Ctrl-Room: Controllable Text-to-3D Room Meshes Generation with Layout Constraints

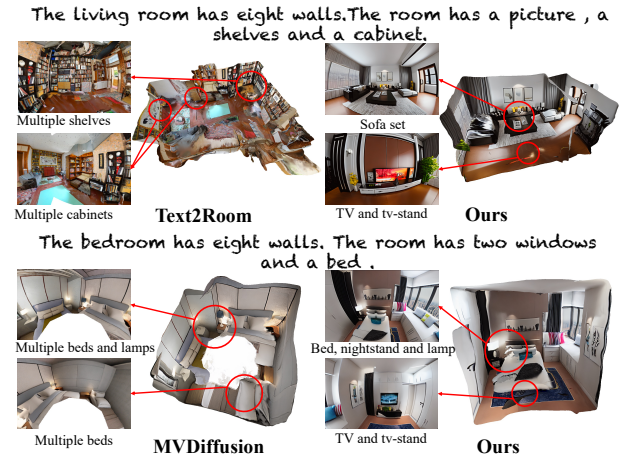
Chuan Fang[1,2*], Yuan Dong[3*], Kunming Luo[1,2], Xiaotao Hu[1,2], Rakesh Shrestha[4], Ping Tan[1,2†]
[1] Hong Kong University of Science and Technology [2] LightIllusion, China.
[3] Alibaba Group [4]Simon Fraser University, Canada.
[1]cfangac@connect.ust.hk, [2]dy283090@alibaba-inc.com, [3]pingtan@ust.hk

Abstract

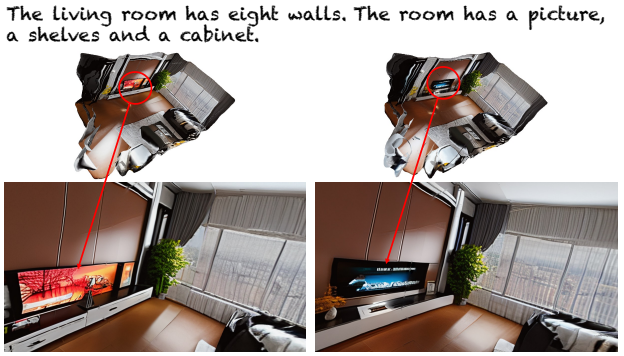
Text-driven 3D indoor scene generation is useful for gaming, film industry, and AR/VR applications. However, existing methods cannot faithfully capture the scene layout based on text descriptions, nor do they allow flexible editing of individual objects in the room. To address these problems, we present Ctrl-Room, which can generate convincing 3D rooms with designer-style layouts and high-fidelity textures from just a text prompt. Our key insight is to separate the modeling of layouts and appearance. Our proposed method consists of two stages: a Layout Generation Stage and an Appearance Generation Stage. The Layout Generation Stage trains a text-conditional diffusion model to learn the layout distribution with our holistic scene code parameterization. Next, the Appearance Generation Stage employs a fine-tuned ControlNet to produce a vivid panoramic image of the room guided by the 3D scene layout, then further upgrades to a panoramic NeRF model. Benefiting from the scene code parameterization, we can easily edit the generated room model through our mask-guided editing module, without expensive edit-specific training. Extensive experiments on the Structured3D dataset demonstrate that our method outperforms existing methods in producing more reasonable, view-consistent, and editable 3D rooms from text prompts.

1. Introduction

High-quality 3D indoor scenes play a crucial role across a wide array of applications, ranging from interior design and video games to simulators for embodied AI. Traditionally, indoor scenes are crafted manually by professional artists, which is both time-consuming and costly. Recent advancements in generative models [5, 18, 21, 27] have attempted to simplify the creation of 3D models from textual descriptions, However, extending this capability to text-driven 3D indoor scene generation presents unique challenges as they exhibit strong semantic layout constraints, such as neighboring walls are perpendicular and the TV set often faces a sofa, that are more complicated than objects.



(a) Comparison with Text2Room [12] and MVDiffusion [33].



Replace the TV and TV stand

(b) Flexible editing by instruction or mouse clicks.

Figure 1. We present *Ctrl-Room* to achieve fine-grained textured 3D indoor room generation and editing. (a) compared with the Text2Room [12] and MVDiffusion[33], Ctrl-Room can generate rooms with more plausible 3D structures. (b) Ctrl-Room supports flexible editing. Users can replace furniture items or change their positions easily.

Existing text-driven 3D indoor scene generation approaches, such as Text2-Room [12] and Text2NeRF [42], are designed with an incremental framework. They create 3D indoor scenes by incrementally generating different

viewpoints frame-by-frame and reconstructing the 3D mesh of the room from these sub-view images. However, these approaches often fail to model the global layout of the room, resulting in unconvincing results. As shown in the first row of Fig. 1 (a), the result of Tex2Room exhibits repetitive objects, e.g. several cabinets in a living room, and does not follow the furniture layout patterns. We refer to this problem as the *'Penrose Triangle problem'*, where a generated scene has plausible 3D structures everywhere locally but lacks global consistency. Furthermore, prior approaches do not offer user-friendly interaction, as the resulting 3D geometry and textures are not editable. Other method [16, 17, 28, 33] represent the scene as a panorama image and generate it from a text prompt. However, these works cannot guarantee reasonable scene layouts. As shown on the middle row of Fig. 1 (a), a bedroom generated by MVDiffusion [33] contains multiple beds, which violates room layout priors.

