kept the same as the full model (e), except that the number of 3D Gaussians we generate in the static stage is fewer. As shown in the figure, using fewer Gaussians results in less detailed texture and less realistic geometry. In summary, our full model (e) delivers the best results both quantitatively and qualitatively.

5 Limitations and Future Work

Despite the exciting results produced by Comp4D, our framework still has some limitations. First, we are leveraging the zero-shot ability of ChatGPT-4, which can be further improved if the language model is fine-tuned to generate a more precise trajectory and more complex motion. Although the LLM model performs well in trajectory design in relatively simple scenes with several objects, we observe it tends to fail with more objects due to the potential collisions. In such circumstances, our framework requires human involvement for the trajectory design. Second, the generated motions are currently limited by the capabilities of video diffusion models. Future work will focus on extending the motion duration and complexity to support more practical and diverse 4D content creation.

6 Conclusion

In this work, we present Comp4D, a novel framework for generating compositional 4D scenes from text input. With the help of ChatGPT-4, we decompose scene generation into the creation of individual objects as well as their interactions. Given a compositional scene description, we first leverage GPT-4 to generate object prompts for the independent creation of 3D objects. Subsequently, it is tasked to design the trajectory for the moving objects. This predefined trajectory then guides the compositional score distillation process, which optimizes a composable 4D representation comprising deformable 3D Gaussians for each object. Our experiments demonstrate that Comp4D significantly surpasses existing text-to-4d generation methods in terms of visual quality, motion fidelity, and object interactions.

References

Zeroscope. `https://huggingface.co/cerspense/zeroscope_v2_576w`, 2023.

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. *arXiv preprint arXiv:2311.17984*, 2023.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Arthur Bucker, Luis Figueredo, Sami Haddadin, Ashish Kapoor, Shuang Ma, Sai Vemprala, and Rogerio Bonatti. Latte: Language trajectory transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7287–7294. IEEE, 2023.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023a.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023b.

Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d gaussian splatting: Towards efficient novel view synthesis for dynamic scenes. *arXiv preprint arXiv:2402.03307*, 2024.

Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *European Conference on Computer Vision*, pp. 624–642. Springer, 2022.

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as I can, not as I say: Grounding language in robotic affordances. In *Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318, Auckland, New Zealand, 2022. PLMR.

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

Kai Katsumata, Duc Minh Vo, and Hideki Nakayama. An efficient 3d gaussian representation for monocular/multi-view dynamic scenes. *arXiv*, 2023.

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

Teyun Kwon, Norman Di Palo, and Edward Johns. Language models as zero-shot trajectory generators. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *Conference on Computer Vision and Pattern Recognition*, pp. 10955–10965, New Orleans, LA, USA, 2022. IEEE.

Yunxin Li, Longyue Wang, Baotian Hu, Xinyu Chen, Wanqi Zhong, Chenyang Lyu, and Min Zhang. A comprehensive evaluation of gpt-4v on knowledge-intensive visual question answering. *arXiv preprint arXiv:2311.07536*, 2023a.

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. *arXiv preprint arXiv:2312.16812*, 2023b.

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

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

Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. *arXiv preprint arXiv:2312.13763*, 2023.

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. *arXiv preprint arXiv:2303.11328*, 2023.

Yujie Lu, Pan Lu, Zhiyu Chen, Wanrong Zhu, Xin Eric Wang, and William Yang Wang. Multimodal procedural planning via dual text-image prompting, 2023.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*, 2019.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

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.

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*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=08Yk-n5l2Al.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. *arXiv preprint arXiv:2309.16534*, 2023.

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*, 2023a.

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

Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. Text-to-4d dynamic scene generation. *arXiv preprint arXiv:2301.11280*, 2023.

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.

Archana Tikayat Ray, Anirudh Prabhakara Bhat, Ryan T. White, Van Minh Nguyen, Olivia J. Pinon Fischer, and Dimitri N. Mavris. Examining the potential of generative language models for aviation safety analysis: Case study and insights using the aviation safety reporting system (asrs). *Aerospace*, 10(9), 2023.

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

Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12619–12629, 2023a.

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. *arXiv preprint arXiv:2305.16213*, 2023b.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023a.

Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*, 2023b.

Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360 {\deg} views. *arXiv preprint arXiv:2211.16431*, 2022.

Yunhao Yang, Neel P Bhatt, Tyler Ingebrand, William Ward, Steven Carr, Zhangyang Wang, and Ufuk Topcu. Fine-tuning language models using formal methods feedback. *arXiv preprint arXiv:2310.18239*, 2023a.

Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023b.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv preprint arXiv:2311.04257*, 2023.

Yuyang Yin, Dejia Xu, Zhangyang Wang, Yao Zhao, and Yunchao Wei. 4dgen: Grounded 4d content generation with spatial-temporal consistency. *arXiv preprint arXiv:2312.17225*, 2023.

Yuyang Zhao, Zhiwen Yan, Enze Xie, Lanqing Hong, Zhenguo Li, and Gim Hee Lee. Animate124: Animating one image to 4d dynamic scene. *arXiv preprint arXiv:2311.14603*, 2023.

Yufeng Zheng, Xueting Li, Koki Nagano, Sifei Liu, Otmar Hilliges, and Shalini De Mello. A unified approach for text-and image-guided 4d scene generation. *arXiv preprint arXiv:2311.16854*, 2023.

A Additional Implementation Details

Similar to Magic123 Qian et al. (2023) and 4DFY Bahmani et al. (2023), we first generate the static 3D objects via NeRF representation using joint score distillation from MVDream Shi et al. (2023b) and Stable Diffusion 2.1 Rombach et al. (2022). After obtaining the static objects, we convert them to point clouds which are consecutively used to initialize 3D Gaussians. Our full model utilizes point clouds containing 60,000 colored points. We preprocess 60 nearest neighbors for each 3D Gaussian in order to speed up the calculation of $\mathcal{L}_{\text{rigidity}}$. In the dynamic optimization stage, we randomly assign training iterations to adopt single-object rendering (with a probability of 0.2) or compositional rendering (with a probability of 0.8). We train for 3,000 iterations in the dynamic stage with a learning rate of 1e-4 for the deformation MLP. In each iteration, we render 16 frames via uniformly sampled timesteps. We use the frozen video diffusion model, Zeroscope zer (2023), in our experiments. To improve the 2D appearance, we also randomly sample 4 frames out of 16 rendered frames for image score distillation, where the Stable Diffusion 2.1 Rombach et al. (2022) is used as the image diffusion model. In our experiments, we compare with two open-source baseline text-to-4D generation methods, 4DFY Bahmani et al. (2023) and Animate124 Zhao et al. (2023).

Our overall loss function can be summarized as follows,

$$\mathcal{L} = \mathcal{L}_{\text{SDS-static}} + \mathcal{L}_{\text{SDS-dynamic}} + \mathcal{L}_{\text{reg}}, \tag{11}$$

where $\mathcal{L}_{\text{reg}}$ refers to regularizations:

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{contact}} + \omega_1 \mathcal{L}_{\text{acc}} + \omega_2 \mathcal{L}_{\text{rigidity}}. \tag{12}$$

ω_1 and ω_2 are weighting coefficients, set to $1e-4$ and $1e3$, respectively. For MVDream Shi et al. (2023b), we use negative prompt as their default configuration, "ugly, bad anatomy, blurry, pixelated obscure, unnatural colors, poor lighting, dull, and unclear, cropped, lowres, low quality, artifacts, duplicate, morbid, mutilated, poorly drawn face, deformed, dehydrated, bad proportions". For Zeroscope zer (2023), we use negative prompts that avoid static generations, "static, low motion, static statue, not moving, no motion, text, watermark, copyright, blurry, nsfw".

B Baselines Implementations

For baseline methods 4DFY Bahmani et al. (2023) and Animate124 Zhao et al. (2023), we use the codebase released by the authors. For 4DFY Bahmani et al. (2023), since it is a text-to-4D generation method, we use the same scene text description when generating 4D results. For Animate124 Zhao et al. (2023), due to its image-to-4D nature, we use the static assets produced by our method to obtain the initial image for Animate124. Specifically, we place the static assets at the $t = 0$ position in the trajectory and render a static image that does not contain occlusions. Then, we follow the workflow of Animate124 first to perform textual inversion and then the following static and dynamic stages.

