Move Anything with Layered Scene Diffusion

Jiawei Ren[1,2,*] Mengmeng Xu[1] Jui-Chieh Wu[1] Ziwei Liu[2] Tao Xiang[1] Antoine Toisoul[1]
[1]Meta AI [2]S-Lab, Nanyang Technological University
{jiawei011,ziwei.liu}@ntu.edu.sg {frostxu,jerryjcw,txiang,atoisoul}@meta.com

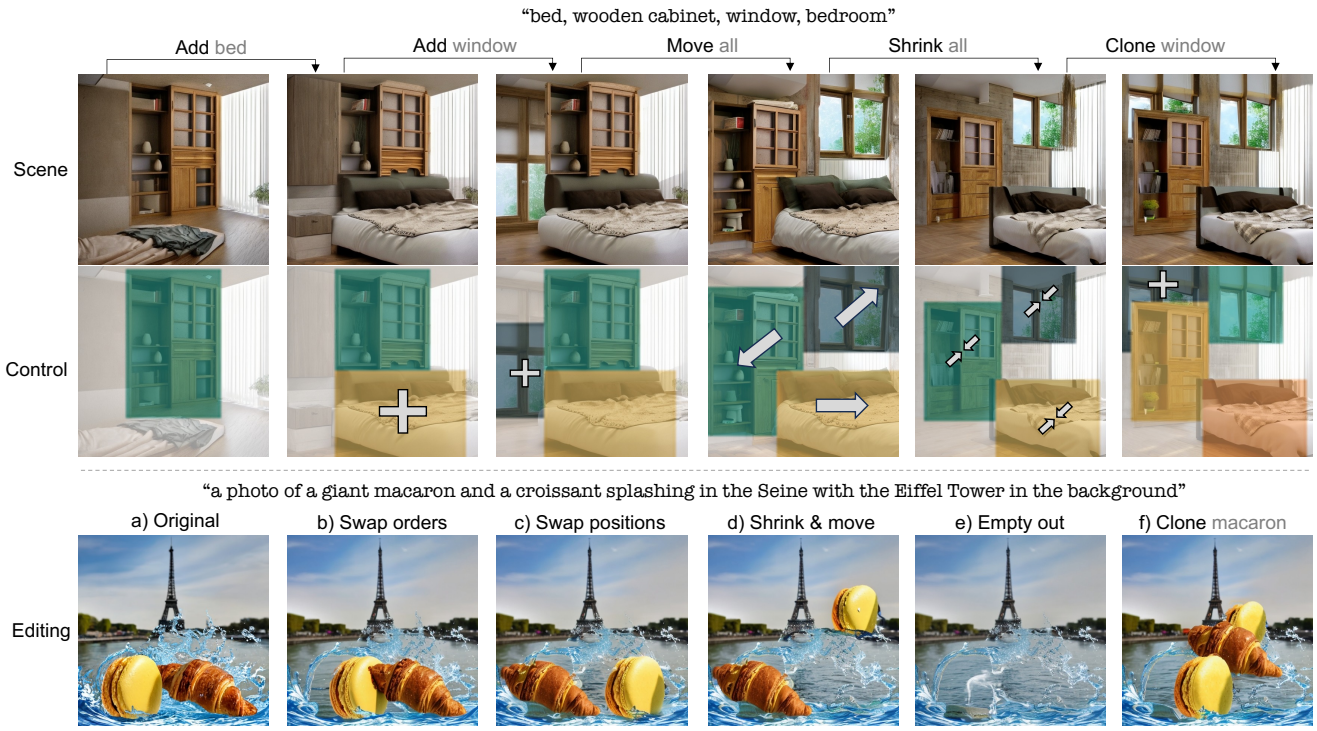

Figure 1. **Move anything on an image. Top**: our approach generates *playable* scenes: objects are spatially disentangled, thus can be freely moved, resized, and cloned in the scene. **Bottom**: a scene can be generated conditioned on a reference image, thus supporting extensive spatial image editing operations. Our approach is training-free and compatible with general text-to-image diffusion models. Once optimized, rendering a new layout requires *less than a second* on a single GPU, allowing interactive interactions.

Abstract

Diffusion models generate images with an unprecedented level of quality, but how can we freely rearrange image layouts? Recent works generate controllable scenes via learning spatially disentangled latent codes, but these methods do not apply to diffusion models due to their fixed forward process. In this work, we propose ***SceneDiffusion*** *to optimize a layered scene representation during the diffusion sampling process. Our key insight is that spatial disentanglement can be obtained by jointly denoising scene renderings at different spatial layouts. Our generated scenes support a wide range of spatial editing operations, including moving, resizing, cloning, and layer-wise appearance editing operations, including object restyling and replacing. Moreover, a scene can be generated conditioned on a reference image, thus enabling object moving for in-the-wild images. Notably, this approach is training-free, compatible with general text-to-image diffusion models, and responsive in less than a second.*

*Work done during an internship at Meta AI.

1. Introduction

Controllable scene generation, *i.e.*, the task of generating images with rearrangeable layouts, is an important topic of generative modeling [27, 42] with applications ranging from content generation and editing for social media platforms to interactive interior design and video games.