To address these shortcomings, we propose a novel two-stage method to generate a high-fidelity and editable 3D room. The key insight is to separate the generation of 3D geometric layouts from that of visual appearance, which allows us to better capture the room layout and achieve vivid textures at the same time. In the first stage, from text input, our method creates plausible scene layouts with various furniture types and positions. Unlike previous scene synthesis methods [20, 31] that only focus on the furniture arrangement, our approach further considers walls with doors and windows, which play an essential role in the layout. To achieve this goal, we parameterize the room by a holistic scene code, which represents a room as a set of objects. Each object is represented by a vector capturing its position, size, semantic class, and orientation. Based on our compact parameterization, we design a diffusion model to learn the 3D room layout distribution from the Structured3D dataset [44].

Our method then generates the room appearance with the guidance of the 3D room layout. We first generate a panorama using a text-to-image latent diffusion model, then iteratively upgrade the generated images to a NeRF model and generate additional novelty view panorama images. During the panorama generation, unlike previous text-to-panorama works [6, 33], our method explicitly enforces scene layout constraints and guarantees plausible 3D room structures and furniture arrangement. To achieve this goal, we convert the 3D layout synthesized in the first stage into a semantic segmentation map and feed it to a fine-tuned ControlNet [43] model to create the panorama image. We also use this layout information to estimate scene depth and inpaint missing regions at novel viewpoints.

Benefiting from the separation of layout and appearance, our method enables flexible editing on the generated 3D room. The user can replace or modify the size and position of furniture items, e.g. replacing the TV and TV stand as in Fig. 1 (b). Our method can update the room according to the edited room layout through our mask-guided editing module without expensive edit-specific training. The updated room appearance maintains consistency with the original version while satisfying the user's edits.

The main contributions of this paper are summarized as:

- To address the Penrose Triangle Problem, we design a two-stage method for 3D room generation from pure text input, which separates the geometric layout generation and appearance generation. In this way, our method can better capture the scene layout constraints in real-world data and produce a vivid appearance simultaneously.
- Within the separated layout and appearance generation, we introduce novel techniques, including holistic scene code parametrization, layout-guided panorama generation, layout-guided panoramic NeRF, and a mask-guided editing module to achieve high-quality and flexible 3D room generation.
- Qualitative and quantitative experiments confirm that our method excels in producing more realistic and editable 3D rooms compared to existing approaches.

2. Related Work

2.1. Text-based 3D Object Generation

Early methods employ 3D datasets to train generative models. Text2Shape [4] learns a feature representation from paired text and 3D data and uses GAN to generate 3D shapes from the text. Point-E [19] and Shap-E [13] enlarge the scope of the training dataset and employ a latent diffusion model [23] for object generation. However, 3D datasets are scarce, which makes these methods difficult to scale. More recent methods [5, 18, 19, 21, 36, 38] exploit the powerful 2D text-to-image diffusion models [23, 24] for 3D model generation. Typically, these methods generate one or multiple 2D images in an incremental fashion and optimize the 3D model accordingly. DreamFusion [21] introduces a loss based on probability density distillation and optimizes a randomly initialized 3D model through gradient descent. Magic3D [18] uses a coarse model to represent 3D content and accelerates it using a sparse 3D hash grid structure. To alleviate over-saturation and low-diversity problems, ProlificDreamer [38] models and optimizes the 3D parameters through variational score distillation. However, these methods are limited to 3D object generation and cannot be directly extended to 3D scene generation which has additional layout constraints.

2.2. Text-based 3D Room Generation

Room Layout Synthesis Layout generation has been greatly boosted by transformer-based methods. LayoutTransformer [10] employs self-attention to capture relationships between elements to accomplish layout completion.

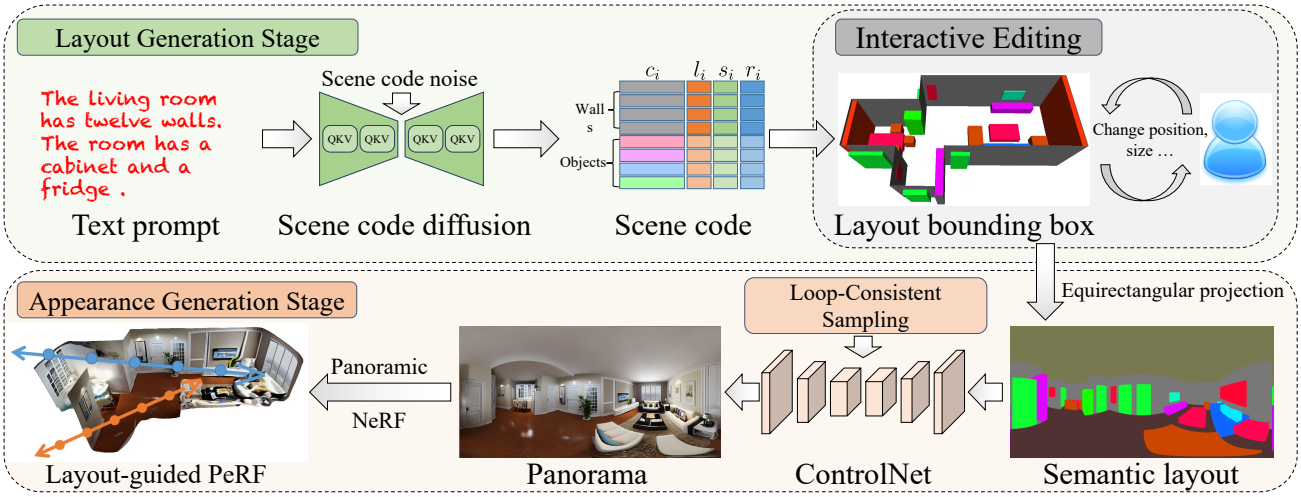


Figure 2. Framework overview. In Layout Generation Stage, we synthesize a scene code from the text input and convert it to a 3D bounding box representation to facilitate editing. In Appearance Generation Stage, we project the bounding boxes into a semantic segmentation map to guide the panorama synthesis. The panorama is then reconstructed into a panoramic NeRF (PeRF)[35]model with layout guidance.

