DreamScene: 3D Gaussian-based Text-to-3D Scene Generation via Formation Pattern Sampling

Haoran Li[1,2], Haolin Shi[1,2], Wenli Zhang[1,2], Wenjun Wu[1,2], Yong Liao[1,2]*, Lin Wang[3], Lik-Hang Lee[4], and Peng Yuan Zhou[5]*

[1] University of Science and Technology of China
[2] CCCD Key Lab of Ministry of Culture and Tourism
{lhr123, mar, zwl384369, wu_wen_jun}@mail.ustc.edu.cn, yliao@ustc.edu.cn
[3] AI Thrust, HKUST(GZ) and Dept. of Computer Science Eng., HKUST
linwang@ust.hk
[4] The Hong Kong Polytechnic University
lik-hang.lee@polyu.edu.hk
[5] Aarhus University
pengyuan.zhou@ece.au.dk

Abstract. Text-to-3D scene generation holds immense potential for the gaming, film, and architecture sectors. Despite significant progress, existing methods struggle with maintaining high quality, consistency, and editing flexibility. In this paper, we propose **DreamScene**, a 3D Gaussian-based novel text-to-3D scene generation framework, to tackle the aforementioned three challenges mainly via two strategies. First, DreamScene employs Formation Pattern Sampling (FPS), a multi-timestep sampling strategy guided by the formation patterns of 3D objects, to form fast, semantically rich, and high-quality representations. FPS uses 3D Gaussian filtering for optimization stability, and leverages reconstruction techniques to generate plausible textures. Second, DreamScene employs a progressive three-stage camera sampling strategy, specifically designed for both indoor and outdoor settings, to effectively ensure object and environment integration and scene-wide 3D consistency. Last, DreamScene enhances scene editing flexibility by integrating objects and environments, enabling targeted adjustments. Extensive experiments validate DreamScene's superiority over current state-of-the-art techniques, heralding its wide-ranging potential for diverse applications. Code and demos are released at `https://dreamscene-project.github.io`.

Keywords: Text-to-3D · Text-to-3D Scene · 3D Gaussian · Scene Generation · Scene Editing

1 Introduction

The advancement of text-based 3D scene generation [4, 10, 12, 19, 26, 28, 41, 42] marks a notable evolution in 3D content creation [3, 11, 13, 16–18, 20, 21, 25, 29, 36,

* Corresponding authors

40], broadening its scope from crafting simple objects to constructing detailed, complex scenes directly from text. This leap forward eases 3D modelers' workload and fuels growth in the gaming, film, and architecture sectors.

Text-to-3D methods [3, 11, 13, 16–18, 20, 21, 25, 29, 36, 40] typically employ pre-trained 2D text-to-image models [30–32] as prior supervision to generate object-centered 3D differentiable representations [14, 22, 24, 27, 33]. While text-to-3D scene generation methods demand rendering from predefined camera positions outward, capturing the scene from these perspectives. These text-to-3D generation methods, as depicted in Fig. 1, encounter several critical challenges, including: **1)** The inefficient generation process that often leads to low-quality outputs [4,19,28,41] and long completion time [10,38]. **2)** Inconsistent 3D visual cues [4, 10, 26, 28, 38, 41, 42], with acceptable results limited to specific camera poses, akin to 360-degree photography. **3)** Unable to separate objects from the environments, hindering flexible edition on individual elements [10, 26, 38, 41].

In this paper, we introduce ***DreamScene***, a pioneering 3D Gaussian-based text-to-3D scene generation framework that primarily utilizes the innovative Formation Pattern Sampling (FPS) method. Accompanied by a strategic camera sampling and the seamless object-environment integration, DreamScene efficiently tackles the challenges above, paving the way for crafting high-quality and consistent scenes. Specifically, based on the observed patterns in 3D representation formation, FPS employs multi-timestep sampling (MTS) to swiftly balance the semantic information and shape consistency in order to generate high-quality and semantically rich 3D representations. FPS ensures stable generation performance by eliminating redundant internal 3D Gaussians during optimization. Finally, employing DDPM [8] with small timestep sampling and 3D reconstruction techniques [14], FPS efficiently generates surfaces with plausible textures from various viewpoints in only ***tens of seconds***.

Next, we propose an incremental three-stage camera sampling strategy to ensure 3D consistency. First, we generate a coarse environment representation via camera sampling at the scene center. Second, we adapt ground formation to the scene type: a) indoors, by dividing into regions and randomly selecting a camera position for rendering; b) outdoors, by organizing into concentric circles based on the radius, sampling camera poses at varying circles along the same direction. Finally, we consolidate the scene through reconstructive generation in FPS, utilizing all camera poses to further refine the scene.

We integrate optimized objects into the scene based on specific layouts to enhance scene generation, thereby preventing the production of duplicate or physically unrealistic artifacts, such as the multi-headed phenomena (generating the same content in all directions) in text-to-3D. Following scene generation, flexible scene editing can be achieved by individually adjusting objects and environments (e.g., modifying positions and changing styles).

Our main contributions can be summarized as follows:

– We introduce DreamScene, a novel framework for text-driven 3D scene generation. Leveraging Formation Pattern Sampling, a strategic camera sampling

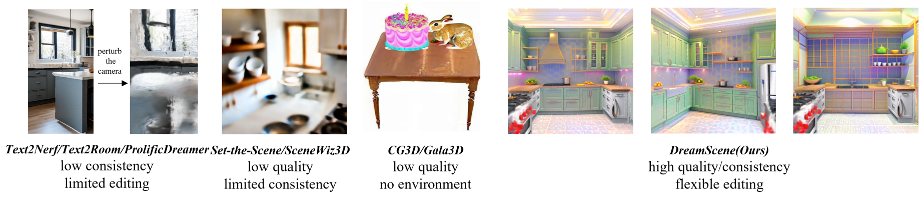


Fig. 1: DreamScene compared with current SOTA text-to-3D scene generation methods. Text2NeRF [41], Text2Room [10], and ProlificDreamer [38] require 7 $\sim$ 12 hours to generate the scene while DreamScene only needs ***1 hour***. Moreover, DreamScene is capable of generating scenes that accommodate up to 20 objects as shown in later figures.

approach, and seamless object-environment integration, DreamScene efficiently produces high-quality, scene-wide consistent and editable 3D scenes.
- Formation Pattern Sampling, central to our approach, harnesses multi-timestep sampling, 3D Gaussian filtering, and reconstructive generation, delivering high-quality, semantically rich 3D representations in 30 minutes.
- Qualitative and quantitative experiments prove that DreamScene outperforms existing methods in text-driven 3D object and scene generation, unveiling substantial potential for numerous fields such as gaming and film.

2 Related Work

2.1 Differentiable 3D Representation

Differentiable methods such as NeRF [1,22], SDF [27,33], and 3D Gaussian Splatting [14] enable the representation, manipulation, and rendering of 3D objects and scenes. These representations are compatible with optimization algorithms like gradient descent, facilitating the automatic tuning of 3D representation parameters to reduce loss. Particularly, the recent approach [14] of modeling 3D scenes with differentiable 3D Gaussians has achieved superior real-time rendering via splatting. Unlike implicit representations [1,22,24], 3D Gaussians provide a more explicit framework, simplifying the integration of multiple scenes. Hence, we adopt 3D Gaussians for their clarity of explicit representation and ease of scene combination.

2.2 Text-to-3D Generation

Current text-to-3D tasks mainly generate 3D representation directly [13, 25, 34] or distilled from large 2D text-to-image models [3,18,29]. Direct methods require annotated 3D datasets for rapid generation, often suffering from lower quality and high GPU demands, thus commonly serving as initial steps for distillation techniques [17, 39]. For instance, Point-E [25] generates an image via a diffusion model based on text, which is then transformed into the point cloud. Shap-E [13] maps 3D assets to implicit function parameters with an encoder, then trains a conditional diffusion model on these parameters.