C Automatic scene decomposition with the LLM model

In our framework, we take advantage of an LLM model (ChatGPT-4) to automatically decompose the scene into multiple assets, and design their moving trajectories. We visualize this procedure in Fig. 7 and Fig. 8, where we provide a conversation between a human user and ChatGPT-4 to accomplish this task. The output of ChatGPT includes a function that delineates the moving trajectory of the object. We also visualize the trajectory in Fig. 9. Note that in the circumstances where the scene is too complex for ChatGPT to handle, our framework supports human involvement to manually decompose the scene and design the trajectories.

D Additional Visualizations

We provide more results and comparisons in the video.

Consider the scene: "A fluffy corgi jumps over a crackling campfire." Provide the descriptions for each object in the sentence with the adjective associated with it. Infer from the prompt which object is moving and which is static. Provide your output in this form: adjective + object (moving or static).

GPT-4

There are two objects in the sentence. Here are the two objects in the scene with their associated adjectives:

1. **A fluffy corgi (moving)**
2. **Crackling campfire (static)**

Now, based on the object information you provided and assuming the static object is always located at the world origin, first, make reasonable assumptions of the object scales in 3D. Make sure both objects are not too large or too small and fit in the [0, 1] range. Pick a reasonable starting location and velocity for the moving object.

GPT-4

Let's assume the following reasonable 3D scales, starting location, and velocity for the objects:

Object Scales in 3D:

1. **Fluffy Corgi (moving)**:
   - Scale: [0.6, 0.3, 0.2]
   - (Represents the approximate dimensions of the corgi: length, width, and height.)
2. **Crackling Campfire (static)**:
   - Scale: [0.5, 0.3, 0.05]
   - (Represents the approximate dimensions of the campfire: diameter and height.)

Starting Location and Velocity for the Moving Object (Corgi):

- **Starting Location**: [-1.0, 0.0, 0.05]
  - (The corgi starts on the ground some distance away from the campfire along the x-axis, at ground level.)
- **Velocity**: [3,5, 0.0, 3.0]
  - (The corgi moves straight forward along the x-axis while jumping upwards along the z-axis.)

These assumptions position the corgi so that it is jumping over the campfire, with the velocity directing its motion forward and upward. The scales ensure that both objects fit within the [0, 1] range and are reasonably proportioned relative to each other.

Using the objects and their scales, initial positions, and velocity, use physics-based formulas to represent the 3D locations of the moving object for a duration of 1 second. In addition, ensure the following: 1. If the moving object collides with the static object placed at the origin during the 1 second, return the timestep where collision occurs. 2. The moving object trajectory should be limited to a range of [-2, 2] through scaling, but not clipping. Give me a python function named `generate_coordinates` that takes timestep as input and outputs 3D coordinates of the moving object as np.array and the object scales as a list. Use x-y plane as horizontal and use z axis as the vertical axis.

Figure 7: Automatic scene decomposition and trajectory design with ChatGPT-4 (part 1). We provide an example of the conversation between a human user and ChatGPT-4 for scene decomposition and trajectory design. Given a text description, "A fluffy corgi jumps over a crackling campfire", the human user first prompts the ChatGPT-4 to extract the component objects from the description, followed by scale assignment over the component objects. Then, the human user prompts the ChatGPT-4 to give initial positions, velocities, and moving trajectories of the objects.

GPT-4

Here's a Python function named **`generate_coordinates`** that implements the described behavior:

python

```
import numpy as np

def generate_coordinates(timestep):
    # Flag to indicate collision
    done = False

    # Object scales
    corgi_scale = [0.6, 0.3, 0.2]
    campfire_scale = [0.5, 0.3, 0.05]

    # Initial conditions for the corgi
    initial_position = np.array([-1.0, 0.0, 0.1])
    velocity = np.array([3.5, 0.0, 3.0])
    gravity = np.array([0.0, 0.0, -9.81])

    # Calculate position
    position = initial_position + timestep * velocity + 0.5 * (timestep**2) * gravity

    # Check for collision
    if (abs(position[0]) <= campfire_scale[0]) and (abs(position[1]) <= campfire_scale[1]):
        if (position[2] <= campfire_scale[2]):
            print(f"Collision occurs at t = {timestep}s")
            done = True

    # Scale position to fit in [-2, 2] range
    position = np.clip(position, -2, 2)

    return position, [corgi_scale, campfire_scale], done

# Example usage
timestep = 1.0
position, scales, collision = generate_coordinates(timestep)
print("Position:", position)
print("Scales:", scales)
print("Collision Detected:", collision)
```