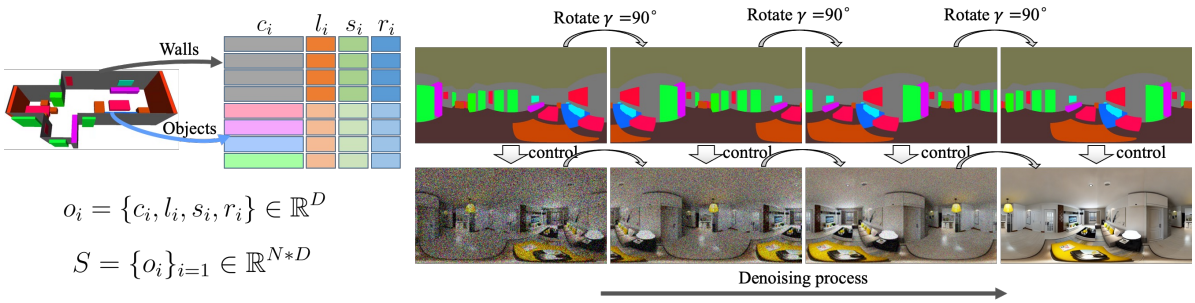
ATISS [20] proposes an autoregressive transformer to generate proper indoor scenes with only the room type and floor plan as the input. DiffuScene [31] and InstructScene [15] model a union of furniture as a fully connected scene graph and propose a diffusion model to sample physically plausible scenes. While these methods generate reasonable furniture layouts, they do not consider the walls, doors, and windows which are crucial in the furniture arrangement. Thus they do not always generate realistic indoor environments.

Panoramic Image Generation Another line of work [16, 17, 28] represent an indoor scene by a panorama image without modeling 3D shapes. These methods enjoy the benefits of abundant training data and produce vivid results. COCO-GAN [16] produces a set of patches and assemble them into a panoramic image. InfinityGAN [17] uses the information of two patches to generate the parts between them, to finally obtain a panoramic image. [28] proposes a 360-aware layout generator to produce furniture arrangements and uses this layout to synthesize a panoramic image based on the input scene background. MVDiffusion [33] simultaneously generates multi-view perspective images and proposes a correspondence-aware attention block to maintain multi-view consistency, and then transfers these images to a panorama. These methods might suffer from incorrect room layout since they do not enforce layout constraints. Furthermore, the results of these methods cannot be easily edited, e.g. resizing or moving furniture around, because they do not maintain an object-level representation.

3D Room Generation GAUDI [2] generates immersive 3D indoor scenes rendered from a moving camera. It disentangles the 3D representation and camera poses to ensure the consistency of the scene during camera movement. CC3D [1] proposes a 3D-aware GAN for multi-object scenes conditioned on a single semantic layout image and is trained using posed multi-view RGB images. Another related line of work [26, 29, 40] deals with re-texturizing a given 3D scene. They employ 2D diffusion models to stylize and further improve the given geometry. Text2Room [12] incrementally synthesizes nearby images with a 2D diffusion model and recovers its depth maps to assemble into a 3D room mesh. Unfortunately, it cannot handle the geometric and textural consistency among the images, resulting in the *'Penrose Triangle problem'*. In our method, we take both geometry and appearance into consideration and create a more geometrically plausible 3D room. A concurrent work [26] also guides the 3D room mesh generation by leveraging the user-input scene layouts. In contrast, our method is capable of synthesizing professional designer-style layouts solely from text prompts.

3. Method

In order to achieve text-based 3D indoor scene generation, we propose **Ctrl-Room**. We first generate the room layout from an input text and then generate the room appearance represented by panoramic images according to the layout, followed by layout-guided panoramic NeRF [35] to generate the final 3D room. This mechanism solves the *Penrose Triangle Problem* to generate physically plausible 3D rooms, while also enabling users to edit the scene layout interactively. The overall framework of our method is depicted in Fig. 2, which consists of two stages: the Layout Generation Stage and the Appearance Generation Stage. In the Layout Generation Stage, we parameterize the indoor scene with a holistic scene code and design a diffusion model to learn its distribution. Once the holistic scene code is generated from text, we recover the room as a set of orientated bounding boxes of walls and objects. Note that users can edit these bounding boxes by adjusting their semantic



(a) Scene parametrization (b) Loop-consistent sampling

Figure 3. (a) A 3D scene S is represented by its scene code $x_0 = \{o_i\}_{i=1}^N$, where each wall or furniture item o_i is a row vector storing attributes like class label c_i, location l_i, size s_i, orientation r_i. (b) During the denoising process, we rotate both the input semantic layout panorama and the denoised image for γ degree at each step. Here we take $\gamma = 90^\circ$ for example.

types, positions, or scales, enabling the customization of 3D room generations. In the Appearance Generation Stage, we obtain an RGB panorama through a conditioned image diffusion model to represent the room texture. Specifically, we project the generated layout bounding boxes into a semantic segmentation layout. We then fine-tune a pre-trained ControlNet [43] model to generate an RGB panorama from the input semantic layout. To ensure loop consistency, we propose a loop-consistent sampling during the inference process. Finally, we integrate the layout and the panorama, then generate a full 3D room through the layout-guided panoramic NeRF module [35]. This module progressively inpaints panoramas at new viewpoints using the fine-tuned ControlNet. To extract meshes from reconstructed NeRF, we render depth maps of the new views and utilize truncated signed distance fusion (TSDF) to obtain the final mesh.

