SceneWeaver: Text-Driven Scene Generation with Geometry-aware Gaussian Splatting

Xiaolu Hou[1,2] XLHOU23@M.FUDAN.EDU.CN
Mingcheng Li[1,2] MINGCHENGLI21@M.FUDAN.EDU.CN
Jiawei Chen[1,2] 22210860033@M.FUDAN.EDU.CN
Dingkang Yang[1,2] DKYANG20@FUDAN.EDU.CN
Ziyun Qian[1,2] ZYQIAN22@M.FUDAN.EDU.CN
Lihua Zhang[1,2,3,4,5*] LIHUAZHANG@FUDAN.EDU.CN
[1] *Academy for Engineering and Technology, Fudan University, Shanghai, China*
[2] *Cognition and Intelligent Technology Laboratory (CIT Lab), Shanghai, China*
[3] *Institute of Metaverse & Intelligent Medicine, Fudan University, Shanghai*
[4] *Jilin Provincial Key Laboratory of Intelligence Science and Engineering, Changchun, China*
[5] *Engineering Research Center of AI and Robotics, Ministry of Education, Shanghai, China*

Editors: Vu Nguyen and Hsuan-Tien Lin

Abstract

With the widespread use of virtual reality applications, 3D scene generation has become a challenging new research frontier. 3D scenes have highly complex structures, so it is crucial to ensure that the output is dense, coherent, and includes all necessary structures. Many current 3D scene generation methods rely on pre-trained text-to-image diffusion models and monocular depth estimators, but they often lack rich geometric constraint information within the scene, leading to geometric distortion in the generated results. Therefore, we propose a two-stage geometry-aware progressive scene generation framework, SceneWeaver, which creates diverse, high-quality 3D scenes from text or image inputs. In the first stage, we introduce a multi-level depth refinement mechanism combined with image inpainting and point cloud updating strategies to construct a high-quality initial point cloud. In the second stage, 3D Gaussians are initialized based on the point cloud and continuously optimized. To address the challenge of insufficient geometric constraints in the Gaussian Splatting optimization process, we utilize the rich appearance and geometry information within the scene to perform a geometry-aware optimization, resulting in high-quality scene generation results. Comprehensive experiments across multiple scenes demonstrate the significant potential and advantages of our framework compared with several baselines.

Keywords: Scene Generation · Gaussian Splatting · Generative Models

1. Introduction

The current demand for 3D content in virtual reality is substantial. However, creating such content is labor-intensive and requires extensive domain knowledge, making 3D content generation a challenging research frontier. Transitioning from 2D to 3D content generation is a complex process involving the intricate replication of the physical world, particularly about spatial relationships and depth perception. In the 2D domain, the availability of extensively annotated datasets has significantly advanced text-to-image generative models

* Corresponding Author

(Rombach et al., 2022). In contrast, the scarcity of annotated 3D datasets limits the application of supervised learning in 3D content generation (Ouyang et al., 2023).

To address the challenge of scarce 3D annotated data, several object-centered generation methods (Poole et al., 2022; Lin et al., 2023) attempt to optimize 3D content by extracting 2D priors from pre-trained diffusion models as supervisory signals without annotated data. Moreover, progressive outward-facing generation methods (Höllein et al., 2023; Chung et al., 2023) combining a pre-trained text-conditioned diffusion model (Rombach et al., 2022) and a monocular depth estimator (Bhat et al., 2023) have received wider attention due to their ability to facilitate the generation of complex scenes. However, these methods suffer from the following limitations: (i) The boundaries of objects in the depth map are not clear enough and some necessary details are missing. This boundary-blurring problem may affect the subsequent rendering quality and accuracy. (ii) Insufficient scene constraints during the scene optimization stage may lead to geometric distortion problems in the generated scenes.

