

PaintScene4D: Consistent 4D Scene Generation from Text Prompts

Vinayak Gupta¹Yunze Man²Yu-Xiong Wang²¹ Indian Institute of Technology, Madras ² University of Illinois Urbana-Champaign

<https://paintscene4d.github.io/>

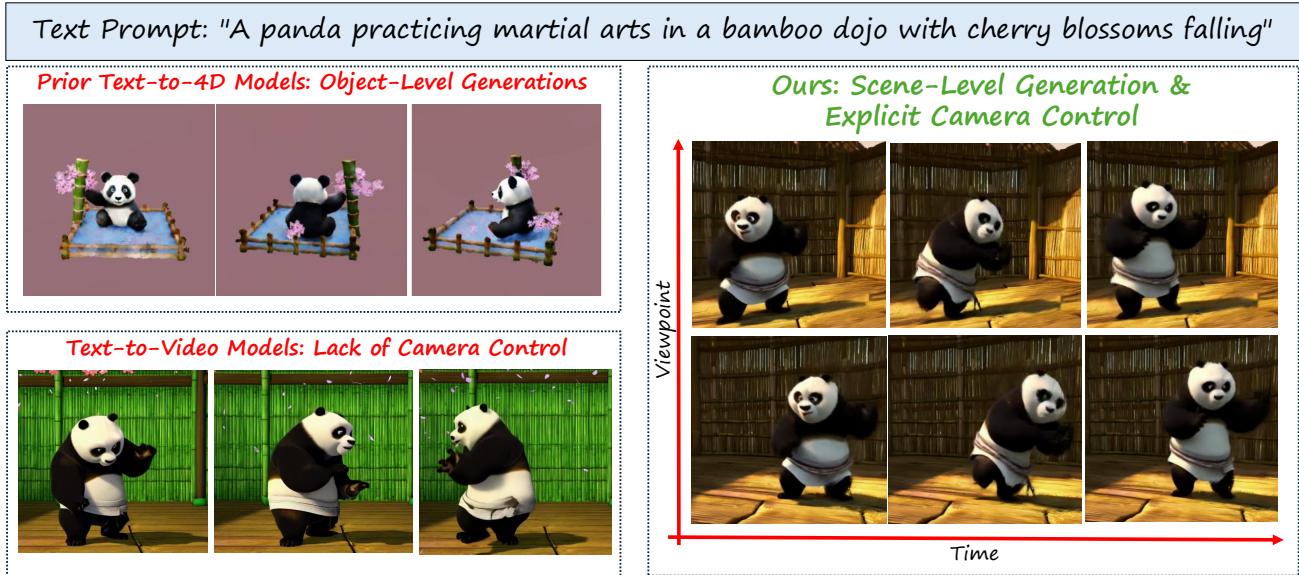


Figure 1. **4D Text-to-Scene Generation:** Unlike prior methods that restrict text-to-4D generation to object-level reconstruction or text-to-video models lacking explicit camera control, our approach reconstructs full 4D scenes from a single text prompt. This leads to realistic 4D scenes that can be viewed from different trajectories, achieving via an efficient training-free architecture.

Abstract

Recent advances in diffusion models have revolutionized 2D and 3D content creation, yet generating photorealistic dynamic 4D scenes remains a significant challenge. Existing dynamic 4D generation methods typically rely on distilling knowledge from pre-trained 3D generative models, often fine-tuned on synthetic object datasets. Consequently, the resulting scenes tend to be object-centric and lack photorealism. While text-to-video models can generate more realistic scenes with motion, they often struggle with spatial understanding and provide limited control over camera viewpoints during rendering. To address these limitations, we present PaintScene4D, a novel text-to-4D scene generation framework that departs from conventional multi-view generative models in favor of a streamlined architecture that harnesses video generative models trained on diverse real-world datasets. Our method first generates a reference

video using a video generation model, and then employs a strategic camera array selection for rendering. We apply a progressive warping and inpainting technique to ensure both spatial and temporal consistency across multiple viewpoints. Finally, we optimize multi-view images using a dynamic renderer, enabling flexible camera control based on user preferences. Adopting a training-free architecture, our PaintScene4D efficiently produces realistic 4D scenes that can be viewed from arbitrary trajectories. The code will be made publicly available.

1. Introduction

Generating dynamic 3D scenes from text descriptions, known as text-to-4D scene generation, represents one of the most challenging frontiers in computer vision and graphics. While recent advances have revolutionized our ability to create static 3D content [23, 28, 29, 31, 33, 34, 42] and

2D image and videos [38, 39, 43, 49] from text, the synthesis of temporally coherent and animated 3D scenes remains a fundamental challenge. This task requires not only generating spatially consistent 3D geometry and appearance, but also producing realistic motion that adheres to real-world physics and semantic constraints – all while maintaining temporal consistency across multiple viewpoints.

The complexity of 4D scene generation stems from several interconnected challenges. First, unlike static 3D generation, which only needs to ensure spatial consistency, 4D scenes must maintain both spatial and temporal coherence simultaneously. This means that any generated motion must be physically plausible and semantically meaningful, while preserving the scene’s geometric structure across time. Second, the lack of large-scale, diverse 4D scene datasets has limited the development of robust generation methods, with most existing approaches relying on object-centric data that fail to capture the rich dynamics of full scenes. Third, the computational complexity of optimizing both spatial and temporal dimensions makes it difficult to achieve high-quality results within reasonable time constraints.

Current approaches to these challenges broadly fall into two categories, each with significant drawbacks. The first category extends static 3D generation methods (*e.g.*, MV-Dream [42] and Zero123 [28, 41], trained on object-centric datasets like Objaverse [9]) to incorporate temporal dynamics [2, 19, 25, 36, 44, 56, 61, 62]. These methods, while effective at maintaining geometric consistency, typically struggle with generating complex motion, often producing only subtle deformations or simple translations. This limitation stems from their reliance on Score Distillation Sampling (SDS) [33] optimization, which, while effective for static content, becomes computationally intractable when scaled to temporal sequences. The second category are the Text-to-Video models (*e.g.*, Animatediff [13], CogVideo [54]) that generate dynamic content. However, these approaches lack explicit 3D understanding, resulting in temporal inconsistencies and geometric artifacts. Neither approach adequately addresses the fundamental challenge of generating spatially and temporally coherent 4D scenes.

To address these limitations, we present **PaintScene4D**, a novel framework that harness the strengths of both Text-to-Video generation and 3D-aware neural rendering. *Our key insight* is that by using video generation as an initial prior and reconstructing the 3D scene through a *progressive warping and inpainting technique*, we can maintain both spatial and temporal consistency while enabling complex motion generation. Specifically, our method first generates a base video using a pretrained Text-to-Video model, which provides rich motion priors. We then construct a “web of cameras” around the scene by warping video frames to nearby viewpoints and using inpainting to fill any resulting gaps. This approach allows us to build a comprehensive

multi-view representation of the dynamic scene *without* requiring explicit 3D supervision or costly optimization procedures.