Distilling 3D representation from 2D text-to-image models [3,11,18,21,29,40] has become the predominant strategy. DreamFusion [29], a trailblazer in this domain, introduced Score Distillation Sampling (SDS) to ensure that images rendered from multiple perspectives conform to the distribution of 2D text-to-image models [30–32]. Following its lead, subsequent advancements [3, 11, 13, 16–18, 20, 21, 25, 36, 40] refined 3D generation in terms of quality, speed, and diversity. For example, DreamTime [11] accelerates the convergence of generation by monotonically non-increasing sampling of timesteps t in 2D text-to-image model, while LucidDreamer [17] utilizes DDIM inversion [7, 23] to ensure 3D consistency of the object generation process. Inspired by these pioneering works, our method introduces a more efficient approach for generating high-quality and semantically rich 3D representation.

2.3 Text-to-3D Scene Generation Methods

Current text-to-3D scene generation methods, as illustrated in Fig. 1 face significant limitations. [10, 26, 41] relying on inpainted images for scene completion can generate realistic visuals but suffer from limited 3D consistency. Even minor disturbances can lead to the collapse of the scene's visual integrity, indicating a lack of robustness. Moreover, such methods typically do not allow for the editing of objects within the scene and have unrealistic scene compositions, such as a living room cluttered with an excessive number of sofas. While some methods [4, 42], akin to ours, attempt to merge objects with environments to ensure consistency, they generally yield scenes of lower quality and can accommodate only a limited number of objects. Approaches [19, 37, 44] focusing solely on object integration fall short of generating comprehensive scenes. The constraints on quality, consistency, and editability have thus far hindered the broader application of text-to-3D technologies. Our method represents a leap forward by ensuring high-quality, scene-wide consistency and enabling flexible editing.

3 Preliminary

Diffusion Models [8, 35] guide data x ($x \sim p(x)$) generation by estimating the gradients of log probability density functions $\nabla_x \log p_{data}(x)$. The training process involves adding noise to the input x over t steps,

$$x_t = \sqrt{\bar{\alpha}_t} x + \sqrt{1 - \bar{\alpha}_t} \epsilon, \tag{1}$$

where $\bar{\alpha}_t$ is the predefined coefficient and $\epsilon \sim \mathcal{N}(0, I)$ is the noise. Then fitting the noise prediction network ϕ by minimizing the prediction loss $\mathcal{L}_t$:

$$\mathcal{L}_t = \mathbb{E}_{x, \epsilon \sim \mathcal{N}(0,I)} \left[\|\epsilon_\phi(x_t, t) - \epsilon\|^2 \right], \tag{2}$$

During the sampling phase, it can deduce x from the noisy input and its noise estimation $\epsilon_\phi(z_t, t)$.

Score Distillation Sampling (SDS) was proposed by DreamFusion [29] to address the issue of distilling 3D representations from a 2D text-to-image diffusion model. Considering a differentiable 3D representation parameterized by θ and a rendering function denoted as g, the rendered image produced for a given camera pose c can be expressed as $x = g(\theta, c)$. Then SDS distills θ through a 2D diffusion model ϕ with frozen parameters as follows:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\theta) = \mathbb{E}_{t,\epsilon,c}\left[w(t)(\epsilon_\phi(x_t; y, t) - \epsilon)\frac{\partial g(\theta, c)}{\partial \theta} \right], \tag{3}$$

where $w(t)$ represents a weighting function that varies according to the timesteps t and y denotes the text embedding derived from the provided prompt.

Classifier Score Distillation [40] is a variant of SDS inspired by Classifier-Free Guidance (CFG) [9], divides the noise difference in SDS into generation prior $\epsilon_\phi(x_t; y, t) - \epsilon$ and classifier score $\epsilon_\phi(x_t; y, t) - \epsilon_\phi(x_t; \emptyset, t)$. It posits that the classifier score is sufficiently effective for text-to-3D, expressed as follows:

$$\nabla_\theta \mathcal{L}_{\text{CSD}}(\theta) = \mathbb{E}_{t,\epsilon,c}\left[w(t)(\epsilon_\phi(x_t; y, t) - \epsilon_\phi(x_t; \emptyset, t))\frac{\partial g(\theta, c)}{\partial \theta} \right]. \tag{4}$$

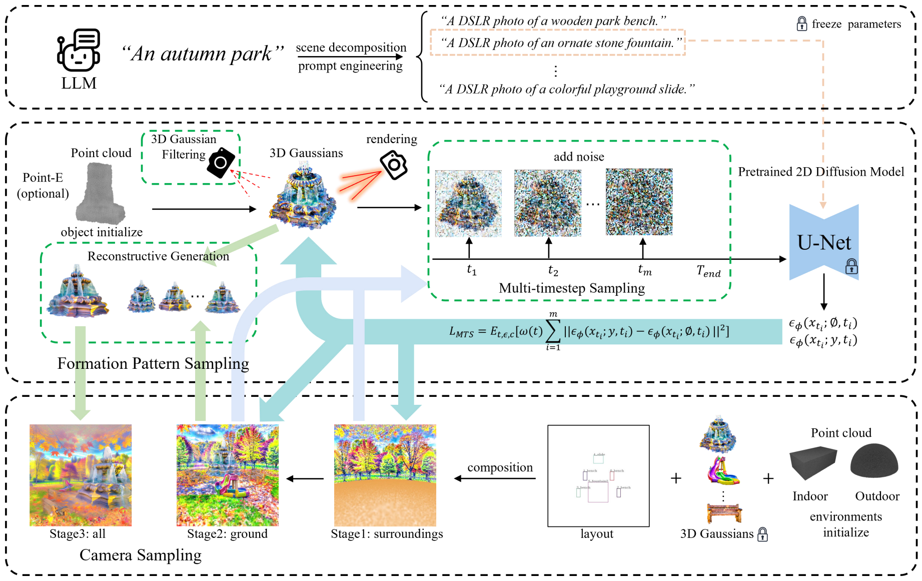
3D Gaussian Splatting [2, 14] is a cutting-edge technique in the field of 3D reconstruction. A 3D Gaussian is characterized by a full 3D covariance matrix Σ defined in world space centered at point (mean) μ:

$$G(\mathbf{x}) = e^{-\frac{1}{2}\mathbf{x}^T \Sigma^{-1} \mathbf{x}}, \tag{5}$$

spherical harmonics (SH) coefficients, and opacity α. By employing interlaced optimization and density control for these 3D Gaussians, particularly optimizing the anisotropic covariance, one can achieve precise scene representation. The tile-based rendering strategy enables efficient anisotropic splatting, thereby accelerating the training process and achieving real-time rendering capability.

4 Method

DreamScene divides the scene into objects and environments. It first generates the objects and then places them within the environments to create the entire scene. This approach is designed to prevent the generation of objects in the environments and to mitigate the multi-head problem in 3D generation (outdoor environments can be generated separately). Specifically, we employ prompt engineering to develop a large language model (LLM) [43] agent that transforms scene text into detailed object descriptions $y_i (i = 1, .., N)$ and environment description y_e, and rapidly create high-quality objects using Formation Pattern Sampling, which consists of multi-timestep sampling, 3D Gaussian filtering and



LLM
"An autumn park"
scene decomposition
prompt engineering
"A DSLR photo of a wooden park bench."
"A DSLR photo of an ornate stone fountain."
"A DSLR photo of a colorful playground slide."
freeze parameters
Point-E (optional)
Point cloud
object initialize
3D Gaussian Filtering
3D Gaussians
rendering
add noise
Pretrained 2D Diffusion Model
U-Net
t_1
t_2
t_m
T_{end}
Multi-timestep Sampling
Reconstructive Generation
Formation Pattern Sampling
$L_{MTS} = E_{t,\epsilon,c}[\omega(t)\sum_{i=1}^{m}\|\epsilon_\phi(x_{t_i};y,t_i) - \epsilon_\phi(x_{t_i};\emptyset,t_i)\|^2]$
$\epsilon_\phi(x_{t_i};\emptyset,t_i)$
$\epsilon_\phi(x_{t_i};y,t_i)$
Stage3: all
Stage2: ground
Stage1: surroundings
composition
layout
3D Gaussians
Point cloud
Indoor
Outdoor
environments initialize
Camera Sampling

Fig. 2: The overview of DreamScene. We primarily employ Formation Pattern Sampling, which includes multi-timestep sampling, 3D Gaussian filtering, and reconstructive generation to rapidly produce high-quality and semantically rich 3D representations with plausible textures and low storage demands. Additionally, DreamScene ensures scene-wide consistency through camera sampling and allows for flexible editing by integrating objects with the environments in the scene.

reconstructive generation. Then, we initialize cuboid 3D Gaussians to simulate walls, floors, and ceilings for bounded indoor scenes and hemispherical 3D Gaussians to simulate the ground and faraway surroundings for outdoor unbounded scenes. Through scene layout diagrams, we designate the positions of objects within the scene and use affine transformations (scale s, translation t, rotation r) to place objects in the scene coordinate system.

$$world(x) = r \cdot s \cdot obj_i(x) + t, i = 1, ..., N, \tag{6}$$

where x denotes the collection of coordinates for 3D Gaussians. Finally, we employ a camera sampling strategy for multi-stage optimization of the environments, achieving scene generation with high 3D consistency.