To address the aforementioned issues, we propose SceneWeaver, a geometry-aware progressive scene generation framework that can generate diverse and high-quality 3D scenes from text or image inputs. Specifically, we design a multi-level depth refinement mechanism to enhance the edges of objects in the depth map by fusing the relative depth features with the metric depth features. Moreover, a geometry-aware optimization strategy is presented to sufficiently mine and exploit the rich geometric and appearance information, thus improving the quality of scene generation. Our contributions are summarized as follows:

• We present SceneWeaver, a text-driven geometry-aware progressive scene generation framework to generate high-quality 3D scenes.

• We introduce a multi-level depth refinement mechanism to optimize the edges of objects in the depth map by fusing multi-level depth semantics. Furthermore, we propose a geometry-aware optimization strategy to capture the rich appearance and geometric information within the scene to optimize the scene generation quality effectively.

• Comprehensive experiments demonstrate that the scenes generated by our framework significantly outperform baselines in terms of fidelity and geometric consistency, proving its significant potential and advantages in complex 3D scene generation.

2. Related Work

2.1. 3D Scene Representation.

In the field of 3D content generation, choosing the right 3D representation is crucial. Classical explicit representations (Munkberg et al., 2022; Daniels et al., 2008) play an important role in computer graphics and vision, as they allow intuitive control over each element and enable efficient processing of common scenes through the rasterization rendering pipeline. However, in highly complex or larger-scale scenes, explicit representations may encounter challenges such as increased memory consumption and reduced rendering efficiency. Thus, some studies (Park et al., 2019; Mildenhall et al., 2021) have started to employ neural networks for implicit representation. Neural Radiance Field (NeRF) (Mildenhall et al., 2021) as a typical method to represent the scene as an implicit neural field. End-to-end differentiable rendering is achieved by querying information about the scene at different positions through a Multi-Layer Perceptron. Although methods dedicated to improving rendering

quality (Barron et al., 2022) and speed (Müller et al., 2022; Fridovich-Keil et al., 2022) are emerging, these methods still face the challenge of balancing rendering speed and quality.

In contrast, the proposal of unstructured, explicit GPU-friendly 3D Gaussians (Kerbl et al., 2023) makes it possible to achieve faster rendering speed and better rendering quality without the need for neural components. 3D Gaussian Splatting (3DGS) models a scene from a point cloud using a set of Gaussian ellipsoids. It builds learnable 3D Gaussian representations centered on each point and achieves efficient rendering by rasterizing these Gaussian ellipsoids into images. Therefore, we use 3DGS as a scene representation to achieve faster scene generation speed while maintaining high-quality rendering results.

2.2. Text-to-3D Generation.

Generating 3D content using natural language allows users to express their requirements clearly, without needing specialized 3D modeling skills or knowledge. Some methods (Mohammad Khalid et al., 2022; Lee and Chang, 2022) utilize a priori knowledge from CLIP (Radford et al., 2021) to constrain the optimization process. Text-to-image diffusion models (Rombach et al., 2022) offer more powerful generation capabilities than CLIP, owing to their extensive training data and superior architecture. DreamFusion (Poole et al., 2022) and SJC (Wang et al., 2023a) introduce Score Distillation Sampling (SDS) to extract a priori knowledge from pre-trained diffusion models. Subsequent works (Lin et al., 2023; Wang et al., 2024) further refine the SDS-based optimization process to enhance generation quality. While these object-centered approaches can generate high-quality 3D objects, they are challenging to extend to 3D scene generation with outward-facing viewpoints. Because scene generation must consider dense and coherent outward-facing viewpoints while ensuring the inclusion of high-quality textures and essential structures (Höllein et al., 2023).