This function generates the 3D coordinates of the corgi at a given timestep, considering its initial position, velocity, and the possibility of collision with the static object (campfire). It also ensures that the corgi's trajectory remains within the specified range of [-2, 2] through scaling.

Figure 8: Automatic scene decomposition and trajectory design with ChatGPT-4 (part 2).

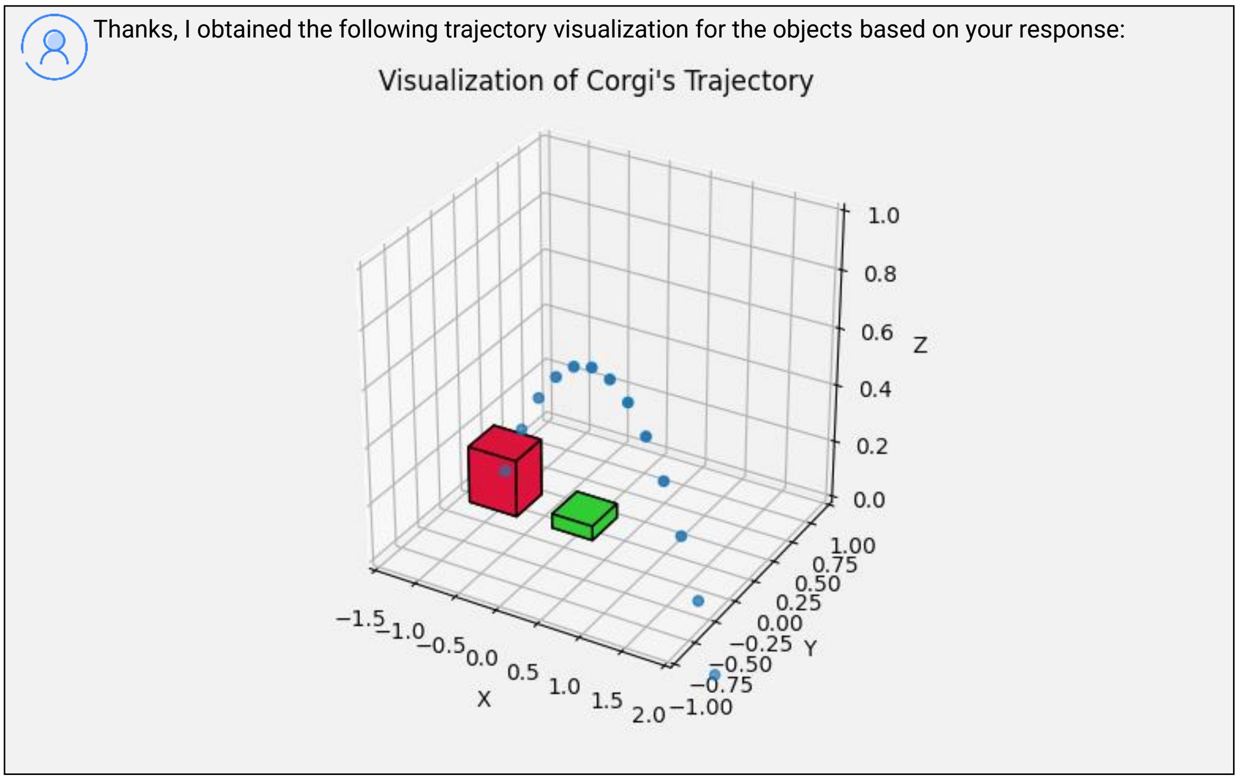

Figure 9: Automatic scene decomposition and trajectory design with ChatGPT-4.