In the GAN era, latent spaces have been designed to offer a mid-level control on generated scenes [7, 26, 40, 41]. Such latent spaces are optimized to provide a disentanglement between scene layout and appearance in an unsupervised manner. For instance, BlobGAN [7] uses a group of splattering blobs for 2D layout control, and GIRAFFE [26] uses compositional neural fields for 3D layout control. Although these methods provide good control of the scene layout, they remain limited in the quality of the generated images. On the other hand, diffusion models have recently shown unprecedented performance at the text-to-image (T2I) generation task [4, 6, 13, 31, 33, 35]. Still, they cannot provide fine-grained spatial control due to the lack of mid-level representations stemming from their fixed forward noising process [13, 35].

In this work, we propose a framework to bridge this gap and allow for controllable scene generation with a general pretrained T2I diffusion model. Our method, entitled **SceneDiffusion**, is based on the core observation that spatial-content disentanglement can be obtained during the diffusion *sampling* process by denoising multiple scene layouts at each denoising step. More specifically, at each diffusion step t, we optimize a scene representation by first randomly sampling several scene layouts, running locally conditioned denoising on each layout in parallel, and then analytically optimizing the representation for the next diffusion step $t-1$ to minimize its distance with each of denoised result. We employ a *layered* scene representation [15, 16, 19], where each layer represents an object with its shape controlled by a mask and its content controlled by a text description, allowing us to compute object occlusions using depth ordering. Rendering of the layered representation is done by running a short schedule of image diffusion, which is usually completed within a second. Overall, SceneDiffusion generates rearrangable scenes without requiring finetuning on paired data [25, 43], mask-specific training [31], or test-time optimization [29, 39], and is agnostic to denoiser architecture designs.

In addition, to enable in-the-wild image editing, we propose to use the sampling trajectory of the reference image as an *anchor* in SceneDiffusion. When denoising multiple layouts simultaneously, we increase the weight of the reference layout in the noise update to keep the scene's faithfulness to the reference content. By disentangling the spatial location and visual appearance of the contents, our approach better reduces hallucinations and preserves the overall content across different editing compared to baselines [8, 20, 24].

To quantify the performance, we build an evaluation benchmark by creating a dataset containing 1,000 text prompts and over 5,000 images associated with image captions, local descriptions, and mask annotations. We evaluate our proposed approach on this dataset and show that it outperforms prior works on both image quality and layout consistency metrics by a clear margin on both controllable scene generation and image spatial editing tasks.

In summary, our contributions are:

- We propose a novel sampling strategy, *SceneDiffusion*, to generate layered scenes with image diffusion models.
- We show that the layered scene representation supports flexible layout rearrangements, enabling interactive scene manipulation and in-the-wild image editing.
- We build an evaluation benchmark and observe that our method achieves state-of-the-art performance quantitatively on both scene generation and image editing tasks.

2. Related Works

2.1. Controllable Scene Generation

Generating controllable scenes has been an important topic in generative modeling [27, 42] and has been extensively studied in the GAN context [7, 26, 40, 41]. Various approaches have been developed on applications that include controllable image generation [7, 40], 3D-aware image generation [2, 14, 26, 41] and controllable video generation [21]. Usually, control at the mid-level is obtained in an unsupervised manner by building a spatially disentangled latent space. However, such techniques are not directly applicable to T2I diffusion models. Diffusion models employ a fixed forward process [13, 35], which constrains the flexibility of learning a spatially disentangled mid-level representation. In this work, we solve this issue by optimizing a layered scene representation during the diffusion *sampling* process. It is also noteworthy that recent works enable diffusion models to generate images grounded on given layouts [9, 18, 25, 43]. However, they do not focus on spatial disentanglement and do not guarantee similar content after rearranging layouts.

2.2. Diffusion-based Image Editing

Off-the-shelf T2I diffusion models can be powerful image editing tools. With the help of inversion [23, 36] and subject-centric finetuning [10, 32], various approaches have been proposed to achieve image-to-image translation including concept replacement and restylization [5, 11, 17, 22, 38]. However, these approaches are restricted to in-place editing, and editing the spatial location of objects has been rarely explored. Moreover, many of the approaches exploit an attention correspondence [3, 8, 11, 38] or a feature correspondence [24, 34, 37] with the final image, making the approach dependent to a specific denoiser architec-