With the widespread use of diffusion models in image inpainting, many methods (Fridman et al., 2024; Höllein et al., 2023; Yu et al., 2024; Chung et al., 2023; Ouyang et al., 2023) utilize the inpainting capabilities of diffusion models combined with monocular depth estimators (Ranftl et al., 2020; Bhat et al., 2023) to inpaint and update the scene progressively. Among them, SceneScape (Fridman et al., 2024) and Text2Room (Höllein et al., 2023) use meshes as the 3D representation, which limits the generated results to indoor scenes. When applied to outdoor scenes, they face problems with distortions or over-smoothed surfaces. In contrast, LucidDreamer (Chung et al., 2023) and Text2Immersion (Ouyang et al., 2023) use 3DGS as the 3D representation, employing a progressive scene generation framework that follows the optimization objective in Kerbl et al. (2023), which enables domain-free scene generation. However, the generated scenes may have geometric distortions due to insufficient constraints in the optimization process. Therefore, we introduce a multi-level depth refinement mechanism and utilize the rich appearance and geometric information within the scene to achieve geometry-aware 3D scene optimization.

3. Preliminaries

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) utilizes a set of 3D Gaussians to model a scene and quickly render images by "splatting" (Yifan et al., 2019) 3D Gaussians to the 2D plane. Each Gaussian is defined by a full 3D covariance

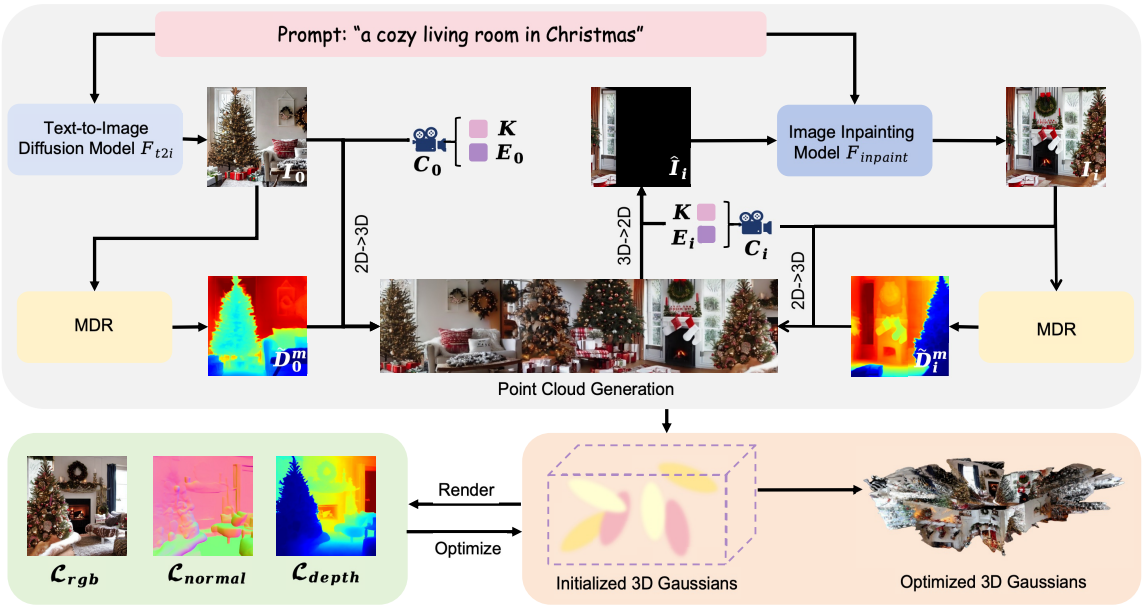

Figure 1: **The overall framework of the proposed SceneWeaver**. SceneWeaver constructs a point cloud of the scene from a text prompt by progressive inpainting, refining, and updating. In the refining step, the Multi-level Depth Refinement (MDR) mechanism is used to clarify the boundaries of each object in the depth map. Then, the Geometry-aware Optimization (GAO) strategy of 3D Gaussians is performed using regularization terms that capture the scene's rich geometric information to generate a high-quality scene.

matrix $\boldsymbol{\Sigma}$ and its position (mean) $\boldsymbol{\mu}$ in world space:

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

where $\boldsymbol{\Sigma}$ can be computed from the scaling matrix $\boldsymbol{S}$ and rotation matrix $\boldsymbol{R}$ as $\boldsymbol{\Sigma} = \boldsymbol{RSS}^T\boldsymbol{R}^T$. The $\boldsymbol{S}$ and $\boldsymbol{R}$ are further denoted by the scaling vector $\boldsymbol{s} \in \mathbb{R}^3$ and the rotation quaternion $\boldsymbol{q} \in \mathbb{R}^4$. 3DGS employs alpha-blending to compute the color $\hat{\boldsymbol{C}}_{gs}$ of each pixel:

$$\hat{\boldsymbol{C}}_{gs} = \sum_{i\in\mathcal{N}} \boldsymbol{c}_i\alpha_i' \prod_{j=1}^{i-1}(1-\alpha_j'), \tag{2}$$

where $\mathcal{N}$ represents the projected 2D Gaussians that cover the pixel, ordered by depth in the camera space. $\boldsymbol{c}_i$ is the view-dependent colors computed from the spherical harmonic (SH) coefficients of the i^{th} Gaussian and α_i' is the blending coefficient. Per-pixel depth $\hat{\boldsymbol{D}}_{gs}$ (Turkulainen et al., 2024) can be rendered using a discrete volume rendering similar to $\hat{\boldsymbol{C}}_{gs}$:

$$\hat{\boldsymbol{D}}_{gs} = \sum_{i\in\mathcal{N}} \boldsymbol{d}_i\alpha_i' \prod_{j=1}^{i-1}(1-\alpha_j'), \tag{3}$$

where $\boldsymbol{d}_i$ is the depth of the i^{th} Gaussian in camera space.

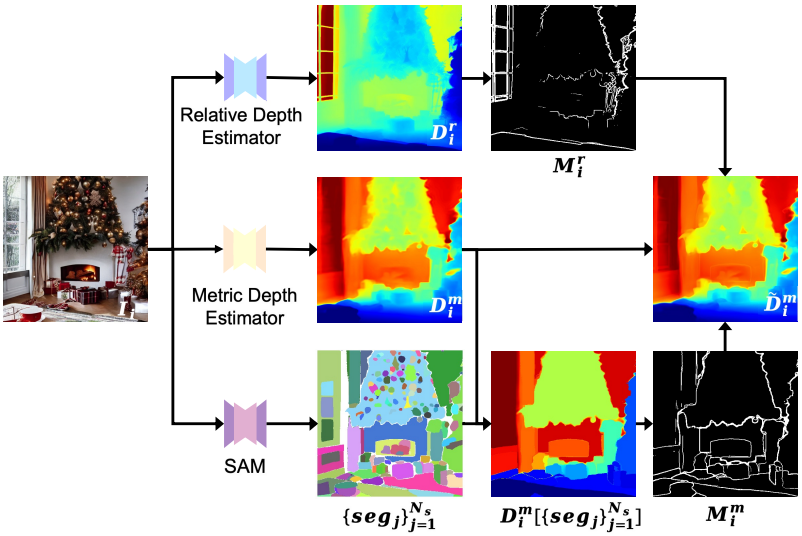

Figure 2: **The workflow of the Multi-level Depth Refinement (MDR) mechanism.** MDR starts from the image $\boldsymbol{I}_i$ and processes the metric depth $\boldsymbol{D}_i^m$ using the object mask $\{seg_j\}_{j=1}^{N_s}$ obtained by SAM. The edge map $\boldsymbol{M}_i^m$ and $\boldsymbol{M}_i^r$ computed by the Sobel operator are combined to optimize $\boldsymbol{D}_i^m$ and obtain the refined depth $\tilde{\boldsymbol{D}}_i^m$.

The parameters of 3D Gaussians are optimized by stochastic gradient descent strategy. Optimization of the parameters alternates with adaptive density control to better represent the scene. The loss term is computed by combining the $\mathcal{L}_1$ and $\mathcal{L}_{D-SSIM}$ terms between the rendered image and the ground truth image:

$$\mathcal{L}_{rgb} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM}, \tag{4}$$

where the trade-off hyperparameter λ is set to 0.2.