The effectiveness of PaintScene4D is demonstrated through several empirical contributions. As shown in Figure 1, our method achieves state-of-the-art results in text-to-4D scene generation, producing visually compelling results that maintain both spatial and temporal consistency. The generated scenes exhibit complex motion while preserving geometric structure across multiple viewpoints. Notably, our framework reduces the computational requirements significantly, generating high-quality 4D content in approximately 3 hours on a single A100 GPU – a substantial improvement over existing methods [2, 62] that often require 10+ hours. Through extensive experiments and ablation studies, we demonstrate the superiority of our approach across various metrics, including temporal consistency, motion complexity, and rendering quality. *Our method also offers unprecedented flexibility, allowing users to edit existing videos or specify custom trajectories during inference.*

Our main contributions can be summarized as follows:

- A novel end-to-end framework for text-to-4D scene generation that effectively distills video generation prior to 4D-aware neural rendering.
- A progressive warping and inpainting technique that enables the construction of spatially and temporally consistent multi-view representations.
- Comprehensive evaluation and analysis of PaintScene4D demonstrates state-of-the-art results in 4D scene generation, with significantly reduced computational requirements and enhanced camera control options.

2. Related Work

Text-to-3D Generation. Text-to-3D generation has evolved significantly over the past decades. Initial approaches rely on rule-based systems that parse text inputs into semantic representations for scene generation using object databases [1, 6, 8]. The field has advanced substantially with the introduction of data-driven approaches that leverage multimodal datasets [7] and pretrained models like CLIP [35], enabling more sophisticated manipulation of 3D meshes [11, 18] or radiance fields [48]. This progress has led to the development of methods utilizing CLIP-based supervision for comprehensive 3D scene synthesis [17, 40], which subsequently evolves into techniques that optimize meshes and radiance fields through Score Distillation Sampling (SDS) [22, 33, 50]. The introduction of multi-view-aware diffusion models has further enhanced the quality of generated 3D structures [24, 28, 42]. Parallel developments in diffusion and transformer architectures have enabled advanced image-to-3D conversion for novel view synthesis [5, 12, 29, 34, 45, 47, 57]. These approaches primarily address object-level reconstruction.

Recent advances in text-to-3D scene generation have introduced innovative approaches to address scene-level complexity. Text2Room [14] proposes a warping and inpainting methodology for mesh population and scene creation, while Text2NeRF [59] shifts away from mesh-based reconstruction to utilize radiance fields as scene generation priors. Although these methods are initially constrained to camera-centric scenes, subsequent work [60] expands capabilities to support general 3D scene generation with arbitrary 6 degree-of-freedom (DOF) camera trajectories. However, these approaches remain limited to static scenes, lacking the ability to incorporate motion, which is a crucial element for representing dynamic, real-world environments.

Object-centric Text-to-4D Generation. The extension from 3D to 4D scene generation introduces significant additional complexity. MAV3D [44] pioneers this direction by introducing a dynamic NeRF representation using Hex-Plane [4] and video-based SDS loss, structuring the generation process into static, dynamic, and superresolution phases. Building on this foundation, Dream-in-4D [63] employs a deformation field-based dynamic NeRF, organizing text-to-4D generation into distinct static and dynamic phases. Similarly, 4D-fy [2] introduces a hybrid representation that combines static and dynamic voxels, leveraging multiple SDS loss functions [30, 33, 42] to enable dynamic content generation. Additionally, Ling et al. [26] achieve dynamic rendering through the application of a dynamic network to 3D Gaussian Splatting (3D-GS), optimized under both video SDS and 3D Gaussian constraints.

Recent developments have focused on decomposing and controlling motion generation. TC4D [3] separates motion into global trajectories, represented by spline curves, and local object motion, with global paths being user-defined and local motion generated in segments using video-based SDS loss. Comp4D [53] employs large language models LLMs to decompose prompts into entities, generating and combining 4D objects based on LLM-derived trajectory information. These approaches primarily focus on object-level reconstruction, limiting their broader applicability. A notable departure from this trend is 4Real [58], which circumvents multi-view generative models by leveraging video generative models trained on large-scale datasets. Our work builds upon these advances while addressing their limitations, proposing a method for generating 4D scenes that captures both geometry and texture in real-world environments while offering enhanced control over camera movement and rendering capabilities.

3. Preliminary: 3D Gaussian Splatting

3D Gaussian Splatting (3D-GS) [21] utilizes point clouds to explicitly represent scenes, with each point modeled as a Gaussian distribution in three-dimensional space. Each Gaussian is defined by a central point \mathcal{X} , representing its

mean, and a covariance matrix Σ , capturing its spread. The Gaussian function $G(\mathcal{X})$ for a point is represented by:

$$G(\mathcal{X}) = \exp\left(-\frac{1}{2}\mathcal{X}^T\Sigma^{-1}\mathcal{X}\right). \quad (1)$$

To enable differentiable optimization, the covariance matrix Σ is decomposed into a rotation matrix \mathbf{R} and a scaling matrix \mathbf{S} as $\Sigma = \mathbf{R}\mathbf{S}\mathbf{S}^T\mathbf{R}^T$. Rendering novel views involves a differentiable splatting technique [55], where 3D Gaussians are projected onto a 2D image plane. As detailed in [64], the transformed covariance Σ' in camera coordinates is computed using the viewing transformation matrix W and the Jacobian J of the projection, yielding $\Sigma' = JW\Sigma W^T J^T$. In this setup, each Gaussian is parameterized by its position $\mathcal{X} \in \mathbb{R}^3$, its color defined by spherical harmonic (SH) coefficients $\mathcal{C} \in \mathbb{R}^k$ (where k is the number of SH functions), opacity $\alpha \in \mathbb{R}$, rotation $r \in \mathbb{R}^4$, and scale $s \in \mathbb{R}^3$. The resulting pixel color and opacity are derived from the Gaussian blend defined in Equation 2, where the color C from N overlapping Gaussians is calculated as:

$$C = \sum_{i=1}^N c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (2)$$

where c_i and α_i represent the color and opacity of the i -th Gaussian point, with these values optimized per-point from the Gaussian's SH coefficients and adjustable opacity.