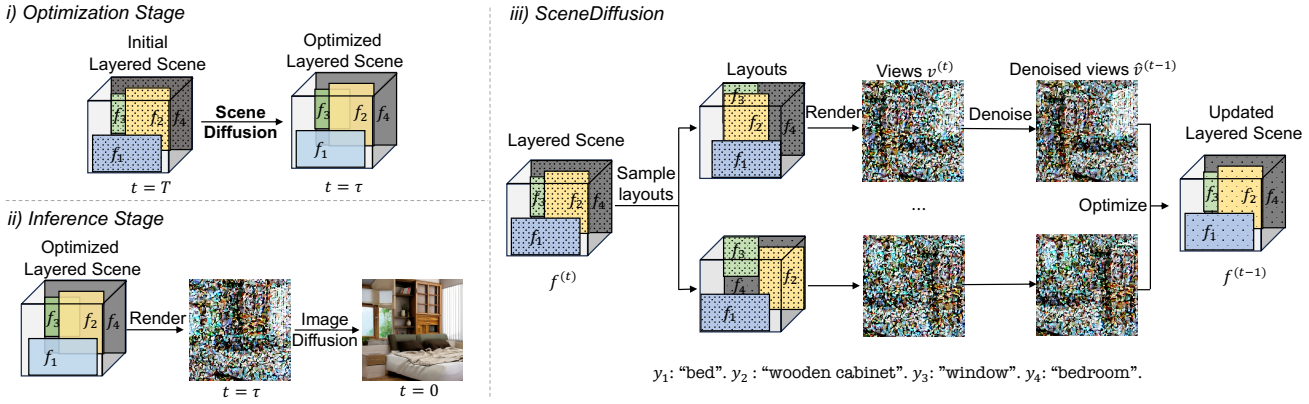

Figure 2. **Method overview.** Our framework has two stages: **i) optimization stage**, we optimize a layered scene representation with SceneDiffusion for $T - \tau$ diffusion steps, and **ii) inference stage**, we render the optimized layered scene with τ-step standard image diffusion. **iii) SceneDiffusion** updates the layered scene by denoising multiple randomly sampled layouts in parallel. In the illustration, the scene has 4 layers. Each layer consists of a feature map f, a mask m (shown as a box), and a text prompt y (shown at the bottom). At denoising step t, we randomly sample N layouts and render them to get different views $v^{(t)}$. We then denoise the views using a pretrained T2I diffusion model for one step to get $\hat{v}^{(t-1)}$, which are used to update the feature maps $f^{(t)} \rightarrow f^{(t-1)}$ in the layered scene. Note that boxes here only serve as a rough geometry of objects (like blobs in Epstein et al. [7]), and can be replaced by more accurate masks.

ture. Compared with concurrent works on spatial image editing with diffusion models using self-guidance [8, 24] and feature tracking [34], our method is different in: *1)* we generate scenes that preserve the content across different spatial editing, *2)* we use an explicit layered representation that gives intuitive and precise control, and *3)* we render a new layout via a short schedule of image diffusion, while guidance-based approaches require a long sampling schedule and feature tracking requires gradient-based optimization for each editing.

3. Our Approach

Framework Overview. An overview of our framework is shown in Figure 2. In Section 3.1, we briefly introduce preliminary works on diffusion models and locally conditioned diffusion. Then, in Section 3.2, we present how we obtain a spatially disentangled layered scene with SceneDiffusion. Finally, in Section 3.3, we discuss how SceneDiffusion enables spatial editing on in-the-wild images.

3.1. Preliminary

Diffusion Models. Diffusion models [13, 35] are a type of generative model that learns to generate data from a random input noise. More specifically, given an image from the data distribution $x_0 \sim p(x_0)$, a *fixed* forward noising process progressively adds random Gaussian noise to the data, hence creating a Markov Chain of random latent variable $x_1, x_2, ..., x_T$ following:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_i}x_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where $\beta_1, ...\beta_T$ are constants corresponding to the noise schedule chosen so that for a high enough number of diffusion steps x_T is assumed to be a standard Gaussian. We then train a denoiser θ that learns the backward process, *i.e.*, how to remove the noise from a noisy input [13]. At inference time, we can sample an image by starting from a random standard Gaussian noise $x_T \sim \mathcal{N}(0; \mathbf{I})$ and iteratively denoise the image following the Markov Chain, i.e., by consecutively sampling x_{t-1} from $p_\theta(x_{t-1}|x_t)$ until x_0:

$$x_{t-1} = \frac{1}{\sqrt{\lambda_t}}\Big(x_t - \frac{1-\lambda_t}{\sqrt{1-\bar{\lambda}_t}}\epsilon_\theta(x_t, t)\Big) + \sigma_t\mathbf{z}, \tag{2}$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, $\bar{\lambda}_t = \prod_{s=1}^{t}\lambda_s$, $\lambda_t = 1 - \beta_t$, and σ_t is the noise scale.

Locally Conditioned Diffusion. Various approaches [1, 28] have been proposed to generate partial image content based on local text prompts using pretrained T2I diffusion models. For K local prompts $\mathbf{y} = \{y_1, y_2, ..., y_K\}$ and binary non-overlapping masks $\mathbf{m} = \{m_1, m_2, ...m_K\}$, locally conditioned diffusion [28] proposes to first predict a full image noise $\epsilon_\theta(x_t, t, y_k)$ for each local prompt y_k with classifier-free guidance [12], and then assign it to its corresponding region masked by m_k:

$$\epsilon_\theta^{\text{LCD}}(x_t, t, \mathbf{y}, \mathbf{m}) = \sum_{k=1}^{K} m_k \odot \epsilon_\theta(x_t, t, y_k), \tag{3}$$

where $\odot$ is element-wise multiplication.