Progressive Scene Generation Framework. Given a text prompt y, we aim to generate 3D scenes matching the prompt. To achieve this, recent works (Chung et al., 2023; Ouyang et al., 2023) use a text-conditioned diffusion model (Rombach et al., 2022) and a monocular depth estimator (Bhat et al., 2023) to inpaint and update the scene progressively. The pre-trained text-to-image diffusion model F_{t2i} is used to generate the initial image $\boldsymbol{I}_0 \in \mathbb{R}^{3\times H\times W}$ from the text prompt y, where H and W denote the height and width of the image. The monocular depth estimator F_d is then used to obtain the corresponding depth map $\boldsymbol{D}_0^m \in \mathbb{R}^{H\times W}$ from $\boldsymbol{I}_0$. The predefined cameras $\{\boldsymbol{C}_i\}_{i=0}^{N}$ are denoted by the extrinsic parameters $\boldsymbol{E}_i \in \mathbb{R}^{3\times 4}$ and the shared intrinsic parameter $\boldsymbol{K} \in \mathbb{R}^{3\times 3}$, where $N+1$ denotes number of cameras. Based on the initial camera $\boldsymbol{C}_0$, 2D pixels are lifted to 3D space to construct the initial point cloud $\boldsymbol{P}_0$:

$$\boldsymbol{P}_0 = f_{2\to 3}(\boldsymbol{I}_0, \boldsymbol{D}_0^m, \boldsymbol{E}_0, \boldsymbol{K}), \tag{5}$$

where $f_{2\to 3}$ is a series of geometric transformations that convert the coordinates of each pixel in the RGBD image from the pixel coordinate system to the world coordinate system.

The generated point cloud for each camera pose should be fused to the existing one. Specifically, at the i^{th} $(i \neq 0)$ camera, the existing point cloud $\boldsymbol{P}_{i-1}$ is projected into 2D-

pixel space. Due to the change in camera position, this projection results in a partial image $\hat{\boldsymbol{I}}_i$ and a mask $\hat{\boldsymbol{M}}_i$ indicating the area to be inpainted:

$$\hat{\boldsymbol{I}}_i, \hat{\boldsymbol{M}}_i = f_{3\to 2}(\boldsymbol{P}_{i-1}, \boldsymbol{E}_i, \boldsymbol{K}), \tag{6}$$

where $f_{3\to 2}$ is a series of geometric transformations made to project the point cloud from the world coordinate system to the pixel coordinate system.

The image inpainting model $F_{inpaint}$ is used to obtain the full image $\boldsymbol{I}_i$ based on the partial image $\hat{\boldsymbol{I}}_i$, the mask $\hat{\boldsymbol{M}}_i$ and the text prompt y. The monocular depth estimator F_d is utilized to get the corresponding depth $\boldsymbol{D}_i^m$. Since there is some difference between the neighboring depth maps, $\boldsymbol{D}_i^m$ needs to be processed by minimizing the distance between the overlapping regions of the two point clouds to get the aligned depth $\bar{\boldsymbol{D}}_i$:

$$\bar{\boldsymbol{D}}_i = f_{align}(f_{2\to 3}(\boldsymbol{I}_i, \boldsymbol{D}_i^m, \boldsymbol{E}_i, \boldsymbol{K}), \boldsymbol{P}_{i-1}, \hat{\boldsymbol{M}}_i = 1), \tag{7}$$

where $f_{align}(\cdot)$ is the function that minimizes the distance between the overlapping regions ($\hat{\boldsymbol{M}}_i = 1$) of the two point clouds. Then the inpainted 2D pixels need to be transformed from pixel coordinates to world coordinates to get the updated point cloud $\boldsymbol{P}_i$:

$$\boldsymbol{P}_i = f_{update}(f_{2\to 3}(\boldsymbol{I}_i, \bar{\boldsymbol{D}}_i, \boldsymbol{E}_i, \boldsymbol{K}), \boldsymbol{P}_{i-1}, \hat{\boldsymbol{M}}_i = 0), \tag{8}$$

where $f_{update}(\cdot)$ is the function that fuses the new point cloud into the existing point cloud $\boldsymbol{P}_{i-1}$. $\hat{\boldsymbol{M}}_i = 0$ means only the inpainted pixels needs to be transformed. Repeat the above steps N times to get the final point cloud $\boldsymbol{P}_N$. 3D Gaussians are initialized by $\boldsymbol{P}_N$ and optimized by following the optimization objective $\mathcal{L}_{rgb}$ proposed in Kerbl et al. (2023).