3.1. Layout Generation Stage

Scene Code Definition. Different from previous methods [20, 31], we consider not only furniture but also walls, doors, and windows to define the room layout. We employ a unified encoding of various objects. Specifically, given a 3D scene $\mathcal{S}$ with m walls and n furniture items, we represent the scene layout as a holistic scene code $\mathbf{x_0} = \{\mathbf{o_i}\}_{i=1}^N$, where $N = m + n$. We encode each object o_j as a node with attributes including center location $l_i \in \mathbb{R}^3$, size $s_i \in \mathbb{R}^3$, orientation $r_i \in \mathbb{R}$, class label $c_i \in \mathbb{R}^C$. The concatenation of these attributes characterizes each node as $\mathbf{o}_i = [c_i, l_i, s_i, r_i]$. As can be seen in Fig. 3 (a), we represent a scene layout as a tensor $\mathbf{x_0} \in \mathbb{R}^{N\times D}$, where D is the attribute dimension of a node. In all the data, we choose the normal direction of the largest wall as the 'main direction'. For other objects, we take the angles between their front directions and the main direction as their rotations. We use the one-hot encoding to represent their semantic types.

Scene Code Diffusion. With the scene code definition, we build a diffusion model to learn its distribution. A scene layout is a point in $\mathbb{R}^{N\times D}$. The forward diffusion process is a discrete-time Markov chain in $\mathbb{R}^{N\times D}$. Given a clean scene code $\mathbf{x}_0$, the diffusion process gradually adds Gaussian noise to $\mathbf{x}_0$, until the resulting distribution is Gaussian, according to a pre-defined, linearly increased noise schedule $\beta_1, ..., \beta_T$:

$$q(\mathbf{x_t}|\mathbf{x_0}) := \mathcal{N}(\mathbf{x_t}; \sqrt{\bar{\alpha}_t}\mathbf{x_0}, (1 - \sqrt{\bar{\alpha}_t})\mathbf{I}) \quad (1)$$

where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{r=1}^t \alpha_r$ define the noise level and decrease over the timestep t.

The denoising network is trained to reverse the above process by minimizing the training objectives which includes the denoising objective $\mathcal{L}_{\text{denoise}}$ and a regularization term $\mathcal{L}_{\text{physical}}$ to penalize the penetration among objects and walls as follows,

$$\mathcal{L} = \mathcal{L}_{\text{denoise}} + \mathcal{L}_{\text{physical}}, \quad (2)$$

$$\mathcal{L}_{\text{denoise}} = \mathbf{E}_{\mathbf{x_0},t,y,\epsilon}\|\epsilon - \epsilon_\theta(x_t, t, y)\|^2, \quad (3)$$

$$\mathcal{L}_{\text{physical}} = \sum_{t=1}^{T} w_t * (\mathcal{L}_{\text{w-o}} + \mathcal{L}_{\text{o-o}}). \quad (4)$$

where ϵ_θ is the noise estimator which aims to find the noise ϵ added into the input x_0. Here, y is the text embedding of the input text prompts. The hyperparamter w_t is set to $\bar{\alpha}_\mathbf{t} * 0.1$. $\mathcal{L}_{\text{w-o}}$ is the physical violation loss between walls and objects. We adopt the 3D IoU loss $\mathcal{L}_{\text{o-o}}$ in DiffuScene to avoid intersection between furniture.

The denoising network ϵ_θ takes the scene code $\mathbf{x_t}$, text prompt y, and timestep t as input, and denoises them iteratively to get a clean scene code $\hat{\mathbf{x}}_0$. Please refer to appendix Sec.1 for the details of our $\mathcal{L}_{\text{w-o}}$ and denoising network.

3.2. Appearance Generation Stage

Given an indoor scene layout, we seek to generate the 3D textured room model. We achieve this goal by generating panoramic images and reconstructing a panoramic NeRF (PeRF) model from these panoramas. During the panorama generation, instead of incrementally generating multi-view images like [12], we generate the entire panorama at once. We utilize ControlNet [43] to generate a high-fidelity panorama conditioned by the 3D scene layout.

3.2.1 Layout-guided Panorama Generation

Fine-tuning ControlNet. ControlNet controls the image generation of Stable Diffusion [23] model by an ex-

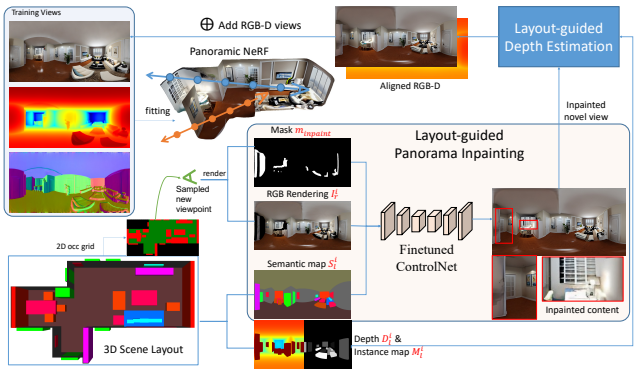


Figure 4. The Layout-guided PeRF takes the input panorama, aligned depth map, and normal map as initialization. Then, a progressive inpainting module is introduced to generate consistent panoramic images at sampled novel views. The progressive inpainting module consists of the layout-guided panorama inpainting and the layout-guided depth estimation module. The final RGB-D panoramic pairs are included as training views to finetune PeRF [35].

tra 2D input. To condition ControlNet on the scene layout, we convert the bounding box representation into a 2D semantic layout panorama through equirectangular projection. In this way, we get a pair of RGB and semantic layout panoramic images for each scene. However, the pretrained ControlNet-Segmentation [9] is designed for perspective images, and cannot be directly applied to panoramas. Thus, we fine-tune it with our pairwise RGB-Semantic layout panoramas on the Structured3D [44]. As the volume of Structured3D is limited, we apply several augmentation techniques for the training data, including standard left-right flipping, horizontal rotation, and Pano-Stretch [30].