3.2. Controllable Scene Generation

Given a list of ordered object masks and their corresponding text prompts, we would like to generate a scene where object locations can be changed on the spatial dimensions while keeping the image content consistent and high quality. We leverage a pretrained T2I diffusion model θ that generates in the image space (or latent space) $I \in \mathbb{R}^{c\times w\times h}$, where c is the number of channels and w and h the width and height of the image, respectively. To achieve controllable scene generation, we introduce a layered scene representation in Section 3.2.1 for mid-level control and propose a new sampling strategy in Section 3.2.2.

3.2.1 Layered Scene Representation

We decompose a controllable scene into K layers $[l_k]_{k=1}^K$, ordered by the depth of the objects. Each layer l_k has *1)* a fixed object-centric binary mask $m_k \in \{0,1\}^{c\times w\times h}$ (*e.g.*, a bounding box or segmentation mask) to show the geometric property of the object, *2)* a two-element offset, $o_k \in [0;\mu_k]\times[0;\nu_k]$, indicating its spatial locations, with μ_k and ν_k defining the horizontal and vertical movement range, and *3)* a feature map $f_k^{(t)} \in \mathbb{R}^{c\times w\times h}$ representing its visual appearance at diffusion step t.

A scene *layout* is defined by the masks and their associated offsets. The offset o_k of each layer can be sampled from the movement range $[0;\mu_k]\times[0;\nu_k]$ to form a new layout. Specially, we set the last layer l_K as the background so that $m_K = \{1\}^{c\times w\times h}$ and $o_K = [0,0]$. Given a layout, the layered representation can be rendered to an image, and we name the image as a *view*. Similar to prior works in controllable scene generation [7] and video editing [16], we use α-blending to composite all the layers during rendering. More concretely, the view $v^{(t)}$ can be calculated as:

$$\begin{aligned} v^{(t)} &= \sum_{k=1}^{K} \alpha_k \odot \overline{\text{move}}(f_k^{(t)}, o_k), \\ \alpha_k &= \overline{\text{move}}(m_k, o_k) \prod_{j=1}^{k-1} (1 - \overline{\text{move}}(m_j, o_j)). \end{aligned} \tag{4}$$

Each element in $\alpha_k \in \{0,1\}^{w\times h}$ indicates that the visibility of that location in the k-th latent feature map, and the function $\overline{\text{move}}(\cdot, o)$ means that we spatially shift the values of the feature map f or mask m by o. The rendering process can be applied to the layered scene at any diffusion step, resulting in a view with a certain noise level.

For initialization at diffusion step T, the initial feature map $f_k^{(T)}$ is independently sampled from a standard Gaussian noise $\mathcal{N}(0, I)$ for each layer. It can be shown that since α is binary and $\sum_{k=1}^K \alpha_k^2 = 1$, the rendered views from the initial layered scene still follow the standard Gaussian distribution. This allows us to denoise the views directly using pretrained diffusion models. In Section 3.2.2, we discuss how to update $f_k^{(t)}$ in a sequential denoising process.

3.2.2 Generating Scenes with SceneDiffusion

We propose *SceneDiffusion* to optimize the feature maps in the layered scenes from Gaussian noise. Each SceneDiffusion step *1)* renders multiple views from randomly sampled layouts, *2)* estimates the noise from the views, and then *3)* updates the feature maps.

Specifically, SceneDiffusion samples N groups of offset $[o_{1,n}, o_{2,n}, \cdots, o_{K,n}]_{n=1}^N$, with each offset $o_{k,n}$ being an element of the movement range $[0;\mu_k]\times[0;\nu_k]$. This leads to N layout variants. A higher number of layouts helps the denoiser locate a better mode while also increasing the computational cost, as shown in Section 4.2. From the K latent feature maps, we render the layouts as N views $v_n \in \{v_1^{(t)}, ..., v_N^{(t)}\}$:

$$v_n^{(t)} = \sum_{k=1}^{K} \alpha_k \odot \overline{\text{move}}(f_k^{(t)}, o_{k,n}). \tag{5}$$

Then, we stack all views in each SceneDiffusion step and predict the noise $\{\hat{\epsilon}_n^{(t)}\}_{n=1}^N$ using locally conditioned diffusion [28] described in Equation 3:

$$\hat{\epsilon}_n^{(t)} = \epsilon_\theta^{LCD}(v_n^{(t)}, t, \mathbf{m}, \mathbf{y}), \forall n \in \{1, 2, \cdots, N\} \tag{6}$$

where $\mathbf{m}$ are the object masks, and $\mathbf{y}$ are local text prompts for each layer. Since we can run multiple layout denoising in parallel, computing $\{\hat{\epsilon}_n^{(t)}\}_{n=1}^N$ brings little time overhead, while costing an additional memory consumption proportional to N. We then update the views $v_n^{(t)}$ from the estimated noise $\hat{\epsilon}_n^{(t)}$ using Equation 2 to get $\hat{v}_n^{(t-1)}$.