4.1 Formation Pattern Sampling

We have refined and broadened the concept of using monotonically non-increasing sampling of timesteps t from DreamTime [11]. We find that the development of high-quality, semantically rich 3D representations benefits from ***integrating information across sampling multiple timesteps*** of a pre-trained 2D text-to-image diffusion model at each iteration. This finding contrasts with all other methods using SDS [29], which typically utilize information from a single timestep at each iteration. Specifically, during the initial and middle phases of optimization, which focus on the primary shape formation, we establish a decremental time window T_{end} that linearly decreases with iterations. T_{end} is segmented into m intervals, within each of which t is randomly sampled and gradients are aggregated. This technique swiftly yields semantically rich 3D representations but may accumulate superfluous 3D Gaussians. To address this, we apply 3D Gaussian filtering to sample only the essential surface Gaussians. In the subsequent stages, focusing on surface texture refinement, we sample t within the range of $0 \sim 200$ and employ 3D reconstruction methods [14] to expedite the generation of plausible textures. In this process, we follow the pattern of 3D model formation to sample different timesteps t at different iterations and 3D Gaussians on the model surface, hence we refer to this process as ***Formation Pattern Sampling.*** We leverage pseudo-Ground-Truth (pseudo-GT) images, derived from a single denoising step in LucidDreamer [17], to encapsulate the diverse information provided by different timesteps $t(0 \sim 1000)$ of the 2D text-to-image diffusion model. By adding t timesteps of noise to images x_0 to obtain x_t, we estimate the pseudo-GT $\hat{x}_0^t$ by:

$$\hat{x}_0^t = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\phi(x_t; y, t)}{\sqrt{\bar{\alpha}_t}}. \tag{7}$$

Multi-timestep Sampling. As shown in Fig. 3a, we discovered when timestep t is small, the 2D diffusion model provides more detailed and realistic surface textures, of which the shape is consistent with the current 3D representation, but lacks in providing rich semantic information of prompt y. When t is large, the 2D diffusion model offers abundant semantic information, but its shape may not align with the current 3D representation (e.g., the orientation of the man, the color of the chair, the direction of the cooker between t values of 600 to 800). Therefore, we propose to hybridize information from different 2D diffusion timesteps in each iteration, to ensure a certain degree of shape constraint while enriching the semantic information. For instance, in Fig. 3a, at the 300-th iteration phase of the man, we need to restrict the shape using the information of timesteps in $200 \sim 400$, get richer semantic information using the information of timesteps in $400 \sim 600$ and $600 \sim 800$. However, at the 1000-th iteration of the cooker, we find that the current 3D representation already possesses ample semantic information, and the information provided by larger t might hinder the current optimization direction. Our i-th sampled t can be expressed as:

$$t_i = T_{end}^{iter} \cdot random(\frac{i-1}{m}, \frac{i}{m}), i = 1, ..., m, \tag{8}$$

where T_{end} denotes a linear decreasingly time window similar to DreamTime, $iter$ represents the current iteration, and m denotes the number of intervals. We adopt DDIM Inversion to sample from t_1 to t_m to ensure content consistency [17].

Therefore, the multi-timestep sampling (MTS) combined with the CSD (Classifier Score Distillation) [40] method can be expressed as:

$$\nabla_\theta \mathcal{L}_{\mathrm{MTS}}(\theta) = \mathbb{E}_{t,\epsilon,c}\left[\sum_{i=1}^{m} w(t_i)(\epsilon_\phi(x_{t_i}; y, t_i) - \epsilon_\phi(x_{t_i}; \emptyset, t_i))\frac{\partial g(\theta, c)}{\partial \theta}\right]. \tag{9}$$

3D Gaussian Filtering. Excessive Gaussians can hinder the optimization process. Unlike current methods [5,15] that filter reconstructed 3D Gaussians using ground truth images, our approach necessitates filtering during optimization. In terms of rendering, the 3D Gaussians closer to the rendering plane inherently have a more significant impact, for which we employ a score function tailored to quantify such contributions. For 3D Gaussians along the rendering ray r_j, their contributions are calculated using the inverse square of their distance to the rendering plane, adjusted by the 3D Gaussians volume. This method prioritizes 3D Gaussians that are both closer to the rendering plane and larger in volume as shown in Fig. 8. By ranking various viewpoints according to scores, we can efficiently eliminate 3D Gaussians falling below a designated threshold.

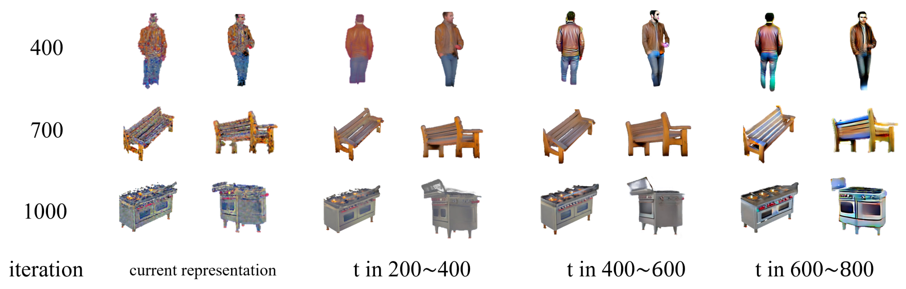
$$Score(i) = \sum_{j=1}^{H\times W\times M} \frac{V(i)}{D(r_j, i)^2 \times maxV(r_j)}, \tag{10}$$

where H and W denote the height and width of the rendered image, M indicates the number of rendered images, $V(i)$ represents the volume of the i-th 3D Gaussian (calculated using the covariance matrix), $maxV(r_j)$ represents the maximum volume of the 3D Gaussians on r_j, and $D(r_j, i)$ is the distance of the i-th 3D Gaussian from the rendering plane along the r_j. Note that this procedure mirrors the rendering process instead of performing the actual rendering.

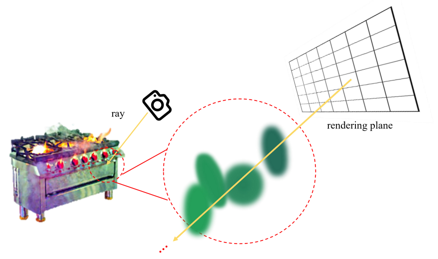
Reconstructive Generation. Next, we can expedite the generation of plausible surface textures using 3D reconstruction methods [14]. During the previous process, we find that when sampling very small timesteps $t(0 \sim 200)$, the image $\hat{x}_0^t$ predicted by Eq. 7 has no difference in 3D shape compared to the image x_0 before noise addition, but offers more detailed and plausible textures. Therefore, under the premise of shape consistency, we can directly ***generate*** a new 3D representation through ***3D reconstruction*** [14]. As shown in Fig. 3c, after obtaining a coarse texture and highly consistent 3D representation, we render K images $x_i, i = 1, ..., K$ from various camera poses c_i around the 3D representation. By adding t timesteps of noise to the images to get x_{it} by Eq. 1, we estimate the images $\hat{x}_{i0}^t$ with plausible textures using Eq. 7, and then reconstruct them on the coarse representation through [14] by minimizing the following reconstruction loss:

$$L_{rec} = \sum_i ||g(\theta, c_i) - \hat{x}_{i0}^t||_2. \tag{11}$$

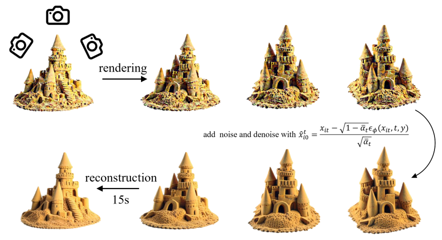
This process generates a representation with detailed and plausible textures within ***15 seconds***.