Loop-consistent Sampling. A panorama should be loop-consistent. In other words, its left and right should be seamlessly connected. Although the horizontal rotation in data augmentation may improve the model's implicit understanding of the expected loop consistency, it lacks explicit constraints and might still produce inconsistent results. Therefore, we propose an explicit loop-consistent sampling mechanism in the denoising process of the latent diffusion model. As shown in Fig. 3 (b), we rotate both the input layout panorama and the denoised image by γ degree in the sampling process, which applies explicit constraints for the loop consistency during denoising. A concurrent work [39] also uses a similar method for panoramic outpainting. More qualitative results in supplementary Fig.8 and Fig.9 verify that our simple loop-consistent sampling method achieves good results without introducing additional learnable parameters.

3.2.2 Layout-guided PeRF Generation

Since a single panorama is only a partial observation of a scene up to occlusions, lifting a single view into a 3D room becomes complex. Fortunately, our generated layout provides valuable geometric and semantic information to lift the 2D panorama into a 3D model. We propose the layout-guided PeRF, which upgrades the generated panorama aforementioned to a 3D panoramic NeRF [35], enabling multi-view consistent panorama generations guided by the scene layout. Specifically, we start with the layout-guided depth estimation, which recovers the depth map using method [41] and then aligns it to the 3D scene layout leveraging its geometric information. This step corrects the biased depth prediction in the background (wall, ceiling, floor) and preserves objects' shape in the foreground.

Then, we fit our layout-guided PeRF as illustrated in Fig. 4. Specifically, we initialize the scene NeRF with the panorama I^0, the aligned depth map D^*, and the normal map N^*. We sample new viewpoints in the occupancy grid that do not conflict with the initial furniture arrangement. At the i-th novel view, we render semantic map S_l^i, depth map D_l^i, and instance map M_l^i from the scene layout, these are then combined with the panoramic rendering I_r^i and inpainting mask $\mathbf{m}_{\text{inpaint}}$ obtained from the NeRF and fed to the layout-guided panorama inpainting module to generate the novel view panorama. Using our fine-tuned ControlNet, it achieves training-free panoramic inpainting, which replaces pixels outside the inpainting mask $\mathbf{m}_{\text{inpaint}}$ with I_r^i and fill $\mathbf{m}_{\text{inpaint}}$ based on the semantic map S_l^i. Subsequently, after generating the novel view image, we apply the layout-guided depth estimation and include it as training views for PeRF following their framework [35]. More details and results can be found in the appendix Sec.2.

3.3. Mask-guided Editing

A user can modify the generated 3D room by changing the position, semantic class, and size of object bounding boxes. The editing should achieve two goals, i.e. altering the content according to the user's input, and maintaining appearance consistency of the scene objects. We propose a mask-guided image editing, including inpainting step and optimization step as illustrated Fig.6 in supplementary file. The inpainting step fills in the modified area while preserving the rest of the panoramic image. The optimization step focuses on keeping the furniture's appearance unchanged before and after movement and scaling operations.

We explain our method by taking the example in Fig.6 in supplementary file, where a chair's position is moved. We denote the semantic panorama from the edited scene as S_{edited}, then we derive the guidance masks based on its difference from the original one S_{ori}. The source mask $\mathbf{m}_{\text{src}}$ shows the position of the original chair, and the target mask $\mathbf{m}_{\text{tar}}$ indicates the location of the moved chair, and the inpainting mask $\mathbf{m}_{\text{inpaint}} = \{m | m \in \mathbf{m}_{\text{src}} \text{ and } m \notin \mathbf{m}_{\text{tar}}\}$ is the unoccluded region. We use $\mathbf{x}_{\mathbf{0}}^{\text{ori}}$ to denote the original image. During the inpainting step, we replace pixels outside the inpainting mask $\mathbf{m}_{\text{inpaint}}$ with $\mathbf{x}_{\mathbf{t}}^{\text{ori}}$ and store $\mathbf{m}_{\text{inpaint}}$ based on the edited semantic panorama S_{edited}.

Figure 5. Qualitative comparison with previous works. For each method, we show a textured 3D mesh in the first row and two rendered images in the second row.

This straightforward approach ensures that the region outside the mask remains unchanged and the area inside the mask is accurately inpainted. In the optimization step, drawing inspiration from DIFT [32], which has shown that learned features from the diffusion network enable strong semantic correspondence, we ensure consistency between the original and moved furniture by requiring their latent features to be consistent. For more details of the Inpainting and Optimization Step, please refer to our supplementary file Sec.3.

4. Experiments

We evaluate Ctrl-Room on three tasks: layout generation, panorama generation, and 3D Room generation. For those panorama generation methods [6, 33], we recover its depth map using method [41] to reconstruct a textured mesh through Possion reconstruction [14] and MVS-texture [34]. We first describe the experimental settings and then validate our method by comparing it with previous methods quantitatively and qualitatively. We further show various scene editing results to demonstrate the flexible control of our method.

4.1. Experiment Setup

Dataset: We train and evaluate our method on the 3D indoor scene dataset Structured3D [44], which consists of 3,500 houses with 21,773 rooms designed by professional artists. A single panoramic image and 3D scene layout are provided in each room. We parse the scene layout using oriented bounding boxes for common indoor room types like the bedroom, kitchen, living room, study, and bathroom. Then, we follow [37] to generate text prompts describing the scene layout. The filtered dataset for training and evaluation consists of 4,961 bedrooms, 1,848 kitchens, 3,039 living rooms, 698 studies, and 1500 bathrooms. For each room type, we use 80% of rooms for training and the remaining for testing. Following DiffuScene [31], we further qualitatively evaluate our layout generation on 3D-FRONT dataset [7].