Since each view corresponds to a different layout and is denoised independently, conflict can happen in overlapping mask regions. Therefore, we need to optimize each feature map $f_k^{(t-1)}$ so that the rendered views from Equation 5 is close to denoised views:

$$f^{(t-1)} = \arg\min_{f^{(t-1)}} \sum_{n=1}^{N} ||\hat{v}_n^{(t-1)} - v_n^{(t-1)}||_2^2 \tag{7}$$

This least square problem has the following closed-form solution:

$$\begin{aligned} f_k^{(t-1)} &= \frac{\sum_{n=1}^N \overline{\text{move}}(\alpha_k \odot \hat{v}_n^{(t-1)}, -o_{k,n})}{\sum_{n=1}^N \overline{\text{move}}(\alpha_k, -o_{k,n})}, \\ &\forall k \in \{1, \cdots, K\}, \end{aligned} \tag{8}$$

where $\overline{\text{move}}(x, -o)$ denotes the values in x translated in the reverse direction of o. The derivation for this solution is similar to the discussion in Bar-Tal et al. [1]. The solution essentially sets $f_k^{(t-1)}$ to a weighted average of cropped denoised views.

3.2.3 Neural Rendering with Image Diffusion

We switch to vanilla image diffusion for τ steps after running SceneDiffusion for $T - \tau$ steps. Since the layer masks $\mathbf{m}$ like bounding boxes only serve as a rough mid-level representation instead of an accurate geometry, this image diffusion stage can be viewed as a *neural renderer* that maps mid-level control to the image space [7, 26, 41]. The value of τ trades off the image quality and the faithfulness to the layer mask. A value of τ in 25% to 50% of the total diffusion steps strikes the best balance, which usually costs less than a second using a popular 50-step DDIM scheduler [36]. The global prompt used for the image diffusion stage can be separately set. In this work, we mainly set the global prompt to the concatenation of local prompts in the depth order $y_{\text{global}} =< y_1, y_2, \ldots, y_K >$ and find this simple strategy sufficient in most cases.

3.2.4 Layer Appearance Editing

The appearance of each layer can be edited individually via modifying local prompts. Objects can be restyled or replaced by changing the local prompt to a new one and then performing SceneDiffusion using the same feature map initialization.

3.3. Application to Image Editing

SceneDiffusion can be conditioned on a reference image by using its sampling trajectory as an *anchor*, allowing us to change the layout of an existing image. Concretely, when a reference image is given along with an existing layout, we set the reference image to be the optimization target at the final diffusion step, *i.e.*, an anchor view denoted as $\hat{v}_a^{(0)}$. Then, we add Gaussian noise to this view at different diffusion noise levels, creating a trajectory of anchor views at different denoising steps.

$$\hat{v}_a^{(t)} = \sqrt{1-\beta_t}\hat{v}_a^{(0)} + \beta_t\epsilon,\ \forall t \in [1, \cdots, T], \tag{9}$$

where $\epsilon \sim \mathcal{N}(0, 1)$. In each diffusion step, we use the corresponding anchor view $\hat{v}_a^{(t)}$ to further constraint $f^{(t-1)}$, which leads to an extra weighted term in Equation 7:

$$\begin{aligned} f^{(t-1)} &= \arg\min_{f^{(t-1)}} \sum_n w_n ||\hat{v}_n^{(t-1)} - v_n^{(t-1)}||_2^2 \\ w_n &= \begin{cases} w & \text{if } n = a, \\ 1 & \text{otherwise.} \end{cases} \end{aligned} \tag{10}$$

where $n \in \{1, \cdots, N\} \cup \{a\}$, and w controls the importance of $\hat{v}_a^{(t)}$. A large enough w produces good faithfulness to the reference image, we set $w = 10^4$ in this work. The closed-form solution of this equation is similar to Equation 8 and can be found in supplementary material.

4. Experiments

4.1. Experimental Setup

We evaluate our method both *qualitatively* and *quantitatively*. For quantitative study, a thousand-scale dataset is required to effectively measure metrics like *FID*. However, populating semantically meaningful spatial editing pairs for multi-object scenes is challenging, particularly when inter-object occlusions should be considered. Therefore, we restrict quantitative experiments to single-object scenes. Please refer to qualitative results for multi-object scenes.

Dataset. We curate a dataset of high-quality, subject-centric images associated with image captions and local descriptions. Object masks are also annotated automatically using GroundedSAM [30]. We first generate 20,000 images from 1,000 image captions and then apply a rule-based filter to remove low-quality images, which results in 5,092 images in total. Object masks and local descriptions are then automatically annotated.

Metrics. Our main metrics for controllable scene generation are *Mask IoU*, *Consistency*, *Visual Consistency*, *LPIPS*, and *SSIM*. *Mask IoU* measures the alignment between the target layout and the generated image. Other metrics compare multiple generated views in the same scene and evaluate their similarity: *Consistency* for mask consistency, *Visual Consistency* for foreground appearance consistency, *LPIPS* for perceptual, and *SSIM* for structural changes. Moreover, in the image editing experiment, we report *FID* to measure the similarity of the edited images to the original ones for image quality quantification.

Implementation By default we set $N = 8$ in our experiments. For quantitative studies, all experiments are averaged on 5 random seeds. Please refer to our supplemental document for more information on our dataset construction, metrics selection, standard deviations of experiments and implementation details.