(a) Multi-timestep Sampling. At different timesteps, the 2D text-to-image diffusion model provides different information (represented by the pseudo-GT $\hat{x}_0^t$) obtained from x_t in a single-step by Eq. 7 in LucidDreamer [17]. We demonstrate it by showcasing three objects at various iterations of the optimization process.



(b) 3D Gaussian Filtering. 3D Gaussians that are closer to the rendering plane and have larger volumes contribute more to the rendering.


rendering
add noise and denoise with $\hat{x}_{i0}^t = \frac{x_{it} - \sqrt{1-\bar{\alpha}_t}\epsilon_\phi(x_{it}, t, y)}{\sqrt{\bar{\alpha}_t}}$
reconstruction
15s

(c) Reconstructive Generation. In the later stages of optimization, generation can be directly achieved through reconstruction based on the denoised images, resulting in 3D representation with fined and plausible textures.

Fig. 3: Formation Pattern Sampling.

4.2 Camera Sampling

To ensure generation quality, current methods [4, 12, 26, 38, 41] often conduct camera sampling within a limited range, failing to guarantee scene-wide observations. Straightforward random camera sampling within the scene can cause scene generation to collapse during optimization. Thus, we propose an incremental three-stage camera sampling strategy:

In the first stage, we generate a coarse representation of the surroundings (indoor walls and outdoor faraway surroundings). We freeze the 3D Gaussians parameters of the ground and objects, sampling camera coordinates within a certain range of the center to optimize the generation of the surroundings.

The second stage focuses on generating the coarse ground. We freeze the 3D Gaussians parameters of the environments and objects. The indoor scene is divided into regions based on the placement of objects, we sample camera poses pointing at these focal regions containing objects and the ground of the room in each iteration. The outdoor scene is divided into concentric circles based on radius, and in each iteration, a same direction is chosen to sample camera poses on different circles to optimize ground generation. Our strategy is able to cover the entire ground as uniformly as possible and focuses on optimizing the parts where the ground is in contact with objects and the surroundings.

In the third stage, we use all the camera poses above to ensure a scene-wide view and refine all environmental elements. This includes optimizing parameters for both the ground and the surrounding environments. Based on the 3D consistency of the previous phases, we then proceed to the reconstructive generation method to obtain more detailed and plausible textures.

Generated camera positions may be obstructed by placed objects, necessitating collision detection between the camera and objects. If a collision occurs, the generated camera should be discarded. Due to space constraints, we have placed the details on the camera sampling strategy in the Supplementary Materials.

5 Experiment

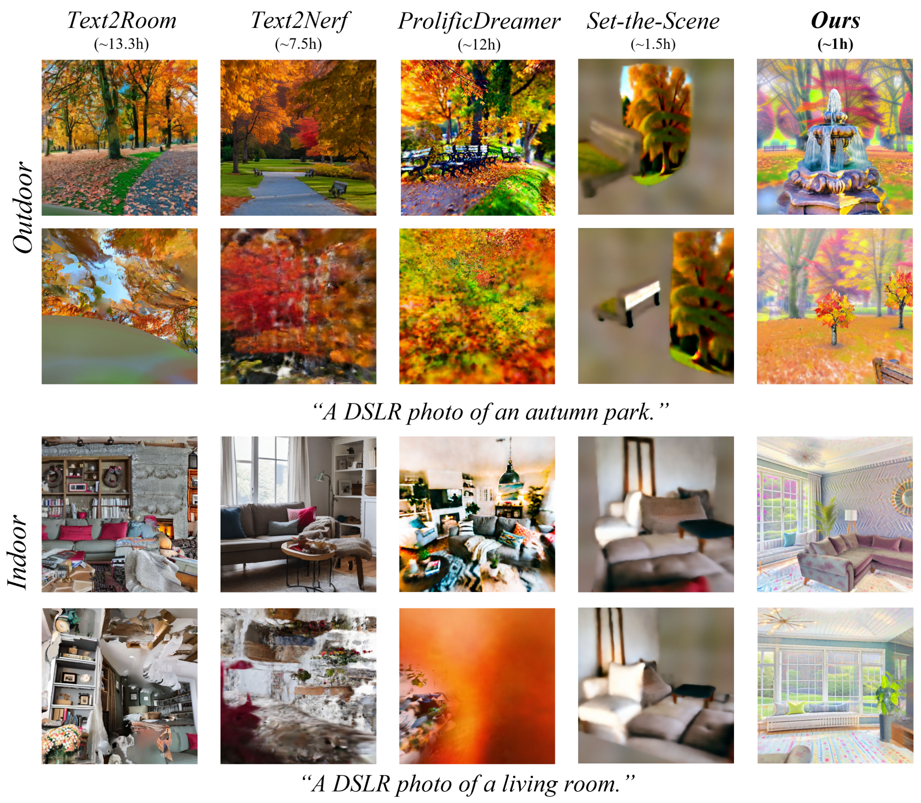
Implementation Details. We utilize GPT-4 as our LLM for scene prompt decomposition, Point-E [25] generated sparse point clouds for the initial representation of objects, Stable Diffusion 2.1 as our 2D text-to-image model. The maximum number of iterations is set to 1,500 and 2,000 rounds for objects and the environment, respectively. The initial time interval value m, starts at 4 and decreases by 1 every 400 rounds. The number of rendering images in the reconstructive generation is set to 20. We tested DreamScene and all the baselines on the same NVIDIA 3090 GPU for fair comparison.

Baselines. For the comparison of the text-to-3D scene generation, we use the current open-sourced SOTA methods Text2Room [10], Text2NeRF [41], ProlificDreamer [38], and Set-the-Scene [4] as the baselines. For text-to-3D generation, we select open-source SOTA methods DreamFusion [29], Magic3D [18], DreamGaussian [36], and LucidDreamer [17] as the baselines (ProlificDreamer, DreamFusion and Magic3D were reimplemented by Three-studio [6]).

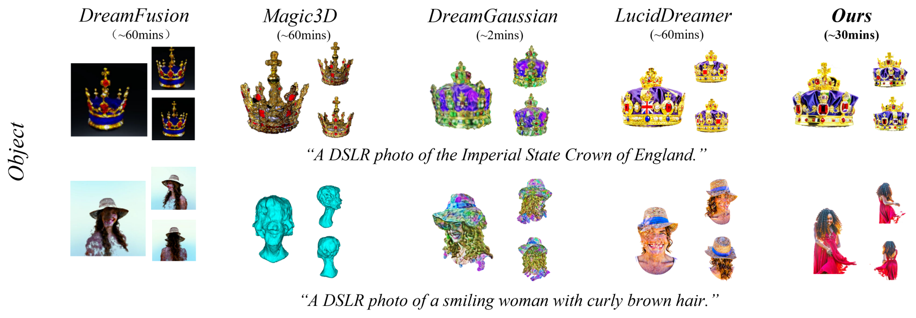
Evaluation Metrics. We tested the generation time of each method [4,10,17,18,29,36,38,41], compared the editing capabilities against their published papers, and did a 100-participant user study to score (out of 5) the quality, consistency, and rationality of the videos generated by each method for 5 scenes (3 indoor, 2 outdoor) of 30 seconds.

5.1 Qualitative Results

Fig. 4a shows the comparison of DreamScene in indoor and outdoor scenes with the SOTA methods [4,12,38,41]. The upper images are rendered with the camera poses that appeared during the generation, and the lower ones with the randomly selected camera poses within the scene. We can see that Text2Room [10] and Text2NeRF [41] only produce satisfactory results under camera poses encountered during generation. Combined with the results from Fig. 5, it shows that DreamScene achieves the best 3D consistency alongside commendable generation quality. For the comparison of generating single objects with text-to-3D



(a) Scene Results. The upper part shows images rendered using camera poses employed during the generation process, whereas the lower part shows the images rendered with the randomly selected camera poses within the scene.