4. Method

Overview. In this work, we present PaintScene4D, a novel framework designed to generate 4D dynamic scenes from textual inputs. Our approach begins with a video diffusion model that produces an initial video serving as both scene and motion reference. Using this video as input, we employ a depth estimation model to derive depth maps from each frame, allowing us to progressively construct a spatial representation of the scene. To create a comprehensive multi-camera view of the scene, we progressively warp the initial frames to new camera positions, beginning with the first frame. In this process, any regions missing due to occlusion or perspective changes are filled in using a spatially consistent inpainting method. For each subsequent frame, our approach reuses inpainted data from prior timestamps for continuity, and only filling in new, unobserved areas. Once we have constructed a network of cameras, where each camera captures all frames over time, we employ a 4D rendering algorithm to reconstruct the scene and generate novel viewpoints. This entire methodology is outlined in Figure 2.

4.1. Scene Initialization

Reference Video Generation and Depth Estimation. To generate the initial scene content from an input prompt t , we start by applying a pre-trained video diffusion model, f_d , conditioned on t to create an initial video $V_0 = f_d(\epsilon \mid t)$,

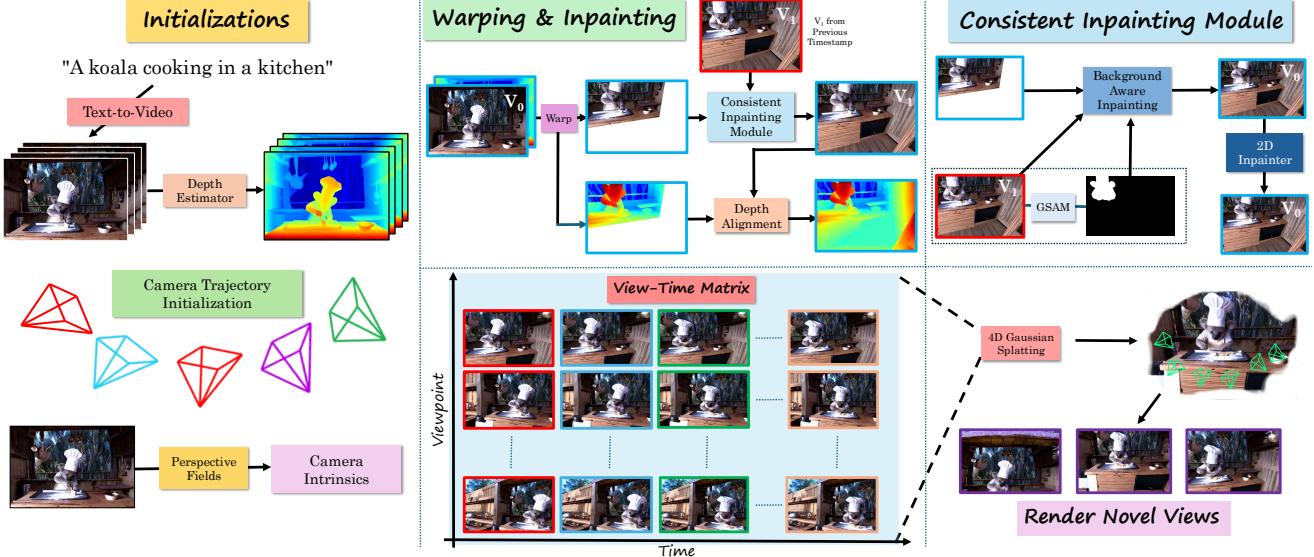


Figure 2. Method Overview. Our approach consists of three stages. First, we initialize the 4D scene using a diffusion prior to establish scene content and motion, estimate depth maps for each video frame, and initialize camera trajectory (extrinsics) and intrinsics for subsequent warping. In the second stage, we perform sequential warping and inpainting from the first timestamp. To ensure spatial and temporal coherence, our consistent inpainting module mitigates artifacts and aligns depth maps, preventing error accumulation. Finally, the generated view-time matrix is used to renders novel views along user-defined camera trajectories, allowing for explicit camera control. V_0, V_1 represent different view points.

where ϵ is random Gaussian noise. Given our approach requires the video to be captured using a stationary, non-moving camera, we enhance the user-defined prompt with additional descriptors, such as “The camera remains stationary, with a fixed frame, stable composition, and no shifts.” This added specificity ensures that the output video aligns with our fixed-camera requirement. As V_0 does not contain inherent geometric depth information, we integrate a video depth estimation model f_e to obtain this depth data, generating depth maps $D_0 = f_e(V_0)$. The video frames in V_0 paired with the corresponding depth maps D_0 serve as the basis for initializing the 4D scene.

Camera Trajectory. To support the intended trajectory of the final rendered output, we establish a network of virtual cameras to match the user’s desired camera path. These cameras represent a structured arrangement of views that form the backbone for the construction of the 4D scene. Given that our framework incorporates warping operations, it is imperative to obtain accurate intrinsics of the camera parameters. However, the utilization of video generated by a diffusion model presents a challenge, as direct access to the intrinsic camera parameters is not available. To address this shortcoming, we employ a pre-trained model, Perspective Field [20], to calculate the intrinsic matrix based on the video frames provided.

4.2. Scene Construction w/ Progressive Inpainting

Given the absence of multi-view supervision, directly employing a single-view video V_0 and its depth maps D_0 to

train a 4D radiance field can lead to issues of overfitting and geometric ambiguity. To address this, we apply a depth image-based rendering (DIBR) technique [10] to establish a network of virtual cameras around the initial view. Specifically, for each pixel p in I_i^t and its corresponding depth z in D_i^t , we compute its transformed coordinates $p_{i \rightarrow j}$ and depth $z_{i \rightarrow j}$ for a neighboring viewpoint j as follows:

$$[p_{i \rightarrow j}, z_{i \rightarrow j}]^T = \mathbf{K} \mathbf{P}_j \mathbf{P}_i^{-1} \mathbf{K}^{-1} [p, z]^T, \quad (3)$$

where \mathbf{K} , \mathbf{P}_i and \mathbf{P}_j are the intrinsic matrix, camera pose for view i and view j , respectively and I_i^t represents the image at the timestamp t of viewpoint i . Following the transformation, we fill occluded or missing regions in the newly warped views with inpainting. Our experimental findings reveal that the diffusion-based prior used for inpainting yields higher quality results when the inpainted regions are larger. Therefore, for each view, we select the *farthest available viewpoint with minimal overlap*, warp the current frame to this viewpoint, and apply inpainting as necessary. Large occlusions are filled using a 2D diffusion-based prior, while smaller gaps are addressed with Telea-based inpainting [46].

Our warping process begins at the first timestamp, progressively warping and inpainting frames across all views before proceeding to subsequent timestamps. For the first timestamp, we start with a base view I_0^0 , warp it to a neighboring viewpoint I_1^0 , and inpaint any missing regions. To ensure spatial consistency, we integrate both the original (I_0^0) and newly warped frames (I_1^0) for further warping (e.g., I_2^0, I_3^0). This approach ensures that any inpainted content in

I_1^0 is preserved in subsequent viewpoints (I_2^0, I_3^0, \dots), maintaining coherence throughout the scene.