4.2. Controllable Scene Generation

Setting. We randomly place an object mask at different positions to form random target layouts. Images should be generated conditioned on the target layouts and local prompts, and the content is expected to be consistent in different layouts. The object masks are from the aforementioned curated dataset. To reduce the chance that objects move out of the canvas, we restrict the maks position to a square centered at the original position with its side length of 40% of the image width. A visual example can be found in Figure 9.

Figure 3. **Sequential manipulations.** Our generated scenes can be manipulated by operating on layers sequentially.

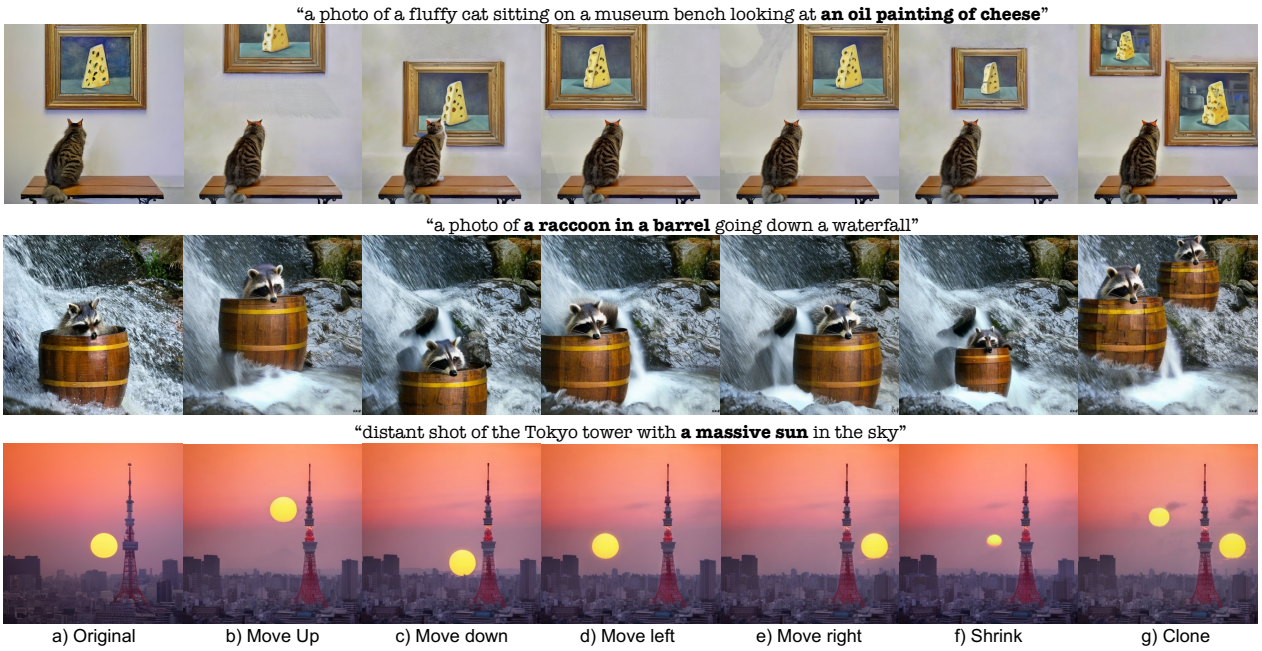

Figure 4. **Object moving.** Our approach can be employed to move objects on a given image. Edited objects are shown in bold in the prompts. Examples are borrowed from Epstein et al. [8] and no access to the initial latent noise is assumed. All layouts for each example are generated from the same scene. As a result, our approach keeps the overall content consistent across different editings, which most prior works fail to achieve. A full comparison with prior works can be found in appendix.

Baselines. We compare our approach to MultiDiffusion [1], which is a training-free approach that generates images conditioned on masks and local descriptions. We use a 20% solid color bootstrapping strategy following their protocol. Foreground and background noise are fixed in the same scene for better consistency.

Results. We present quantitative results in Table 1, which show that SceneDiffusion outperforms MultiDiffusion on all metrics. For qualitative study, we show the results of sequentially manipulation our generated scenes in Figure 3.

Table 1. **Quantitative comparison for controllable scene generation.** †: without the solid color bootstrapping strategy.

Method	M. IoU ↑	Cons.↑	V. Cons.↓	LPIPS ↓	SSIM ↑
MultiDiff. [1]†	0.263	0.257	-	0.521	0.450
MultiDiff. [1]	0.466	0.436	0.236	0.519	0.471
Ours†	0.310	0.609	-	**0.198**	0.761
Ours	**0.522**	**0.721**	**0.112**	0.215	**0.762**

4.3. Object Moving for Image Editing

Setting. Given a reference image, an object mask, and a random target position, the goal is to generate an image where the object has moved to the target position while

Figure 5. **Restyling objects.** Adding style description to the layer prompt restyles the object when fixing the initial noise. The circular arrow shows the restyled object.

Figure 6. **Replacing objects.** Objects can be changed to different objects by modifying their layer prompts without affecting other objects in the scene. The circular arrow shows the replaced object.

Table 2. **Quantitative comparison for object moving.** †: specialized inpainting model trained with masking.