4. Method

We propose SceneWeaver, a geometry-aware progressive framework for text-driven 3D scene generation as shown in Figure.1. SceneWeaver utilizes image inpainting, depth estimation, and point cloud updating strategies in the progressive framework to construct a high-quality point cloud. The point cloud is then used to initialize the 3D Gaussians and the 3D Gaussians are continuously optimized to generate the final 3D scene. In particular, a multi-level depth refinement mechanism (Sec.4.1) is used for depth refinement, which optimizes object edges in the depth map by fusing multi-level depth semantics. The geometry-aware 3D Gaussians optimization strategy (Sec.4.2) is used to capture the rich appearance and geometric information in the scene, which effectively improves the quality of scene generation.

4.1. Multi-level Depth Refinement Mechanism

The depth map $\boldsymbol{D}_i^m$ is the metric depth which is the real distance from the objects to the camera. In practical implementation, we found that the boundaries of objects in the depth map are unclear, and some necessary details are missing (Chung et al., 2023). Thus, we propose the Multi-level Depth Refinement (MDR) mechanism, as shown in Figure.2. It aims to clarify the boundaries of each object in the depth map by integrating the multi-level semantics of the relative depth map $\boldsymbol{D}_i^r$ obtained by Marigold (Ke et al., 2024) and the metric depth map $\boldsymbol{D}_i^m$ obtained by Zoedepth (Bhat et al., 2023).

Specifically, MDR starts from an RGB image and uses the edge maps obtained from the relative depth $\boldsymbol{D}_i^r$ and the metric depth $\boldsymbol{D}_i^m$ to optimize $\boldsymbol{D}_i^m$ and obtain the refined depth $\tilde{\boldsymbol{D}}_i^m$. The Segment Anything Model (SAM) (Kirillov et al., 2023) is a robust basic vision model recognized for its excellent zero-shot capability in image segmentation. SAM efficiently generates segmentation masks $\{seg_j\}_{j=1}^{N_s}$ for all objects from an entire image, and N_s is the number of masks. Inspired by Yu et al. (2024), we combine the depth map $\boldsymbol{D}_i^m$ with the segmentation mask to compute the processed depth $\boldsymbol{D}_i^m[seg_j]$ as follows:

$$\boldsymbol{D}_i^m\left[seg_j\right] \leftarrow \begin{cases} f_{median}\left(\boldsymbol{D}_i^m\left[seg_j\right]\right), & \text{if } \Delta \boldsymbol{D}_i^j < T, \\ \boldsymbol{D}_i^m\left[seg_j\right], & \text{otherwise}, \end{cases} \tag{9}$$

for $j \in \{1, ..., N_s\}$, where $f_{median}(\cdot)$ is a function that returns the median value of the input. $\Delta \boldsymbol{D}_i^j$ denotes the range of disparity (the reciprocal of $\boldsymbol{D}_i^m$), computed as the difference between the maximum and minimum disparity values within the segmentation mask, and T is the threshold used for the computation.