DreamFusion
(~60mins)
Magic3D
(~60mins)
DreamGaussian
(~2mins)
LucidDreamer
(~60mins)
Ours
(~30mins)
Object
"A DSLR photo of the Imperial State Crown of England."
"A DSLR photo of a smiling woman with curly brown hair."

(b) Object Results.

Fig. 4: Comparison with baselines in text-to-3D generation.

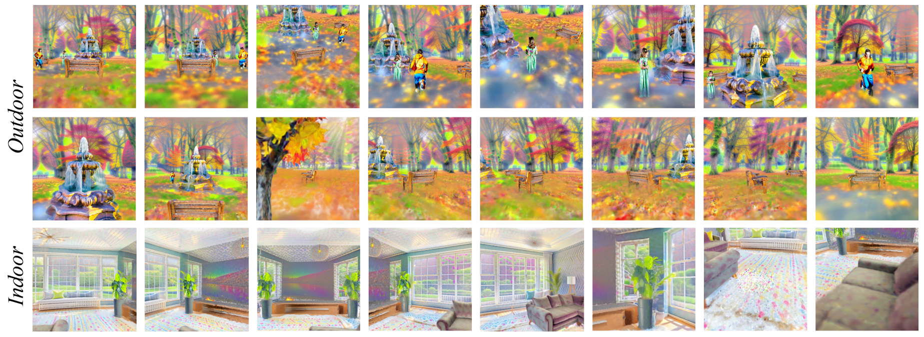


Fig. 5: Consistency results under multiple scene-wide camera poses.

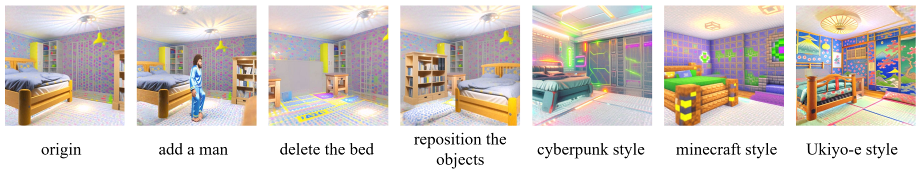


Fig. 6: DreamScene editing results.

methods, Fig. 4b shows that our FPS can generate high-quality 3D representation following the text prompts in a short time. DreamGaussian [36] is faster yet at the cost of too low generation quality. Please refer to the Supplementary Materials for additional images and video results.

5.2 Quantitative Results

Since baselines [10, 38, 41] are not able to generate objects in the environment independently, for a fair comparison we calculate the generation time of our environment generation stage. The left side of Tab. 1 shows that we have the shortest generation time for environments with editing capabilities, and the right side shows the user study, where DreamScene is far ahead of [4, 10, 38, 41] in terms of consistency and rationality with high generation quality.

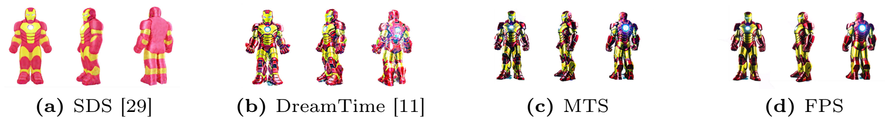
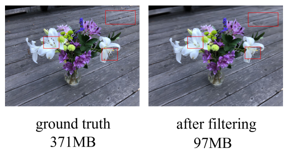
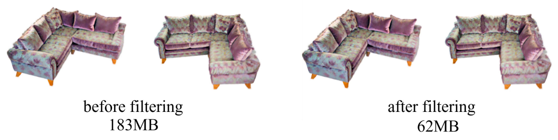


Fig. 7: The ablation results of different sampling strategies.



(a) data in NeRF-360 [22]



(b) data in generating process

Fig. 8: Ablation results of 3D Gaussian filtering algorithm in reconstruction and generation tasks.

Table 1: Quantitative Results of DreamScene compared with baselines. ↑ means the more the better and ↓ means the lower the better.

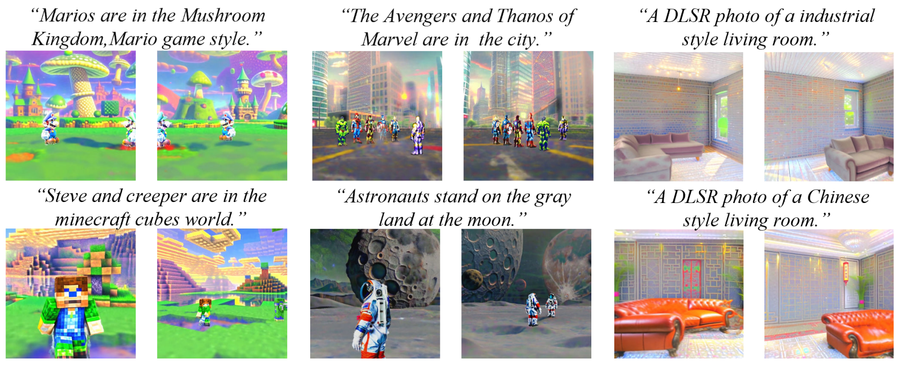
Method	Time (hours) ↓	Editing	User Study		
			quality↑	consistency↑	rationality↑
Text2Room [10]	13.3	✗	2.93	2.57	2.60
Text2NeRF [41]	7.5	✗	3.05	2.71	2.98
ProlificDreamer [38]	12.0	✗	3.48	3.19	2.95
Set-the-Scene [4]	1.5	✓	2.45	3.52	2.88
Ours	**1.0**	✓	**3.92**	**4.24**	**4.05**

5.3 Scene Editing

Fig. 6 demonstrates the flexible editing capabilities of DreamScene. DreamScene can add or remove an object or ressign its position in the scene by adjusting the values of the object's affine components. When making these edits, we need to resample camera poses at the original and new locations of the object, re-optimizing towards the ground and surrounding directions. We can also change the text prompts to alter the style of the environments or the objects in the scene. This editing can be obtained after re-optimizing the coarse 3D representation.

5.4 Ablations

We compared the effects of different sampling strategies via the generation results of a 3D object. Fig. 7 shows the results optimized for 30 minutes under the prompt "A DSLR photo of Iron Man". As illustrated, multi-timestep sampling (MTS) forms better geometric structures and textures compared to the monotonically non-increasing sampling in [11] and the SDS in [29]. Formation Pattern Sampling (FPS), building on top of MTS, employs a reconstruction method to create smoother and more plausible textures. The 3D Gaussian filtering we proposed is specifically designed for optimization tasks and can also be directly applied to reconstruction tasks [14]. Fig. 8 compares the results of reconstruction and generation before and after compression using the Gaussian filtering algorithm. It can be seen that in the reconstruction task, our compression achieved 73.9%, with the overall image slightly blurred and some details lost. In our generation task, however, the compression was 66.1% without signif-


"Marios are in the Mushroom Kingdom,Mario game style."
"The Avengers and Thanos of Marvel are in the city."
"A DLSR photo of a industrial style living room."
"Steve and creeper are in the minecraft cubes world."
"Astronauts stand on the gray land at the moon."
"A DLSR photo of a Chinese style living room."

(a) gaming **(b)** film **(c)** architectural design

Fig. 9: The potential applications of DreamScene.

icant quality loss. Please find the ablation experiments on camera sampling and more comprehensive results in the Supplementary Materials.

6 Applications, Limitation and Conclusion

DreamScene can impact numerous industries as illustrated in Fig. 9. For instance, in VR gaming and the Metaverse, DreamScene, integrated with LLMs, enables the capability to generate a fantasy scene merely from a user's description. Moreover, it allows for easy modification of object positions and the style of the scene through descriptive means. DreamScene can also revolutionize film production by enabling rapid and customizable generation of intricate 3D scenes directly from scripts, significantly enhancing visual storytelling and reducing the reliance on physical sets and manual modeling. For architectural design, DreamScene facilitates the generation of furniture in various styles, enabling users to flexibly design and arrange objects within a scene using layout diagrams.

DreamScene currently cannot generate outdoor scenes as realistically close to actual environments as Text2Room [10] and Text2NeRF [41]. This is because their methods generate scenes through image inpainting, resulting in very limited observable camera poses and poor 3D consistency across the entire scene. In contrast, we sacrificed a degree of realism to ensure overall scene consistency. In future work, we plan to incorporate depth supervision to guide the generation of outdoor scenes with a style as realistic as indoor ones.