Method	FID ↓	M. IoU ↑	V. Cons. ↓	LPIPS ↓	SSIM ↑
RePaint [20]	10.267	0.620	0.166	0.278	0.671
Inpainting†	6.383	0.747	0.112	0.264	0.680
Ours	**5.289**	**0.817**	**0.075**	**0.263**	**0.709**

keeping the rest of the content similar. The aforementioned range is used to prevent moving the object out of the canvas.

Baselines. We compare with inpainting-based approaches. We first crop the object from the reference image, paste it to the target location, and then inpaint the blank areas. We dilate the edge of objects for 30 pixels to better blend the object with the background. We compare our approach with two inpainting models: a standard T2I diffusion model using the RePaint technique [20], and a specialized inpainting model trained with masking. We set all local layer prompts in our approach to the global image caption for a fair comparison.

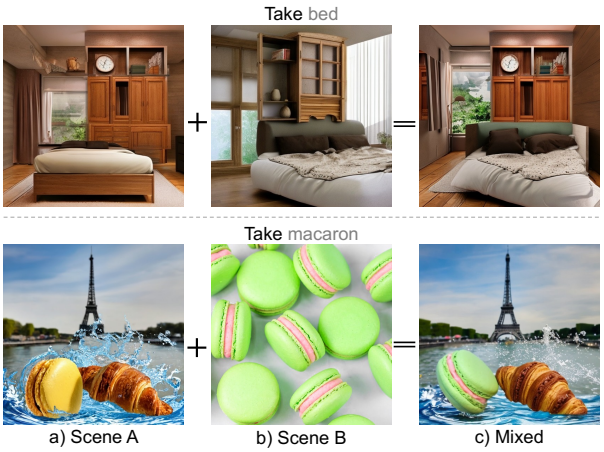

Figure 7. **Mixing scenes.** One may mix scenes by copying a layer from one scene and pasting it in another scene.

Results. We report quantitative results in Table 2. Our approach outperforms both inpainting-based baselines by a clear margin on all metrics. Qualitative results of object moving are shown in Figure 4.

4.4. Layer Appearance Editing

We show the results of object restyling in Figure 5 and object replacement in Figure 6. We observe that changes are mostly isolated to the selected layer, while other layers slightly adapt to make the scene more natural. Furthermore, layer appearance can be transferred across scenes by directly copying a layer from one scene to another, as shown in Figure 7.

Figure 8. **Ablation on τ.** We swap the locations of the two objects. Stopping SceneDiffusion at a later step improves consistency and prevents hallucination.

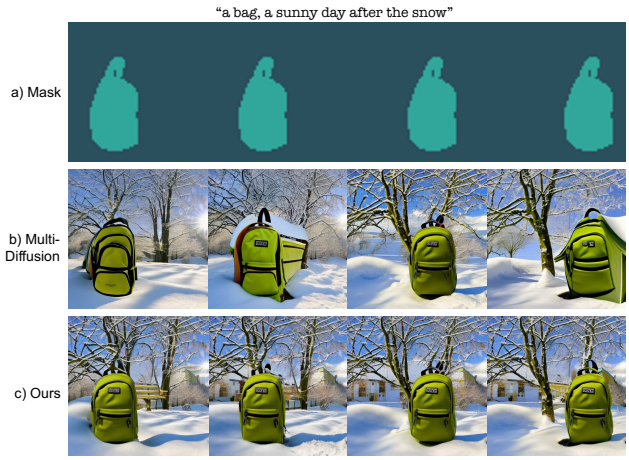

Figure 9. **Qualitative evaluation of controllable scene generation.** Multidiffusion [1] is able to generate a backpack in accordance to the target mask, but both the background and the object change at different layouts. Our method can produce coherent and consistent images with minimal visual appearance difference.

Table 3. **Component analysis.**

Method	**CLIP-a** ↑	**VC** ↓	M. IoU ↑	Cons.↑	LPIPS ↓	SSIM ↑
Ours (N=8, τ=13)	6.12	0.11	0.51	0.72	0.22	0.74
w/o multiple layouts	6.05	0.23	0.46	0.43	0.51	0.47
w/o random sampling	5.98	0.12	0.50	0.68	0.22	0.75
w/o image diffusion	5.96	0.09	0.51	0.72	0.21	0.76

Table 4. **Analysis on N and τ**

N	τ	Optim.↓	Infer.↓	**CLIP-a** ↑	M. IoU ↑	Cons.↑	LPIPS ↓	SSIM ↑
8	13	17.3s	0.82s	6.12	0.514	0.721	0.224	0.749
4	13	9.65s	0.82s	5.99	0.491	0.689	0.225	0.747
2	13	5.73s	0.82s	5.97	0.481	0.672	0.229	0.735
8	25	12.0s	1.53s	6.13	0.502	0.643	0.276	0.685
8	0	22.9s	0.0s	5.96	0.515	0.723	0.211	0.767

4.5. Ablation study

In Table 3, we ablate all components. We additionally measure *CLIP-aesthetic* (CLIP-a) following [1] to quantify the image quality. Without jointly denoising multiple layouts, all metrics drop drastically. With a deterministic sampling of layouts, the image quality degrades. Without the image diffusion stage, although consistency metrics slightly improve, image quality significantly deteriorates. In Table 4, we analyze the effect of the number of views and image diffusion steps. We observe that having more views and more SceneDiffusion steps leads to a better disentanglement between the object and the background, as indicated by higher Mask IoU and Consistency. A qualitative comparison can be found in Figure 8. We also present the accuracy-speed trade-off when limiting to a single 32GB GPU. Larger N increases the optimization time. Larger τ increases the inference time. For all ablation experiments, we use a randomly selected 10% subset for easier implementation.