Metrics: Follow previous work [20, 31], Frechet Inception Distance (FID) [11] and Kernel inception distance (KID) [3] are used to measure the plausibility and diversity of 1,000 synthesized scene layouts. We choose FID, CLIP Score (CS) [22], and Inception Score (IS) [25] to measure the image quality of generated panoramas. To compare the quality of 3D room models, we follow Text2Room [12] to render images of the 3D room model and measure the CLIP Score (CS) and Inception Score (IS). We also conduct a user study and ask 61 users to score Perceptual Quality (PQ) and 3D Structure Completeness (3DS) of the final room mesh on scores ranging from 1 to 5.

More details about data preprocessing, experimental settings, and baseline implementations can be found in supplementary file Sec.4 and Sec.5.

4.2. Comparison with Previous Methods

4.2.1 Qualitative Comparison

Fig. 5 shows some results generated by different methods. The first row shows a textured 3D room model, and the second row shows some perspective renderings from the room model. As we can see, Text2Light [6] fails to ensure the loop consistency of the generated panorama, which leads to distorted geometry and unreasonable room model. Both MVDiffusion [33] and Text2Room [12] can generate vivid local images as demonstrated by the perspective renderings in the second row. But they fail to capture the global room layout. These two methods often repeat a dominating object, e.g. the cabinet in the bedroom appears multiple times at different places and violate the room layout constraint. In comparison, our method does not suffer from these problems and generates high-quality results. More examples are provided in the Fig.12 in supplementary file.

4.2.2 Layout Generation

Fig. 6 verifies that our layout generation results are plausible and can offer reliable 3D scene layout constraints for the following appearance generation stage. As shown in Fig. 6, our text-conditioned layout generation module can synthesize natural and diverse typical indoor scenes. The size and

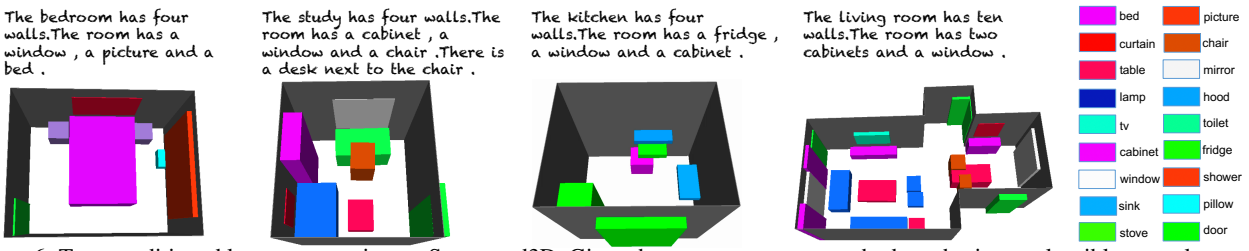


Figure 6. Text-conditioned layout generation on Structured3D. Given the text prompt, our method synthesizes a plausible scene layout that matches the description. The generated layout is represented using different colors to indicate various object categories, such as blue for the sofa and brown for the chair. More results and semantic labels are provided in Fig.10 in supplementary file.

Table 1. Quantitative Comparison of layout generation on 3D-FRONT. Note that DiffuScene-w-SC uses an additional network to learn a Shape Code for each furniture, facilitating the evaluation process to retrieve a more accurate CAD model for each furniture. Nevertheless, our method outperforms others in the common settings, where only the generated semantic class and size are used for retrieval.

Method	Retrieval from	Livingroom			Diningroom		
		FID ↓	KID ↓	SCA	FID ↓	KID ↓	SCA
DiffuScene-w-SC [31]	Shape Code	**35.27**	**0.64**	**54.69**	**32.87**	**0.57**	**51.67**
ATISS [20]	Semantic Bounding Box	40.45	4.57	63.48	36.61	1.90	55.44
DiffuScene-wo-SC [31]	Semantic Bounding Box	38.55	**1.33**	63.54	36.47	1.8	57.04
Ours	Semantic Bounding Box	**36.0**	1.4	**56.42**	**34.78**	**1.3**	**54.37**

spatial location of the furniture are reasonable, and the relative positions between the furniture pieces are accurately recovered. Additional objects not described in the text are automatically generated according to the scene prior.

Table 1 provides a quantitative evaluation against state-of-the-art scene synthesis methods including ATISS [20] and DiffuScene [31] on the 3D-FRONT. Following these methods, we rendered the generated scenes into 256×256 top-down orthographic images to compute the FID, KID, and Scene Classification Accuracy (SCA) scores. To facilitate this computation, ATISS, DiffuScene-wo-SC(without shape code), and our method retrieve the most similar CAD model in the 3D-FUTURE [8] for each object based on generated semantic class and sizes. DiffuScene-w-SC uses an additional network to learn a shape code for each furniture to choose a better 3D mesh model. Note that the SCA score is better when it is closer to 50%. We have excluded walls, doors, and windows from our scene code representation to ensure a fair comparison. Table 1 shows our method achieves results superior to that of ATISS and DiffuScene-wo-SC, indicating that our approach is capable of producing more realistic and natural layouts of indoor scenes.

4.2.3 Panorama Generation

Fig. 7 qualitatively evaluates our generated panoramic images, the image is visualized in a panoramic image viewer to facilitate the user to check the global content. The left side of each column is two zoom-in views, and the right side is the fisheye view. Text2Light [6] suffers from serious inconsistency on the borders of the generated panorama. It also shows a lot of unexpected objects in the image. MVDiffusion [33] suffers from repetitive furniture and fails to synthesize reasonable content for the target room type. In contrast, our method obtains a plausible layout and vivid panorama from the given text prompt.