In summary, we introduce a novel text-to-3D scene generation strategy DreamScene. By employing Formation Pattern Sampling (FPS), a camera sampling strategy, integrating objects and environments, we address the current issues of inefficiency, inconsistency, and limited editability in current text-to-3D scene generation methods. Extensive experiments have demonstrated that DreamScene is a milestone achievement in 3D scene generation, holding potential for wide-ranging applications across numerous fields.

Acknowledgements: This work was supported by the National Key Research and Development Program of China (2022YFB3105405, 2021YFC3300502).

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021)
2. Chen, G., Wang, W.: A survey on 3d gaussian splatting. arXiv preprint arXiv:2401.03890 (2024)
3. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. arXiv preprint arXiv:2303.13873 (2023)
4. Cohen-Bar, D., Richardson, E., Metzer, G., Giryes, R., Cohen-Or, D.: Set-the-scene: Global-local training for generating controllable nerf scenes. arXiv preprint arXiv:2303.13450 (2023)
5. Fan, Z., Wang, K., Wen, K., Zhu, Z., Xu, D., Wang, Z.: Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. arXiv preprint arXiv:2311.17245 (2023)
6. Guo, Y.C., Liu, Y.T., Wang, C., Zou, Z.X., Luo, G., Chen, C.H., Cao, Y.P., Zhang, S.H.: threestudio: A unified framework for 3d content generation (2023)
7. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
8. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
9. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
10. Höllein, L., Cao, A., Owens, A., Johnson, J., Nießner, M.: Text2room: Extracting textured 3d meshes from 2d text-to-image models. arXiv preprint arXiv:2303.11989 (2023)
11. Huang, Y., Wang, J., Shi, Y., Qi, X., Zha, Z.J., Zhang, L.: Dreamtime: An improved optimization strategy for text-to-3d content creation. arXiv preprint arXiv:2306.12422 (2023)
12. Hwang, I., Kim, H., Kim, Y.M.: Text2scene: Text-driven indoor scene stylization with part-aware details. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1890–1899 (2023)
13. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. arXiv preprint arXiv:2305.02463 (2023)
14. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)
15. Lee, J.C., Rho, D., Sun, X., Ko, J.H., Park, E.: Compact 3d gaussian representation for radiance field. arXiv preprint arXiv:2311.13681 (2023)
16. Li, W., Chen, R., Chen, X., Tan, P.: Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. arXiv preprint arXiv:2310.02596 (2023)
17. Liang, Y., Yang, X., Lin, J., Li, H., Xu, X., Chen, Y.: Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. arXiv preprint arXiv:2311.11284 (2023)

18. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 300–309 (2023)
19. Lin, Y., Bai, H., Li, S., Lu, H., Lin, X., Xiong, H., Wang, L.: Componerf: Text-guided multi-object compositional nerf with editable 3d scene layout. arXiv preprint arXiv:2303.13843 (2023)
20. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023)
21. Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-nerf for shape-guided generation of 3d shapes and textures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12663–12673 (2023)
22. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
23. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6038–6047 (2023)
24. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
25. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts. arXiv preprint arXiv:2212.08751 (2022)
26. Ouyang, H., Heal, K., Lombardi, S., Sun, T.: Text2immersion: Generative immersive scene with 3d gaussians. arXiv preprint arXiv:2312.09242 (2023)
27. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 165–174 (2019)
28. Po, R., Wetzstein, G.: Compositional 3d scene generation using locally conditioned diffusion. arXiv preprint arXiv:2303.12218 (2023)
29. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
30. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
31. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: 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)
32. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022)
33. Shen, T., Gao, J., Yin, K., Liu, M.Y., Fidler, S.: Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. Advances in Neural Information Processing Systems **34**, 6087–6101 (2021)

34. Shi, Y., Wang, P., Ye, J., Long, M., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. arXiv preprint arXiv:2308.16512 (2023)
35. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
36. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023)
37. Vilesov, A., Chari, P., Kadambi, A.: Cg3d: Compositional generation for text-to-3d via gaussian splatting. arXiv preprint arXiv:2311.17907 (2023)
38. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. Advances in Neural Information Processing Systems **36** (2024)
39. Yi, T., Fang, J., Wu, G., Xie, L., Zhang, X., Liu, W., Tian, Q., Wang, X.: Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. arXiv preprint arXiv:2310.08529 (2023)
40. Yu, X., Guo, Y.C., Li, Y., Liang, D., Zhang, S.H., Qi, X.: Text-to-3d with classifier score distillation. arXiv preprint arXiv:2310.19415 (2023)
41. Zhang, J., Li, X., Wan, Z., Wang, C., Liao, J.: Text2nerf: Text-driven 3d scene generation with neural radiance fields. IEEE Transactions on Visualization and Computer Graphics (2024)
42. Zhang, Q., Wang, C., Siarohin, A., Zhuang, P., Xu, Y., Yang, C., Lin, D., Zhou, B., Tulyakov, S., Lee, H.Y.: Scenewiz3d: Towards text-guided 3d scene composition. arXiv preprint arXiv:2312.08885 (2023)
43. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)
44. Zhou, X., Ran, X., Xiong, Y., He, J., Lin, Z., Wang, Y., Sun, D., Yang, M.H.: Gala3d: Towards text-to-3d complex scene generation via layout-guided generative gaussian splatting. arXiv preprint arXiv:2402.07207 (2024)

Supplementary Material

7 Discussions of Current Methods

Inpainting-based methods employ text-to-image inpainting for scene generation [10, 26, 41]. They first initialize an image and then partially mask it to represent a view from an alternate angle. Leveraging pretrained image inpainting models like Stable Diffusion [31], alongside depth estimation, they reconstruct the occluded image segments and deduce their depths, iteratively composing the entire scene through depth and image alignment. While such methods can achieve good visual effects at the camera positions (e.g., the center of the scene) during the generation process, they encounter significant limitations in visible range. Venturing beyond the predefined camera areas used during generation precipitates scene degradation, as exemplified in Fig. 4a of the paper, underscoring a deficiency in scene-wide 3D consistency. Concurrently, this method's tendency to replicate certain environmental objects, like the proliferation of sofas in a living room scenario, highlights issues with logical scene composition.
Combination-based methods also employ a combination approach to construct scenes [4,42]. However, they face challenges including low generation quality and slow training speeds. Moreover, [42] utilizes various 3D representations (such as NeRF+DMTet) to integrate objects and scenes, increasing the complexity of scene representation and thus limiting the number of objects that can be placed within the scene (2-3 objects), impacting their applicability. In contrast, DreamScene's Formation Pattern Sampling (FPS) can generate high-quality 3D content in a very short time, using a single 3D representation to compose the entire scene, allowing for more than 20 objects to be placed within the scene. This underscores DreamScene's remarkable superiority.
Objects combination methods do not take the environmental context into account, focusing solely on whether the combination of objects is logical [19, 37, 44]. They generate a simple assembly of objects rather than a complete scene. We believe that the approach to scene composition should be more diverse and offer greater controllability.

DreamScene demonstrates a significant advantage by efficiently, consistently, and flexibly generating 3D scenes, showcasing a substantial superiority over the aforementioned methods.

8 Additional Implementation Details

The overall generation process of DreamScene is shown in Algorithm 1.

8.1 Rendering and Training Settings

We render images with a size of 512×512 for optimization. During the optimization process, we do not reset the opacity, to maintain the consistency of the optimization during the training process and avoid gradient disappearance due to opacity reset.