Subsequently, the Sobel operator is applied to $\boldsymbol{D}_i^m[\{seg_j\}_{j=1}^{N_s}]$ and $\boldsymbol{D}_i^r$ to obtain the edge map $\boldsymbol{M}_i^m$ and $\boldsymbol{M}_i^r$. For the depth map $\boldsymbol{D}_i^m$ at each viewpoint, we optimize by computing the loss function $\mathcal{L}_r$ to get the refined depth $\tilde{\boldsymbol{D}}_i^m$. $\mathcal{L}_r$ is computed as follows:

$$\mathcal{L}_r = \mathcal{F}_{MSE}(\boldsymbol{M}_i^r, \boldsymbol{M}_i^m) + \lambda_r \mathcal{F}_{MSE}(\boldsymbol{D}_i^m \boldsymbol{M}_i^r, \boldsymbol{D}_i^m \boldsymbol{M}_i^m), \tag{10}$$

for $i \in \{0, ..., N\}$, where $\mathcal{F}_{MSE}(\cdot)$ is the Mean Squared Error (MSE) function and λ_r is set to 0.01. $\boldsymbol{D}_i^m$ in Eq.7 is replaced by the refined depth $\tilde{\boldsymbol{D}}_i^m$ to compute the aligned depth $\bar{\boldsymbol{D}}_i$.

4.2. Geometry-Aware Optimization Strategy

Previous methods (Chung et al., 2023; Ouyang et al., 2023) follow the optimization objective $\mathcal{L}_{rgb}$ in Eq.4 to optimize the 3D Gaussians in the scene. However, the generated scenes are geometrically distorted due to insufficient scene constraints. Therefore, we design a Geometry-Aware Optimization (GAO) strategy. The core idea is to utilize the rich appearance and geometric information to generate realistic and high-quality 3D scenes.

We use the gradient-aware depth loss $\mathcal{L}_{depth}^{value}$ based on the gradients of the RGB image for adaptive depth regularization (Turkulainen et al., 2024). In edge regions with large absolute values of the image gradients, the depth loss is reduced, thus ensuring stronger execution of regularization in smoother textureless regions. $\mathcal{L}_{depth}^{value}$ can be calculated as follows:

$$\mathcal{L}_{depth}^{value} = \exp(-|\nabla \boldsymbol{I}_{gt}|) \frac{1}{|\boldsymbol{D}_{gs}|} \sum \log(1 + \|\hat{\boldsymbol{D}}_{gs} - \hat{\boldsymbol{D}}_{gt}\|_1), \tag{11}$$

where $\nabla \boldsymbol{I}_{gt}$ are the gradients of the ground truth RGB image and $\|\cdot\|_1$ represents ℓ_1 norm function. $\hat{\boldsymbol{D}}_{gt}$ and $\hat{\boldsymbol{D}}_{gs}$ are the per-pixel value of the ground truth and rendered depth map at each viewpoint respectively. $|\boldsymbol{D}_{gs}|$ denotes the number of pixels in the rendered depth map $\boldsymbol{D}_{gs}$. The ground truth image and depth map can be obtained by reprojecting the point cloud $\boldsymbol{P}_N$, and normalization is required to compute $\mathcal{L}_{depth}^{value}$. Moreover, we incorporate a Total Variation (TV) loss $\mathcal{L}_{depth}^{smooth}$ on the rendered depth maps as a penalty, together with $\mathcal{L}_{depth}^{value}$ to form the depth supervision term $\mathcal{L}_{depth}$. $\mathcal{L}_{depth}^{smooth}$ can be calculated as follows:

$$\mathcal{L}_{depth}^{smooth} = \frac{1}{|\boldsymbol{D}_{gs}|} \sum_{i,j} \left(\left| \boldsymbol{D}_{gs}^{i,j} - \boldsymbol{D}_{gs}^{i+1,j} \right| + \left| \boldsymbol{D}_{gs}^{i,j} - \boldsymbol{D}_{gs}^{i,j+1} \right| \right). \tag{12}$$