5. Conclusion

We proposed SceneDiffusion that achieves controllable scene generation using image diffusion models. SceneDiffusion optimizes a layered scene representation during the diffusion sampling process. Thanks to the layered representation, spatial and appearance information are disentangled which allows extensive spatial editing operations. Leveraging the sampling trajectory of a reference image as an anchor, SceneDiffusion can move objects on in-the-wild images. Compared to baselines, our approach achieves better generation quality, cross-layout consistency, and running speed. **Limitations**. The object's appearance may not fit tightly to the mask in the final rendered image. Besides, our approach requires a large amount of memory to simultaneously denoise multiple layouts, restricting the applications in resource-limited user cases. **Acknowledgments**. This study is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2021-08-018), the Ministry of Education, Singapore, under its MOE AcRF Tier 2 (MOET2EP20221-0012), NTU NAP, and under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative.

References

[1] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. In *Proceedings of the 23rd International Conference on Machine Learning*, 2023. 3, 4, 6, 8

[2] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 2

[3] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023. 2

[4] Shoufa Chen, Mengmeng Xu, Jiawei Ren, Yuren Cong, Sen He, Yanping Xie, Animesh Sinha, Ping Luo, Tao Xiang, and Juan-Manuel Perez-Rua. Gentron: Delving deep into diffusion transformers for image and video generation. *arXiv preprint arXiv:2312.04557*, 2023. 2

[5] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. Flatten: optical flow-guided attention for consistent text-to-video editing. *arXiv preprint arXiv:2310.05922*, 2023. 2

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[7] Dave Epstein, Taesung Park, Richard Zhang, Eli Shechtman, and Alexei A Efros. Blobgan: Spatially disentangled scene representations. In *European Conference on Computer Vision*, pages 616–635. Springer, 2022. 2, 3, 4, 5

[8] Dave Epstein, Allan Jabri, Ben Poole, Alexei A Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *arXiv preprint arXiv:2306.00986*, 2023. 2, 3, 6

[9] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022. 2

[10] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2

[11] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3

[14] Fangzhou Hong, Zhaoxi Chen, Yushi LAN, Liang Pan, and Ziwei Liu. EVA3d: Compositional 3d human generation from 2d image collections. In *International Conference on Learning Representations*, 2023. 2

[15] Phillip Isola and Ce Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3048–3055, 2013. 2

[16] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021. 2, 4

[17] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 2

[18] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023. 2

[19] Erika Lu, Forrester Cole, Tali Dekel, Weidi Xie, Andrew Zisserman, David Salesin, William T Freeman, and Michael Rubinstein. Layered neural rendering for retiming people in video. *arXiv preprint arXiv:2009.07833*, 2020. 2

[20] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2, 7

[21] Willi Menapace, Stephane Lathuiliere, Sergey Tulyakov, Aliaksandr Siarohin, and Elisa Ricci. Playable video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10061–10070, 2021. 2

[22] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2

[23] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 2

[24] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. *arXiv preprint arXiv:2307.02421*, 2023. 2, 3

[25] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 2

[26] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 2, 5

[27] Yu-ichi Ohta, Takeo Kanade, and Toshiyuki Sakai. An analysis system for scenes containing objects with substructures.

In *Proceedings of the Fourth International Joint Conference on Pattern Recognitions*, pages 752–754, 1978. 2

[28] Ryan Po and Gordon Wetzstein. Compositional 3d scene generation using locally conditioned diffusion. *arXiv preprint arXiv:2303.12218*, 2023. 3, 4

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

[30] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang 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, 2024. 5

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. arxiv. *arXiv preprint arXiv:2112.10752*, 2021. 2

[32] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 2

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2

[34] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. *arXiv preprint arXiv:2306.14435*, 2023. 2, 3

[35] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2, 3

[36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 5

[37] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *arXiv preprint arXiv:2306.03881*, 2023. 2

[38] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023. 2

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

[40] Jianyuan Wang, Ceyuan Yang, Yinghao Xu, Yujun Shen, Hongdong Li, and Bolei Zhou. Improving gan equilibrium by raising spatial awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11285–11293, 2022. 2

[41] Yinghao Xu, Menglei Chai, Zifan Shi, Sida Peng, Ivan Skorokhodov, Aliaksandr Siarohin, Ceyuan Yang, Yujun Shen, Hsin-Ying Lee, Bolei Zhou, et al. Discoscene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4402–4412, 2023. 2, 5

[42] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *International Journal of Computer Vision*, 129: 1451–1466, 2021. 2

[43] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2