Depth Alignment. To transform a 2D image I into a 3D representation, we first estimate the depth for each pixel. Accurate integration of both new and existing content requires precise depth alignment, ensuring that similar elements in the scene, such as walls or furniture, appear at consistent depths across views. Directly projecting the predicted depth often results in abrupt transitions and geometric discontinuities due to inconsistent scale across viewpoints. To address this, we apply a depth alignment procedure inspired by Liu et al. [27], which refines the depth through scale and shift optimization. Specifically, we optimize scale γ and shift β parameters, $\gamma, \beta \in \mathbb{R}$, by minimizing the difference between predicted \hat{d} and rendered depths d in a least-squares sense:

$$\min_{\gamma, \beta} \|m \odot (\gamma \hat{d} + \beta - d)\|^2, \quad (4)$$

where the mask m excludes unobserved pixels from the alignment process.

Additionally, depth estimation models may fail to accurately resolve depth at object boundaries, often yielding smooth transitions where abrupt changes are expected. This issue affects the overall warping quality, resulting in artifacts such as trailing patterns within occluded regions. To address this, we apply bilateral filtering to sharpen the depth boundaries, enhancing inpainting performance. Additional implementation details are provided in the supplementary material.

4.3. Scene Completion with Temporal Consistency

Upon completing the warping and inpainting for the first timestamp, we proceed to apply these operations sequentially across subsequent timestamps. However, directly extending the same approach to each timestamp independently can lead to temporal inconsistencies. This is due to the inherent variability in 2D diffusion-based inpainting, which may produce differing results for the same regions across different timestamps. To address this, we impose temporal consistency by ensuring that background regions remain visually coherent across frames. Specifically, we require that overlapping regions across timestamps exhibit similar content, especially in the background areas.

Foreground and Background Separation. After the inpainting process, we use a segmentation model to separate the foreground and background regions within each frame. For regions that contain significant occlusions, especially large missing areas in the background, we incorporate content from previous timestamps to fill these areas. This approach maintains temporal continuity by sourcing background information from earlier frames. For holes near the foreground boundary, we determine the inpainting source

based on the background or foreground status of the corresponding region in prior timestamps. If a boundary region classified as background in the current frame aligns with a background area in previous timestamps, we inpaint it using information from the earlier frame. Conversely, if the region is identified as part of the foreground in prior frames, we apply the 2D diffusion model for inpainting. This selective inpainting strategy allows us to maintain coherence across timestamps while appropriately filling areas based on temporal foreground and background information.

4.4. Training and Optimization

After performing all warping and inpainting operations across views and timestamps, we establish a comprehensive camera network, where each camera contains video frames as captured from its respective viewpoint. Importantly, this multi-view setup is constructed without the need for model-specific training. Using this multi-view spatial information and temporal dynamics, we employ a 4D rendering approach to synthesize novel perspectives of the scene. For the rendering, we leverage the 4D Gaussians framework [51], which represents the 4D space using a deformable network. The renderer takes Gaussian parameters, along with the timestamp, and compute the timestamp-conditioned deformation of these parameters. This approach enables continuous modeling of deformation, facilitating smooth interpolation between timestamps during novel view synthesis. At test time, any desired viewpoint and timestamp can be selected to generate a novel view.

5. Experiments

5.1. Implementation Details

Our optimization framework comprises two stages: initially reconstructing a network of cameras, each associated with its respective timeframe view, followed by training a 4D renderer. Specifically, we construct a network of 25 cameras and utilize videos spanning 50 timestamps. All experiments were conducted on a single A100 GPU. The complete warping and inpainting process, which operates without any additional training, requires approximately two hours. Following this, the 4D renderer is trained in about one hour, resulting in a total of approximately 3 hours to complete the training and generate novel views along any desired trajectory. This duration is significantly shorter than the training time required by recent state-of-the-art methods: Dream-in-4D requires over four hours, while 4Dfy takes over 20 hours, despite producing only object-level 4D renderings. To initialise the scene and establish motion priors for 4D reconstruction, we utilize CogVideoX-5b [54]. For depth estimation, DepthCrafter [16] is employed, as it produces consistent depth estimates across video frames, enabling reliable warping. Perspective Fields [20] is used to estimate

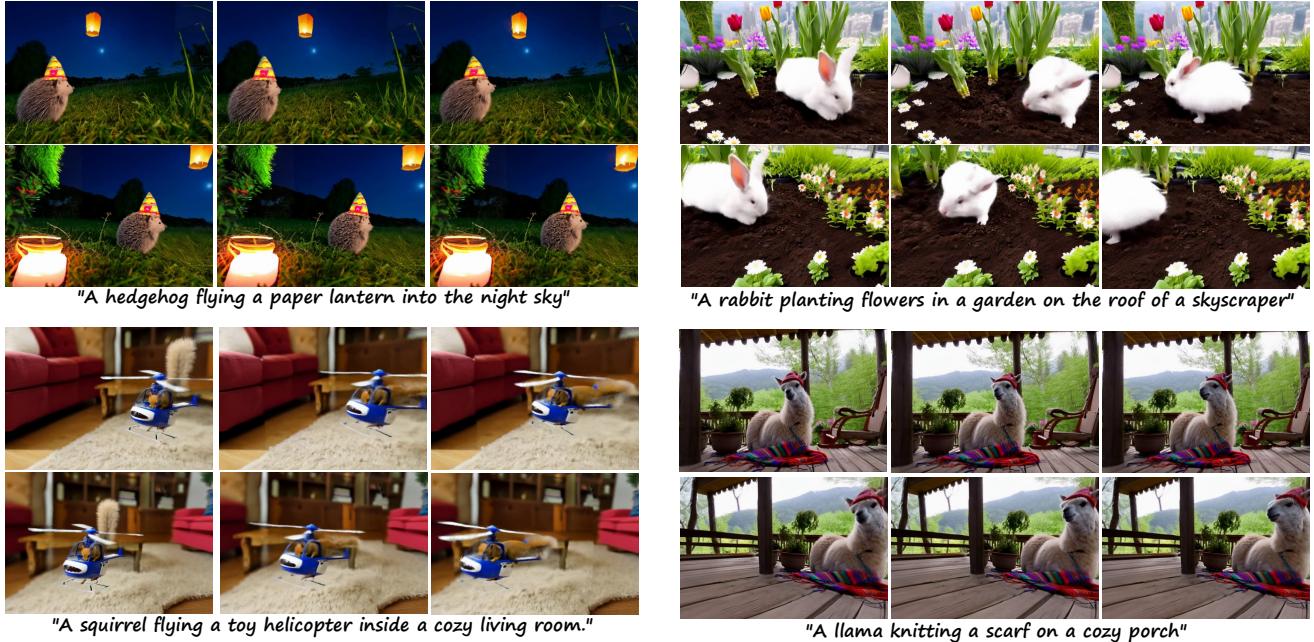


Figure 3. **Gallery of Results.** PaintScene4D successfully generates 4D scenes that maintain both view consistency and temporal coherence. Here, we present visualizations of each scene from 2 fixed viewpoints across varying timestamps. The horizontal axis represents the time progression, while the vertical axis represents different viewpoints. Due to space constraints, we display results for only four scenes. Additional results are provided in the supplementary materials for further reference.

the camera intrinsics for the generated video. For the segmentation model to distinguish foreground and background, we use GroundingSAM-2[37].