Table 2 provides quantitative evaluations. We follow MVDiffusion [33] to crop perspective images from the generated panoramas on the test split and evaluate the FID, CS, and IS scores on the cropped multi-view images. In the left part of Table 2, our method achieves the best score in FID, which indicates that our method can better capture the room appearance because of its faithful recovery of the room layout. However, our score on CS is slightly lower than MVDiffusion, which seems insensitive to the number of objects and cannot reflect the room layout quality. The IS score depends on the semantic diversity of the cropped images as captured by an image classifier. Text2Light has the best IS score, since the generations contain unexpected objects.

In Fig.8 of the supplementary file, we also study the performance of our panorama generation module with and without loop-consistent sampling mechanism, the ablation indicates the loop-consistent sampling helps the generated panorama obtain better texture consistency.

4.2.4 3D Room Generation

We then compare the 3D room models in terms of their rendered images. Because of the expensive running time of Text2Room [12], we only test on 12 examples for this comparison. In this comparison, we further skip Text2light and MVDiffusion since we have compared them on panoramas. As the room layout is better captured with a large FOV, we render 60 perspective images of each scene with a 140° FOV and evaluate their CS and IS scores respectively. The results of this comparison are shown in the middle of Table 2. Our method obtains better scores on both metrics than Text2Room.

We further evaluate the quality of the textured 3D mesh model by user studies. The results of the user study are

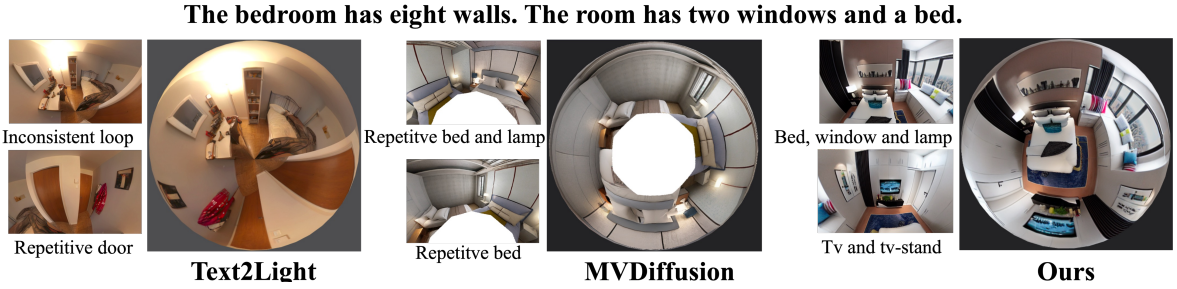
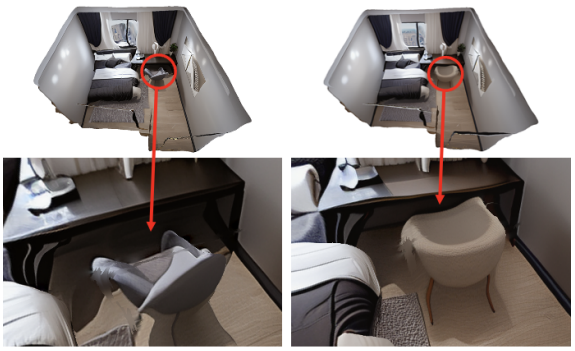


Figure 7. Qualitative comparison for panorama generation. More results are available in the Appendix.

Table 2. Quantitative Comparison of panorama and mesh generation.

Method	Panorama Metrics			2D Rendering Metrics		3D Mesh User Study	
	FID ↓	CS ↑	IS ↑	CS ↑	IS ↑	PQ↑	3DS ↑
Text2Light [6]	56.22	21.45	**4.198**	-	-	2.732	2.747
MVDiffusion [33]	34.76	**23.93**	3.21	-	-	3.27	3.437
Text2Room [12]	-	-	-	25.90	2.90	2.487	2.588
Ours	**21.02**	22.19	3.56	**25.97**	**3.14**	**3.89**	**3.746**

(a) Resize the TV



(b) Replace the chair by a new one

Figure 8. Editing examples. (a) resize the TV, (b) replace the chair with a new one.

shown on the right of Table 2. Users prefer our method over others, for its clear room layout structure and furniture arrangement.

4.3. Interactive Scene Editing

We demonstrate the scene editing capability of our method in Fig. 8. In this case, we resize the TV and replace the chair in the generated results. Fig. 1 (b) shows examples of replacing the TV and TV stand. Our method can keep the visual appearance of the moved/resized objects unchanged after editing. More examples can be found in the appendix.

5. Conclusion

We present **Ctrl-Room**, a flexible method to achieve structurally plausible and editable 3D indoor scene generation. It consists of two stages, the layout generation stage and the appearance generation stage. In the layout generation stage, we design a scene code to parameterize the scene layout and learn a text-conditioned diffusion model for text-driven layout generation. In the appearance generation stage, we fine-tune a ControlNet model to generate a vivid panorama image of the room with the guidance of the layout. Finally, a high-quality 3D room with a structurally plausible layout and realistic textures can be generated via the layout-guided panoramic NeRF. We conduct extensive experiments to demonstrate that **Ctrl-Room** outperforms existing methods for 3D indoor scene generation both qualitatively and quantitatively, and supports interactive 3D scene editing.

6. Limitation

There are still some limitations of Ctrl-Room. Firstly, we only support single-room generation, thus we cannot produce large-scale indoor scenes with multiple rooms. A promising direction is to learn a text-driven diffusion model to produce more consistent RGB-D panorama images cross multiple rooms under the scene layout constraints. Secondly, as we explore injecting 3D scene information into pretrained 2D models, thus we rely on 3D labeled scene dataset to drive the learning and fine-tuning process. Leveraging scene datasets with only 2D labels to learn 3D priors is also a promising direction. Thirdly, the generated 3D model still contains artifacts and incomplete structures in invisible areas because of the occlusion and poor performance of the panoramic depth estimator. We leave the aforementioned limitations for our future efforts.