Algorithm 1 DreamScene

```
1: $Y \rightarrow y_1, y_2, ..., y_N, y_e$;
2: for $n = [1, 2, ..., N, e]$ do
3:     if $n$ is not $e$ then
4:         Initialize 3D Gaussian of $obj_n$
5:     else
6:         Initialize 3D Gaussian of environment
7:     end if
8:     for $iter = [0, 1, ..., max\_iter]$ do
9:         if $n$ is not $e$ then
10:            Sample camera pose $c$
11:        else
12:            Sample camera pose $c$ follow strategy in Sec. 8.2
13:        end if
14:        $x_0 = g(\theta, c)$
15:        $T_{end} = (1 - \frac{iter}{max\_iter})timesteps$
16:        for $i = [1, 2, ..., m]$ do
17:            $t_i = T_{end} \cdot random(\frac{i-1}{m}, \frac{i}{m})$
18:            $x_i = DDIM_(x_{i-1}, i)$
19:            $\epsilon_\phi(x_{t_i}; y_n, t_i)$ =U-Net$(x_{t_i}, y_n, t_i)$
20:            $\epsilon_\phi(x_{t_i}; \emptyset, t_i)$ =U-Net$(x_{t_i}, \emptyset, t_i)$
21:        end for
22:        $\nabla_\theta \mathcal{L}_{\text{MTS}}(\theta) = \mathbb{E}_{t,\epsilon,c}\left[\sum_{i=1}^{m} w(t_i)(\epsilon_\phi(x_{t_i}; y_n, t_i) - \epsilon_\phi(x_{t_i}; \emptyset, t_i))\frac{\partial g(\theta, c)}{\partial \theta}\right]$
23:        Update $\theta$
24:        if $iter\%compress\_iter = 0$ then
25:            $Score_k = \sum_{j=1}^{H \times W \times M} \frac{V(k)}{D(r_j, k)^2 \times maxV(r_j)}$
26:            $Sort(Score_k)$
27:            Delete last $z$ 3D Gaussians
28:        end if
29:    end for
30:    if $n$ is $e$ then
31:        Save 3D Gaussian Representation of the Scene
32:        break
33:    end if
34:    Save 3D Gaussian Representation $obj_n$ of text $y_n$
35:    $world(x) = r \cdot s \cdot obj_n(x) + t$
36:    Add $obj_n$ to the Scene
37: end for
```

8.2 Camera Sampling Strategy

This section outlines a three-stage camera sampling strategy for crafting both outdoor and indoor scenes. The process is as follows:

Outdoor

- In the first stage, we freeze the parameters of the ground and objects, focusing solely on optimizing the surrounding environments without rendering the objects. During this phase, we sample cameras near the center of the scene, the camera's pitch angle is set between 80 to 110 degree. After reaching 70% of the iterations of this stage, we select four camera poses for multi-camera sampling at each later iteration. These four cameras, all directed towards the same direction, are positioned on either side of the scene center at distances of either 1/4 or 1/2 of the radius, ensuring the environments achieve satisfactory visual effects across various distances.
- In the second stage, we freeze the parameters of the surrounding environments and objects, and focus solely on optimizing the ground without rendering objects. We sample four camera poses at each iteration, akin to the sampling strategy in the later part of the first stage. We adjust the pitch angle range to 85∼95 degree, which can reduce the occurrence of a singular ground or environment in the rendered images, thereby enhancing the overall scene generation outcome.
- In the third stage, we optimize both the surrounding environments and the ground, and render objects into the scene to achieve a harmonious and unified effect. We integrate the camera positions used in the previous two stages, ensuring that areas within the scene are evenly and comprehensively covered, thereby attaining a consistent reconstruction result.

Indoor

- In the first stage, we freeze the ground and object parameters and **render** the objects into the scene. We primarily sample camera poses around the center of the scene and set the radius as large as possible to encompass all objects and thereby minimize the multi-head problem. At this stage, the pitch angle range of cameras is set to 75∼115 degree. After the same iterations at outdoor settings, we sample camera poses around the objects to reduce the impact of object occlusion on the environment.
- In the second stage, we freeze the environment parameters and begin optimizing the ground parameters. The indoor camera sampling strategy remains largely unchanged, but we adjust the pitch angle range to 45∼90 degree to ensure coverage of the ground. Additionally, we increase the camera sampling around objects and from the center to the periphery of the scene, thereby enhancing the integration between the ground and the walls.
- In the third stage, we optimize both the surrounding environments and the ground, and render objects into the scene in the same way as outdoor's.

Our indoor and outdoor camera sampling strategies correspond to typical bounded and unbounded scenes, respectively. For bounded scenes, we focus on ensuring consistency across various orientations. Hence we pay more attention to integrating objects with environments to avoid illogical layouts caused by generating excessive objects in the environment. For unbounded scenes, our primary concern lies with maintaining scene-wide consistency across varied distances. Therefore, we employ two different strategies for scene generation.

9 Additional Results

9.1 Qualitative Results

More qualitative results of FPS are shown in Fig. 10. More qualitative results of scene generation are shown in Fig. 11 and Fig. 12.

9.2 Ablation Study

Fig. 13(a) depicts a scene generated by randomly sampling cameras within the scene. Due to the difficulty in ensuring consistency of multi-angle views at the same location, the optimization process often tends to collapse. Fig. 13(b) utilizes a strategy that initiates from the center to the surroundings, where the environment and ground are not differentiated. It can be observed that while scene consistency improves, the connection between the ground and the scene is poorly generated, and the ground is prone to coarse Gaussian points. Fig. 13(c) employs our three-phase strategy, enhancing the generation quality while ensuring the consistency of the surrounding environment and ground.

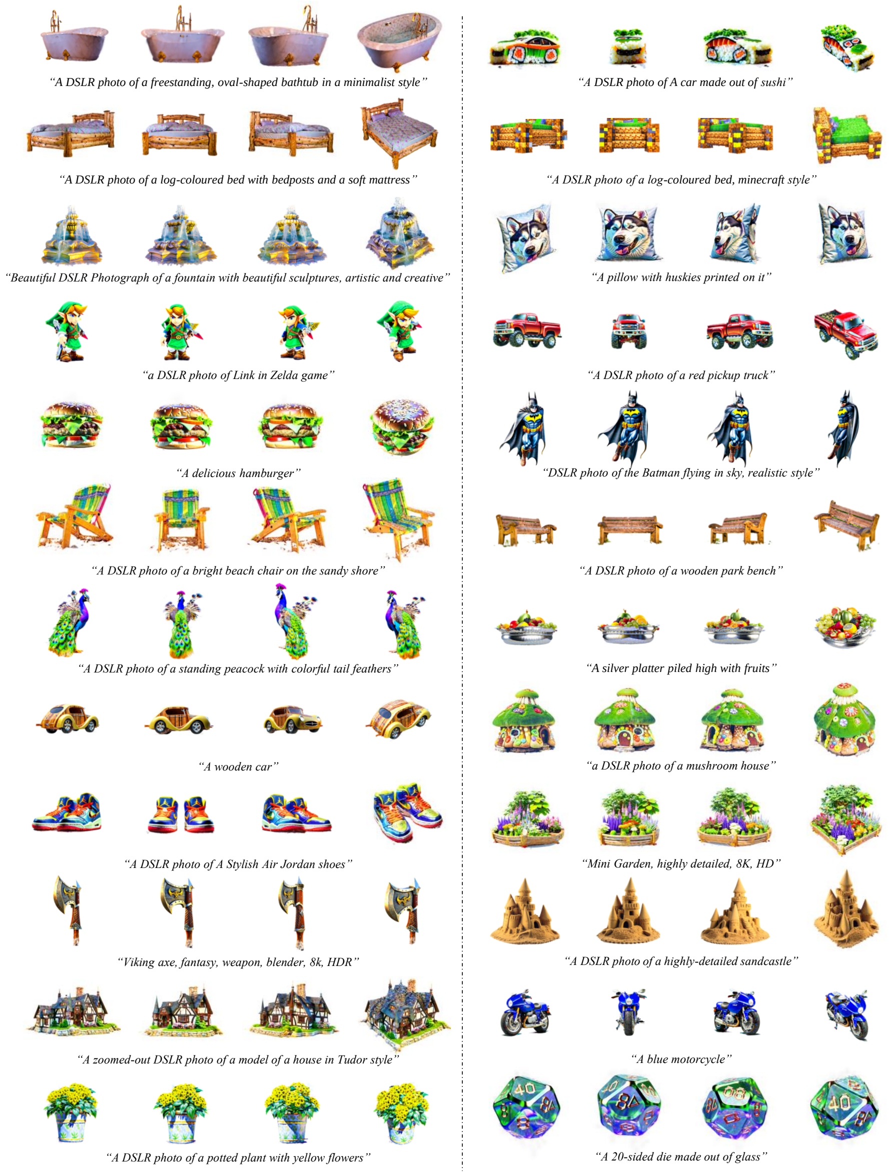
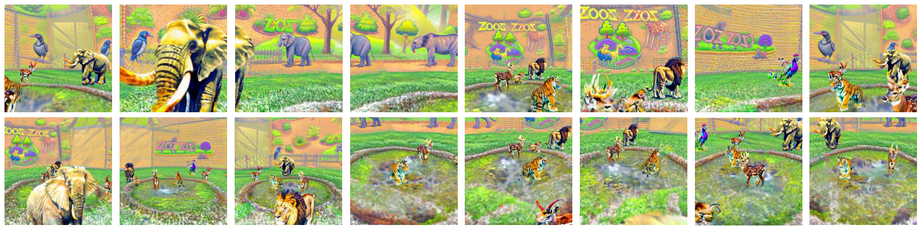
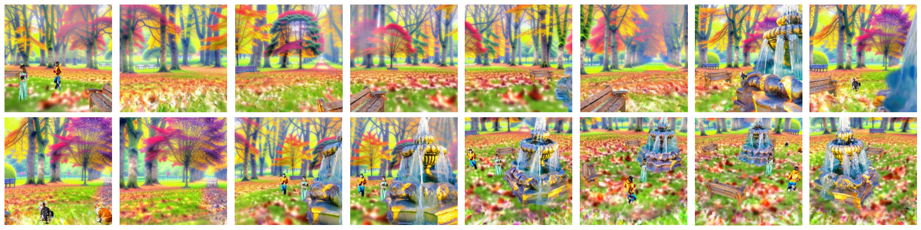