5.2. Baselines and Evaluation Metrics

In the absence of open-source implementations for text-to-4D scene-level generation, we benchmark our approach against state-of-the-art text-to-4D object-level generation methods, namely 4Dfy [2] and Dream-in-4D [62], across a varied set of 20 prompts. To assess the effectiveness of our proposed approach, we utilize the CLIP Score [35] alongside a structured user study.

CLIP Score. The CLIP Score [32] assesses alignment between an input prompt and generated visual content by calculating the cosine similarity between CLIP’s text and visual embeddings [35]. Scored between 0 and 100, a higher value indicates a closer match. We compute the CLIP Score for 4Dfy, Dream-in-4D, and our method by rendering videos from each prompt, evaluating each frame with CLIP ViT-B/32, and averaging scores across all frames and prompts for consistency.

User Study. A comprehensive user study was conducted through Google Forms, involving 30 evaluators per video pair. Each evaluator was presented with three anonymized videos, each capturing a dynamic scene from a camera moving along a circular trajectory. The videos were accompanied by the original text prompt. Evaluators were shown the renderings from 4Dfy, Dream-in-4D, and our approach

Table 1. **Quantitative results.** We compare our method against 4D-fy and Dream-in-4D. The methods are evaluated in terms of CLIP Score (CLIP) and human preference based on motion realism(MR), video-text alignment (VTA), High Dynamicity (HR), general realism (GR), and overall preference (Overall). The reported human preference is the percentage of users who voted for the respective method in a head-to-head comparisons.

Method	CLIP	Human Preference				
		MR	VTA	HR	GR	Overall
4D-fy [2]	31.82	2%	11%	5%	7%	7%
Dream-in-4D [62]	28.14	13%	14%	17%	2%	11%
Ours	36.02	85%	75%	78%	91%	82%

simultaneously, and were asked to rate their preferences based on 4 criteria: motion realism, video-text alignment, high dynamicity, general realism. Detailed results are presented in Table 1.

5.3. Text-to-4D Generation

In Figure 4, we visualize spatio-temporal renderings produced by our method compared to 4D-fy and Dream-in-4D. Although all approaches are capable of synthesizing 4D scenes, 4D-fy and Dream-in-4D focus on object-level renderings and lack fine spatial details. Our approach, by contrast, generates scene-level 4D reconstructions in a significantly reduced time, producing realistic renderings. Notably, 4D-fy struggles to model realistic motion, while Dream-in-4D produces cartoonish effects that diminish re-