References

[1] Sherwin Bahmani, Jeong Joon Park, Despoina Paschalidou, Xingguang Yan, Gordon Wetzstein, Leonidas Guibas, and Andrea Tagliasacchi. Cc3d: Layout-conditioned generation of compositional 3d scenes. *arXiv preprint arXiv:2303.12074*, 2023. 3

[2] Miguel Angel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, et al. Gaudi: A neural architect for immersive 3d scene generation. *NeurIPS*, 35:25102–25116, 2022. 3

[3] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2021. 6

[4] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *ACCV*, pages 100–116. Springer, 2019. 2

[5] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023. 1, 2

[6] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Text2light: Zero-shot text-driven hdr panorama generation. *ACM TOG*, 41(6):1–16, 2022. 2, 6, 7, 8

[7] Huan Fu, Bowen Cai, Lin Gao, Lingxiao Zhang, Jiaming Wang Cao Li, Zengqi Xun, Chengyue Sun, Rongfei Jia, Binqiang Zhao, and Hao Zhang. 3d-front: 3d furnished rooms with layouts and semantics, 2021. 6

[8] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021. 7

[9] github. Controlnetgithubmodel. https://github.com/lllyasviel/ControlNet-v1-1-nightly#controlnet-11-segmentation, 2023. 5

[10] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *ICCV*, pages 1004–1014, 2021. 2

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 6

[12] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. *arXiv preprint arXiv:2303.11989*, 2023. 1, 3, 4, 6, 7, 8

[13] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *ArXiv*, abs/2305.02463, 2023. 2

[14] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, page 0, 2006. 6

[15] Chenguo Lin and Yadong Mu. *arXiv preprint arXiv:2402.04717*, 2024. 3

[16] Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. Coco-gan: Generation by parts via conditional coordinating. In *ICCV*, pages 4512–4521, 2019. 2, 3

[17] Chieh Hubert Lin, Hsin-Ying Lee, Yen-Chi Cheng, Sergey Tulyakov, and Ming-Hsuan Yang. Infinitygan: Towards infinite-pixel image synthesis. *arXiv preprint arXiv:2104.03963*, 2021. 2, 3

[18] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, pages 300–309, 2023. 1, 2

[19] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2

[20] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. *NeurIPS*, 34: 12013–12026, 2021. 2, 3, 4, 6, 7

[21] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. pages 8748–8763. PMLR, 2021. 6

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2, 4

[24] 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. *NeurIPS*, 35:36479–36494, 2022. 2

[25] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 6

[26] Jonas Schult, Sam Tsai, Lukas Höllein, Bichen Wu, Jialiang Wang, Chih-Yao Ma, Kunpeng Li, Xiaofang Wang, Felix Wimbauer, Zijian He, et al. Controlroom3d: Room generation using semantic proxy rooms. *arXiv preprint arXiv:2312.05208*, 2023. 3

[27] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*, 2023. 1

[28] Ka Chun Shum, Hong-Wing Pang, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Conditional 360-degree image synthesis for immersive indoor scene decoration. *arXiv preprint arXiv:2307.09621*, 2023. 2, 3

[29] Liangchen Song, Liangliang Cao, Hongyu Xu, Kai Kang, Feng Tang, Junsong Yuan, and Yang Zhao. Roomdreamer: Text-driven 3d indoor scene synthesis with coherent geometry and texture. *arXiv preprint arXiv:2305.11337*, 2023. 3

[30] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *CVPR*, pages 1047–1056, 2019. 5

[31] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Scene graph denoising diffusion probabilistic model for generative indoor scene synthesis. *arXiv preprint arXiv:2303.14207*, 2023. 2, 3, 4, 6, 7

[32] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *arXiv preprint arXiv:2306.03881*, 2023. 6

[33] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv preprint arXiv:2307.01097*, 2023. 1, 2, 3, 6, 7, 8

[34] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *ECCV*, pages 836–850. Springer, 2014. 6

[35] Guangcong Wang, Peng Wang, Zhaoxi Chen, Wenping Wang, Chen Change Loy, and Ziwei Liu. Perf: Panoramic neural radiance field from a single panorama. *arXiv preprint arXiv:2310.16831*, 2023. 3, 4, 5

[36] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, pages 12619–12629, 2023. 2

[37] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*, pages 106–115. IEEE, 2021. 6

[38] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 2

[39] Tianhao Wu, Chuanxia Zheng, and Tat-Jen Cham. Ipo-ldm: Depth-aided 360-degree indoor rgb panorama outpainting via latent diffusion model. *arXiv preprint arXiv:2307.03177*, 2023. 5

[40] Bangbang Yang, Wenqi Dong, Lin Ma, Wenbo Hu, Xiao Liu, Zhaopeng Cui, and Yuewen Ma. Dreamspace: Dreaming your room space with text-driven panoramic texture propagation. *arXiv preprint arXiv:2310.13119*, 2023. 3

[41] Ilwi Yun, Chanyong Shin, Hyunku Lee, Hyuk-Jae Lee, and Chae Eun Rhee. Egformer: Equirectangular geometry-biased transformer for 360 depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6101–6112, 2023. 5, 6

[42] Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. Text2nerf: Text-driven 3d scene generation with neural radiance fields. *arXiv preprint arXiv:2305.11588*, 2023. 1

[43] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2, 4

[44] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *ECCV*, pages 519–535. Springer, 2020. 2, 5, 6