"A DSLR photo of a freestanding, oval-shaped bathtub in a minimalist style"
"A DSLR photo of A car made out of sushi"
"A DSLR photo of a log-coloured bed with bedposts and a soft mattress"
"A DSLR photo of a log-coloured bed, minecraft style"
"Beautiful DSLR Photograph of a fountain with beautiful sculptures, artistic and creative"
"A pillow with huskies printed on it"
"a DSLR photo of Link in Zelda game"
"A DSLR photo of a red pickup truck"
"A delicious hamburger"
"DSLR photo of the Batman flying in sky, realistic style"
"A DSLR photo of a bright beach chair on the sandy shore"
"A DSLR photo of a wooden park bench"
"A DSLR photo of a standing peacock with colorful tail feathers"
"A silver platter piled high with fruits"
"A wooden car"
"a DSLR photo of a mushroom house"
"A DSLR photo of A Stylish Air Jordan shoes"
"Mini Garden, highly detailed, 8K, HD"
"Viking axe, fantasy, weapon, blender, 8k, HDR"
"A DSLR photo of a highly-detailed sandcastle"
"A zoomed-out DSLR photo of a model of a house in Tudor style"
"A blue motorcycle"
"A DSLR photo of a potted plant with yellow flowers"
"A 20-sided die made out of glass"

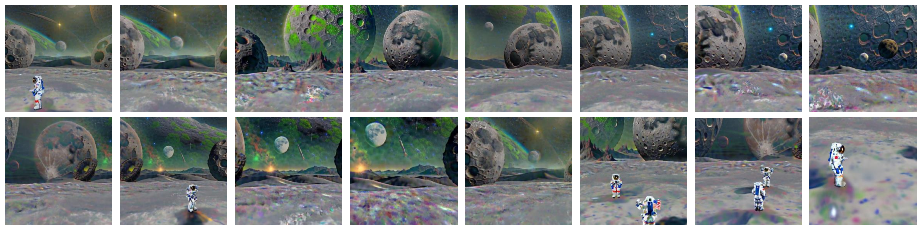
Fig. 10: More object generation results of DreamScene.



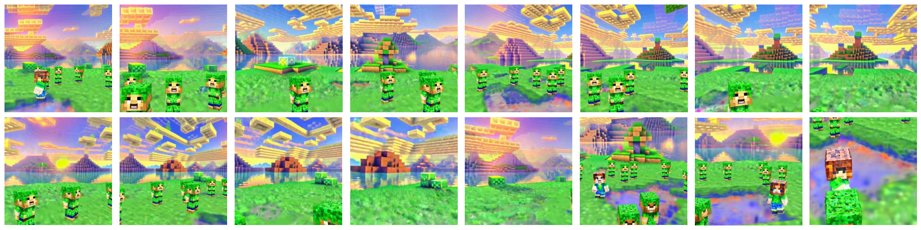
"A DSLR photo in the open area of the zoo"



"A DSLR photo of an autumn park"



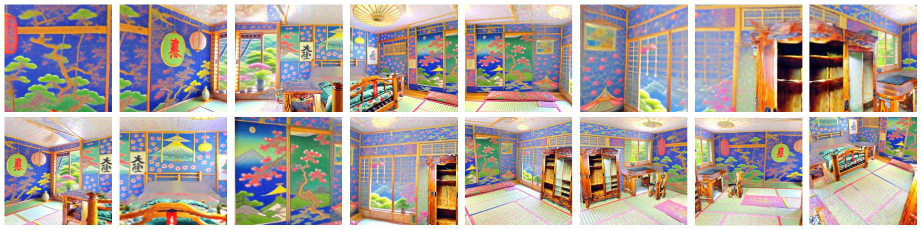
"Gray land at the moon, black tranquil universe in the distance, Sci-fi style"



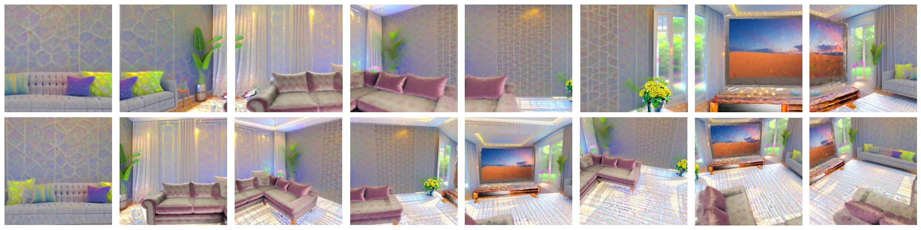
"A minecraft cubes world with lake and mountains in the far distance and grass cubes in the near distance"

Fig. 11: More outdoor scene generation results of DreamScene.

"A DSLR photo of an European style kitchen"

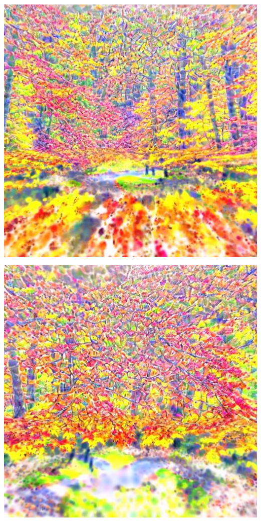


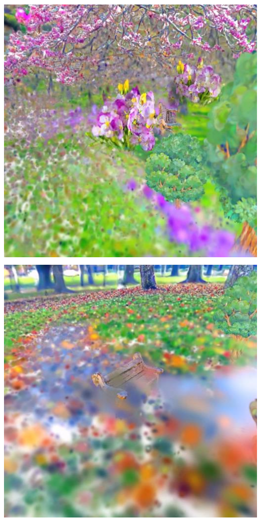
"A DSLR photo of an Ukiyo-e style bedroom, Ukiyo-e style"



"A DSLR photo of a living room"

Fig. 12: More indoor scene generation results of DreamScene.


(a) Randomly camera sampling


(b) No distinction between environment and ground

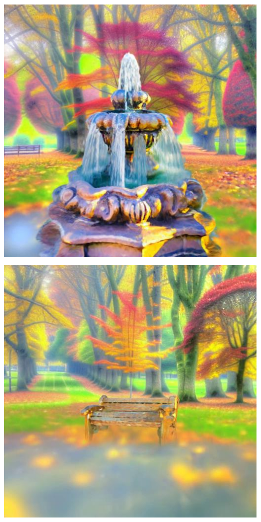

(c) DreamScene three-stage camera sampling strategy

Fig. 13: The ablation results of different camera sampling strategies.