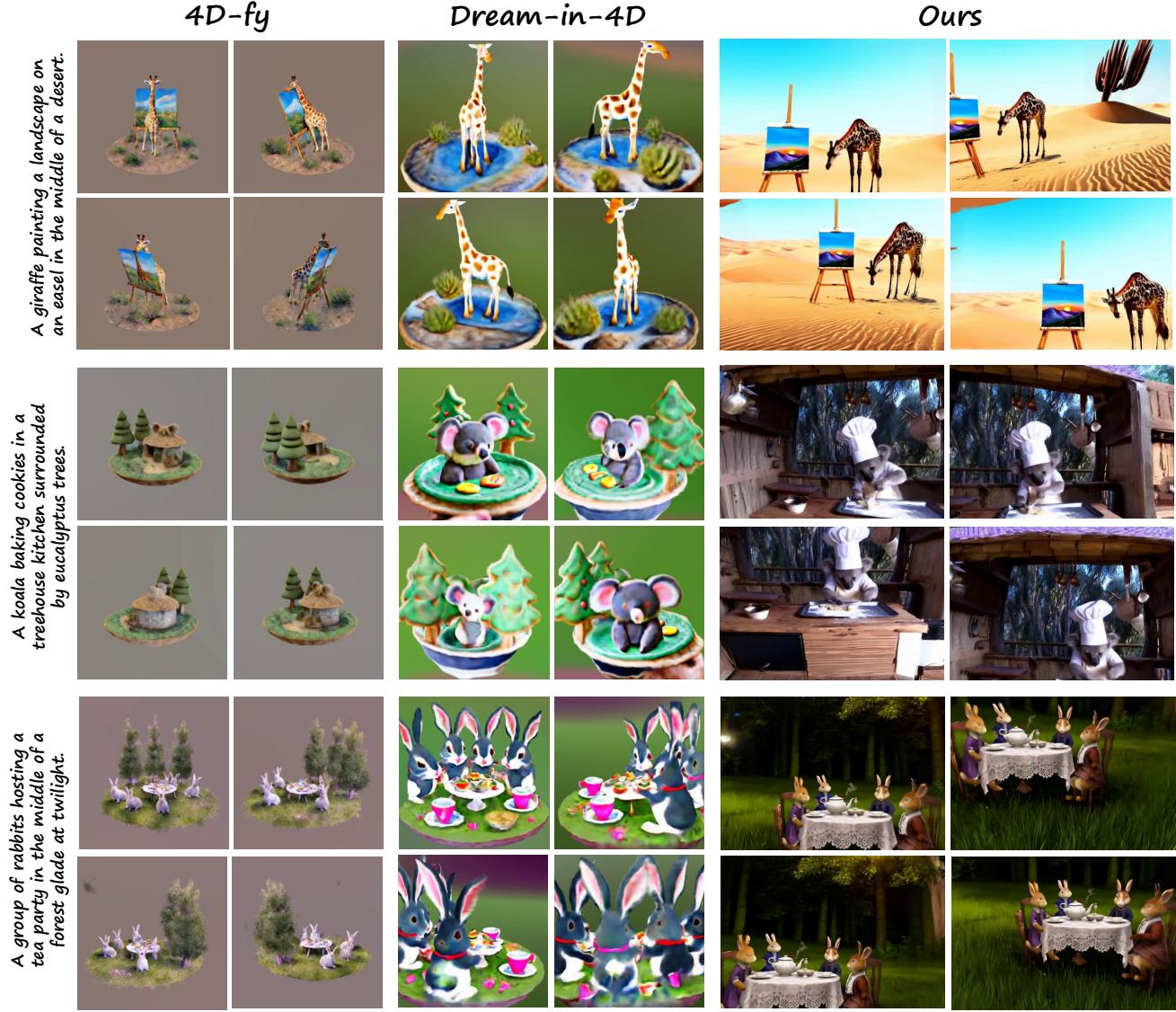


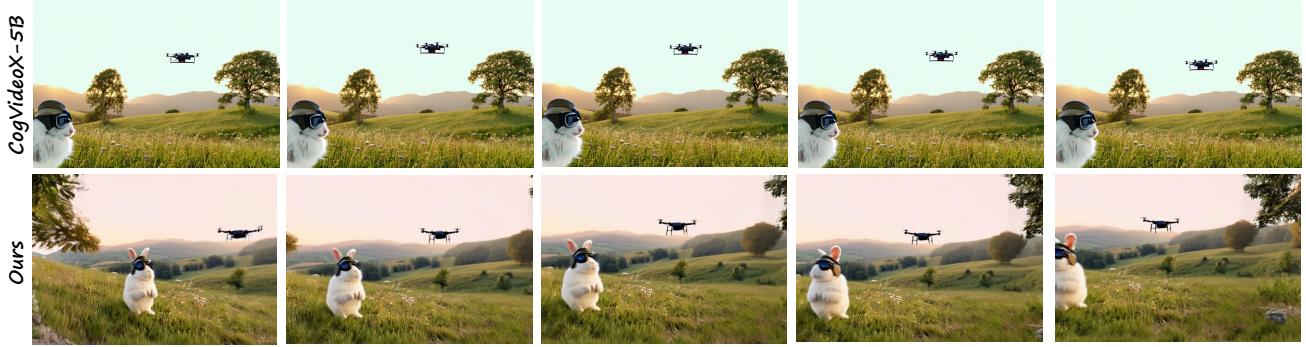
Figure 4. **Text-to-4D Comparisons.** We compare our approach with state-of-the-art text-to-4D generation methods. While both baseline methods produce scenes that broadly align with the text prompts, they lack essential fine details. Specifically, 4D-fy shows minimal motion and limited detail, whereas Dream-in-4D captures dynamics more effectively but produces stylized, cartoon-like renderings, which are less suitable for realistic applications. In contrast, our method synthesizes photorealistic 4D scenes that faithfully follow the input text prompt while presenting significant, realistic dynamics within the scene.

alism. In contrast, our method achieves high photorealistic quality across both spatial and temporal dimensions. We also present a gallery of our results in Figure 3. Quantitative metrics indicating both CLIP score and User study are reported in Table 1. Our method outperforms both 4D-fy and Dream-in-4D in terms of CLIP Score and user study preferences. Evaluators showed a statistically significant preference for PaintScene4D due to its higher motion realism, photorealistic rendering of both foreground and background, overall realism, and better video-text alignment.

5.4. Explicit Camera Control

To assess camera control, we compared our framework with other text-to-video (T2V) models as illustrated in Figure 5.

We input the same text prompts into the T2V model twice, adjusting only the camera movement description to direct it to “tilt towards the right” in one case and “move upwards” in the other. This setup allows us to examine the models’ ability to interpret and execute nuanced camera control instructions. Our observations reveal two key limitations of T2V models. First, even with a fixed seed, the T2V model generates different scenes for each altered prompt. Secondly, although the model simulates an upward camera movement in the second case, it lacks explicit control over the degree of camera motion. In contrast, our approach enables explicit, consistent control over camera trajectory within the same scene and motion dynamics, leveraging 4D modeling for precise camera manipulation.



"A rabbit flying a drone in the middle of mountains. The camera tilt towards the right"



"A rabbit flying a drone in the middle of mountains. The camera moves upwards"

Figure 5. **Camera Control.** PaintScene4D demonstrates strong explicit camera control capabilities. Input text prompt: “A rabbit flying a drone in the middle of mountains.” In contrast, to guide the T2V model, we append camera motion directives such as “The camera tilts to the right” or “The camera moves upwards” to the text prompt. However, due to implicit handling of camera motion, T2V often fails to generate precise or controllable camera movements. Our approach, once trained, allows for flexible camera trajectories within the bounds of the input training cameras, achieving precise and repeatable control over camera movements for the same scene.



Figure 6. **Ablation Study.** We demonstrate that each of our proposed elements is essential for mitigating artifacts and inconsistencies, resulting in smooth consistent renderings.

6. Ablation Study

We present an ablation study to analyze the components of our PaintScene4D framework, with the results shown in Figure 6.

Depth Alignment Module: The inclusion of the depth alignment module is crucial for maintaining the geometric consistency of the foreground. During the warping process, all frames are utilized, and any depth inconsistencies across frames result in error accumulation, leading to noticeable artifacts, particularly at the foreground boundaries.

Farthest View Sampling: In PaintScene4D, we select the farthest view at each step of the warping process to max-

imize the area inpainted. Omitting this step causes severe degradation near the edges of the foreground, such as the panda’s boundary, where needle-like artifacts emerge due to the Gaussian splatting process.

Consistent Inpainting Module: Temporal consistency in inpainting is essential for coherent 4D scene generation. Without this module, inpainting becomes inconsistent at the boundaries of objects (e.g., the panda) across different timestamps, leading to significantly degraded renderings.

7. Conclusion

We introduce PaintScene4D, a novel training-free framework for generating photorealistic 4D scenes from a single text prompt. Our method addresses the challenges of spatial and temporal inconsistencies and enables the generation of novel views along a user-defined camera trajectory. PaintScene4D outperforms existing baselines in terms of visual quality, 3D consistency, and motion accuracy while also providing the capability for explicit camera control during inference. Notably, our approach is the first to implement text-to-4D scene generation using open-source models, offering a valuable contribution to the field.

References

- [1] Giovanni Adorni and Mauro Di Manzo. Natural language input for scene generation. In *EACL*, 1983. 2
- [2] Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B Lindell. 4D-fy: Text-to-4D generation using hybrid score distillation sampling. In *CVPR*, 2024. 2, 3, 6, 1
- [3] Sherwin Bahmani, Xian Liu, Wang Yifan, Ivan Skorokhodov, Victor Rong, Ziwei Liu, Xihui Liu, Jeong Joon Park, Sergey Tulyakov, Gordon Wetzstein, Andrea Tagliasacchi, and David B Lindell. TC4D: Trajectory-conditioned text-to-4D generation. In *ECCV*, 2025. 3
- [4] Ang Cao and Justin Johnson. HexPlane: A fast representation for dynamic scenes. In *CVPR*, 2023. 3
- [5] Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3D-aware diffusion models. In *ICCV*, 2023. 2
- [6] Angel Chang, Manolis Savva, and Christopher D Manning. Learning spatial knowledge for text to 3D scene generation. In *EMNLP*, 2014. 2
- [7] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2Shape: Generating shapes from natural language by learning joint embeddings. In *ACCV*, 2018. 2
- [8] Bob Coyne and Richard Sproat. WordsEye: An automatic text-to-scene conversion system. In *SIGGRAPH*, 2001. 2
- [9] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. In *CVPR*, 2023. 2
- [10] Christoph Fehn. Depth-image-based rendering, compression, and transmission for a new approach on 3D-TV. *SPIE Stereoscopic Displays and Virtual Reality Systems XI*, 2004. 4
- [11] William Gao, Noam Aigerman, Thibault Groueix, Vova Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *SIGGRAPH*, 2023. 2
- [12] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. NeRFDiff: Single-image view synthesis with NeRF-guided distillation from 3D-aware diffusion. In *ICML*, 2023. 2
- [13] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 2
- [14] Lukas Höllerin, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2Room: Extracting textured 3D meshes from 2D text-to-image models. In *ICCV*, 2023. 3
- [15] 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. 1
- [16] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. DepthCrafter: Generating consistent long depth sequences for open-world videos. *arXiv preprint arXiv:2409.02095*, 2024. 5, 1
- [17] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, 2022. 2
- [18] Nikolay Jetchev. Clipmatrix: Text-controlled creation of 3D textured meshes. *arXiv preprint arXiv:2109.12922*, 2021. 2
- [19] Yanqin Jiang, Li Zhang, Jin Gao, Weimin Hu, and Yao Yao. Consistent4D: Consistent 360 degree dynamic object generation from monocular video. *arXiv preprint arXiv:2311.02848*, 2023. 2
- [20] Linyi Jin, Jianming Zhang, Yannick Hold-Geoffroy, Oliver Wang, Kevin Matzen, Matthew Stich, and David F. Fouhey. Perspective fields for single image camera calibration. In *CVPR*, 2023. 4, 5
- [21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 3
- [22] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3D content creation. In *CVPR*, 2023. 2
- [23] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3D content creation. In *CVPR*, 2023. 1
- [24] Yukang Lin, Haonan Han, Chaoqun Gong, Zunnan Xu, Yachao Zhang, and Xiu Li. Consistent123: One image to highly consistent 3D asset using case-aware diffusion priors. *arXiv preprint arXiv:2309.17261*, 2023. 2
- [25] Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4D with dynamic 4D gaussians and composed diffusion models. *arXiv preprint arXiv:2312.13763*, 2023. 2
- [26] Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4D with dynamic 3D gaussians and composed diffusion models. In *CVPR*, 2024. 3
- [27] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite Nature: Perpetual view generation of natural scenes from a single image. In *ICCV*, 2021. 5
- [28] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *ICCV*, 2023. 1, 2
- [29] Y Liu, C Lin, Z Zeng, X Long, L Liu, T Komura, and W Wang. SyncDreamer: Learning to generate multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 1, 2
- [30] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. VideoFusion: Decomposed diffusion models for high-quality video generation. *arXiv preprint arXiv:2303.08320*, 2023. 3

- [31] Gal Metzger, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-NeRF for shape-guided generation of 3D shapes and textures. In *CVPR*, 2023. 1
- [32] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In *NeurIPS*, 2021. 6
- [33] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *ICLR*, 2023. 1, 2, 3
- [34] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. Magic123: One image to high-quality 3D object generation using both 2D and 3D diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. 1, 2
- [35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 6, 1
- [36] Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. DreamGaussian4D: Generative 4D Gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023. 2
- [37] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kun-chang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded SAM: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 6
- [38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2
- [39] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar 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. *NeurIPS*, 2022. 2
- [40] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malek-shan. CLIP-Forge: Towards zero-shot text-to-shape generation. In *CVPR*, 2022. 2
- [41] 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. 2
- [42] 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*, 2023. 1, 2, 3
- [43] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2
- [44] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, and Yaniv Taigman. Text-to-4D dynamic scene generation. In *ICML*, 2023. 2, 3
- [45] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3D: High-fidelity 3D creation from a single image with diffusion prior. *arXiv preprint arXiv:2303.14184*, 2023. 2
- [46] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 2004. 4
- [47] Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Reznikov, Joshua B Tenenbaum, Frédéric Durand, William T Freeman, and Vincent Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *arXiv preprint arXiv:2306.11719*, 2023. 2
- [48] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. CLIP-NeRF: Text-and-image driven manipulation of neural radiance fields. In *CVPR*, 2022. 2
- [49] Jiniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 2
- [50] 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. *NeurIPS*, 2023. 2
- [51] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4D Gaussian splatting for real-time dynamic scene rendering. In *CVPR*, 2024. 5, 1
- [52] Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. SV4D: Dynamic 3D content generation with multi-frame and multi-view consistency. *arXiv preprint arXiv:2407.17470*, 2024. 2
- [53] Dejia Xu, Hanwen Liang, Neel P Bhatt, Hezhen Hu, Hanxue Liang, Konstantinos N Plataniotis, and Zhangyang Wang. Comp4D: LLM-guided compositional 4D scene generation. *arXiv preprint arXiv:2403.16993*, 2024. 3
- [54] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenqi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2, 5, 1
- [55] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics*, 2019. 3
- [56] Yuyang Yin, Dejia Xu, Zhangyang Wang, Yao Zhao, and Yunchao Wei. 4DG: Grounded 4D content generation with spatial-temporal consistency. *arXiv preprint arXiv:2312.17225*, 2023. 2
- [57] Paul Yoo, Jiaxian Guo, Yutaka Matsuo, and Shixiang Shane Gu. DreamSparse: Escaping from plato’s cave with 2D diffusion model given sparse views. *arXiv preprint arXiv:2306.03414*, 2023. 2

- [58] Heng Yu, Chaoyang Wang, Peiyue Zhuang, Willi Menapace, Aliaksandr Siarohin, Junli Cao, Laszlo A Jeni, Sergey Tulyakov, and Hsin-Ying Lee. 4Real: Towards photorealistic 4D scene generation via video diffusion models. *arXiv preprint arXiv:2406.07472*, 2024. [3](#)
- [59] Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. Text2NeRF: Text-driven 3D scene generation with neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 2024. [3](#)
- [60] Songchun Zhang, Yibo Zhang, Quan Zheng, Rui Ma, Wei Hua, Hujun Bao, Weiwei Xu, and Changqing Zou. 3D-SceneDreamer: Text-driven 3D-consistent scene generation. In *CVPR*, 2024. [3](#)
- [61] Yuyang Zhao, Zhiwen Yan, Enze Xie, Lanqing Hong, Zhen-guo Li, and Gim Hee Lee. Animate124: Animating one image to 4D dynamic scene. *arXiv preprint arXiv:2311.14603*, 2023. [2](#)
- [62] Yufeng Zheng, Xuetong 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. [2](#), [6](#), [1](#)
- [63] Yufeng Zheng, Xuetong Li, Koki Nagano, Sifei Liu, Otmar Hilliges, and Shalini De Mello. A unified approach for text-and image-guided 4D scene generation. In *CVPR*, 2024. [3](#)
- [64] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *SIGGRAPH*, 2001. [3](#)

PaintScene4D: Consistent 4D Scene Generation from Text Prompts

Supplementary Material

A. Demo Video

We have provided a project webpage at <https://paintscene4d.github.io/>, including a video demonstration. The demo video presented in the website highlights the versatility and robustness of our framework by showcasing a diverse gallery of results generated from various text prompts and scene configurations. Furthermore, to substantiate the effectiveness of our approach, the demo video includes detailed comparisons with baseline methods such as 4D-fy [2] and Dream-in-4D [62]. Although both 4D-fy and Dream-in-4D are designed for object-level generation, they require significantly more computational time than our approach. Furthermore, these methods are limited to generating object-level renderings, whereas our framework is capable of rendering **complete scene-level generations**. These comparisons visually demonstrate the superior fidelity, consistency, and dynamicity achieved by our method across a wide range of scenarios.

In the demo video, we also showcase **explicit control over camera movements** in a rendered scene, offering a significant advantage over Text-to-Video (T2V) models. T2V models *lack direct camera control*, relying instead on implicit instructions through text prompts, which often yield inconsistent and less effective results. Additionally, T2V models generate a different scene for each iteration, even with identical prompts, limiting reproducibility. In contrast, our method supports precise manipulation of the camera trajectory within the same scene, ensuring consistency and offering greater flexibility for tailored visual outputs.

B. More Qualitative Results

In Figure B and Figure C, we provide more examples generated using our proposed framework to demonstrate the robustness of our methodology. The horizontal axis represents the time axis, and the vertical axis represents different view points. To fully appreciate the quality and diversity of our text-to-4D generation results, we strongly recommend viewing the accompanying video.

C. PaintScene4D: Implementation Details

C.1. Hyperparameters

Warping and Inpainting Module. Table A demonstrates the parameters used in the warping and inpainting module. The values are carefully selected to balance efficiency and quality. The *Number of Cameras* determines the multi-view

Parameters	Value
Number of Cameras	25
Relative Depth Estimator	DepthCrafter [16]
Absolute Depth Estimator	Metric3D v2 [15]
Inference Steps for Inpainting	50
Inpainting Iterations	10
Filter Size for Bilateral Filtering	[3, 5]

Table A. Hyperparameters for the warping and inpainting module.

Parameters	Value
Batch Size	4
Number of Iterations (<i>Coarse Training</i>)	3000
Number of Iterations (<i>Fine Training</i>)	15000
Densification Until Iteration	10000

Table B. Hyperparameters for 4D gaussian splatting training process.

Parameters	Value
Resolution	720×480
Number of Timestamps	49
Number of Inference Steps	50
Guidance Scale	6

Table C. Hyperparameters for generating video from text-to-video model CogVideoX [54].

coverage necessary for generating high-quality reconstructions. The *Relative Depth Estimator* and *Absolute Depth Estimator* are key for warping operation, with DepthCrafter used for relative depth estimation and Metric3D v2 for absolute scaling. We inpaint the missing regions multiple times and pick the best one using a CLIP [35] based selector. The *Inpainting Iterations* represents the number of times we inpaint the missing region. The *Filter Size for Bilateral Filtering* sharpens the edges of a depth map necessitating better inpainting quality.

4D Gaussian Splatting Module. The hyperparameters for training the 4D gaussian splatting [51] framework are presented in Table B. The *Number of Iterations (Coarse Training: 3000, Fine Training: 15000)* ensures robust initialization and detailed refinement. *Densification Until Iteration* specifies when Gaussian points should be densely packed to model finer scene details.

Text-to-Video Generation. Table C presents hyperparameters for video generation with the text-to-video model

CogVideoX [54]. The hyperparameters are chosen to optimize the synthesis quality and temporal consistency. The *Number of Timestamps* defines the temporal resolution of the scene. The *Number of Inference Steps* impacts generation fidelity.



Figure A. Failure Case: Our method is dependent on the assumption that the initial video generation exhibits no large camera movement. Large camera motion in the video, introduces distortions and artifacts during the subsequent rendering process, significantly affecting the visual fidelity of the final output.

D. Limitations and Future Work

While our method successfully generates photorealistic 4D scenes from a single text prompt, several limitations persist. First, our approach assumes that the input video is captured from a nearly static, non-moving camera. This assumption does not always hold when using text-to-video (T2V) models, which typically offer limited control over camera dynamics. Videos with large camera movement results in a degraded visual quality. We demonstrate the failure case in Figure A. Therefore, extending our framework to accommodate videos with more camera movements represents a promising direction for future work. Second, our current method does not explicitly model the 3D structure of the foreground. Instead, we rely on an inpainting model to fill in gaps at the boundaries of the foreground, which means that the model does not possess a comprehensive understanding of the 3D geometry of the scene. A more advanced approach could involve explicitly separating the foreground from the background and modeling the 3D structure of the foreground, potentially using methods like SV4D [52]. Finally, our approach struggles to handle rapid movements in the video due to the limitations of current 4D rendering techniques. Advancements in this area would likely enhance the rendering quality of our method and enable better handling of fast motion.



"A fox building a sandcastle on a beach in the evening"



"A bear flying a kite shaped like a dragon on a hilltop."



"A dog exploring a mystical forest"



"A dragon roasting marshmallows in a cozy cave with a view of castle."



"A fox building a sandcastle on a tropical beach"



"A fox exploring an abandoned lighthouse on a rocky island"



"A giraffe decorating a giant Christmas tree"



"A kangaroo running a farm stand selling fresh produce"

Figure B. Gallery of Results: We present qualitative results of our text-to-4D generation framework, showcasing superior visual fidelity, consistent multi-view reconstructions, plausible scene compositions, and realistic dynamic motions. The horizontal axis represents the time axis and the vertical axis represents different view points. A comprehensive collection of video demonstrations is provided in the supplementary materials.



"A koala playing the drums on stage at an outdoor music festival."



"A lion playing the drums in a rock band during a concert."



"A monkey solving a Rubik's cube while sitting in a treehouse"



"An owl reading a scroll by candlelight in an ancient, dusty library."



"A panda scientist mixes vibrant chemicals in a lab"



"A peacock reading a book on a park bench in spring season"



"A group of raccoons playing video games in a living room"



"A squirrel mixing potions in a wizard's tower"

Figure C. Gallery of More Results. A comprehensive collection of video demonstrations is provided in the supplementary materials.