Method	CLIP-Score ↑	BRISQUE ↓	NIQE ↓	CLIP-IQA		
				Quality ↑	Colorful ↑	Sharp ↑
SceneScape (Fridman et al., 2024)	31.42	30.27	3.80	0.58	0.76	0.38
Text2Room (Höllein et al., 2023)	29.61	27.91	3.51	0.59	0.79	0.32
LucidDreamer (Chung et al., 2023)	31.23	23.18	2.85	0.64	0.80	0.43
Invisible-Stitch (Engstler et al., 2024)	31.18	23.07	3.20	0.62	0.71	0.42
WonderJourney (Yu et al., 2024)	31.09	30.55	3.46	0.58	0.75	0.44
Ours (w/o MDR)	31.50	21.78	2.83	0.66	0.81	**0.47**
Ours (w/o Depth Supervision)	31.50	22.02	2.73	0.65	0.81	0.43
Ours (w/o Normal Supervision)	31.52	20.87	2.61	0.66	0.78	0.44
Ours (SceneWeaver)	**32.09**	**20.33**	**2.60**	**0.70**	**0.82**	**0.47**

Table 1: **Performance comparison among the proposed framework and baselines.** The 2D metrics results include CLIP-Score, BRISQUE, NIQE, and CLIP-IQA (Quality, Colorful, Sharp). Our approach achieves the best results.

We further use normal information to optimize 3D Gaussians in the scene. Recently, surface normal estimation models (Bae and Davison, 2024) have demonstrated strong generalization capabilities. Therefore, we use the pre-trained surface normal estimation model to estimate normal map $\boldsymbol{N}_{gt}$ and $\boldsymbol{N}_{gs}$ from the ground truth and rendered RGB image. The difference between normal maps is minimized by computing the normal loss $\mathcal{L}_{normal}$:

$$\mathcal{L}_{normal} = \mathcal{F}_{MSE}(\boldsymbol{N}_{gs}, \boldsymbol{N}_{gt}). \tag{13}$$

Based on the above analysis, the 3D Gaussians of the scene can be optimized by the composite of all the loss functions:

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_1(\mathcal{L}_{depth}^{value} + \mathcal{L}_{depth}^{smooth}) + \lambda_2\mathcal{L}_{normal}, \tag{14}$$

where trade-off hyperparameters λ_1 and λ_2 are set to 0.8 and 2.0 respectively. $\mathcal{L}_{rgb}$ is the original optimization objective proposed in Kerbl et al. (2023).

5. Experiments

5.1. Experiment Setting

Implementation Details. Since our method is optimized for each input, no dataset is required to train the model. We use ZoeDepth (Bhat et al., 2023) and Marigold (Ke et al., 2024) as our depth estimator. The Stable Diffusion Inpainting model (Rombach et al., 2022) is used for text-conditioned image inpainting. We use the ViT-H model of SAM (Kirillov et al., 2023) for image segmentation. The pre-trained DSINE (Bae and Davison, 2024) model is used to estimate the surface normal from the RGB image. For text input, we use Stable Diffusion v1.5 (Rombach et al., 2022) to generate an initial image. For image input, LLaVa (Contributors, 2023) is used to create a description as a text prompt for text-conditioned inpainting. All experiments are done on a single Tesla V100 GPU.

Model Zoo. To evaluate the performance of SceneWeaver on the task of text-driven 3D scene generation, we compare SceneWeaver with five representative and reproducible methods, including progressive 3D scene generation methods: Text2Room (Höllein et al., 2023),

Figure 3: Qualitative comparison of our method and baselines.

Figure 4: Qualitative comparison of our method and baselines.

LucidDreamer (Chung et al., 2023), and Invisible-stitch (Engstler et al., 2024)), and perpetual view generation methods: SceneScape (Fridman et al., 2024) and WonderJourney (Yu et al., 2024)). Text2Room generates room-scale scenes represented by polygonal textured 3D meshes. LucidDreamer and Invisible-stitch generate diverse scenes represented by 3D Gaussians. SceneScape and WonderJourney both generate long videos with navigation effects. SceneScape generates scenes represented by meshes, and WonderJourney represents scenes at each frame using point clouds. We use the open-source code library of the above models and modify the inputs to start from the same image and text prompt.

Evaluation Metrics. Previous reference-based metrics do not apply to the generation task, such as PSNR and LPIPS (Zhang et al., 2018), because there are no 3D scenes associated with text prompts as ground truth. Thus, we used the six 2D metrics employed in the previous task to fully assess the quality of the generated scenes. We use the

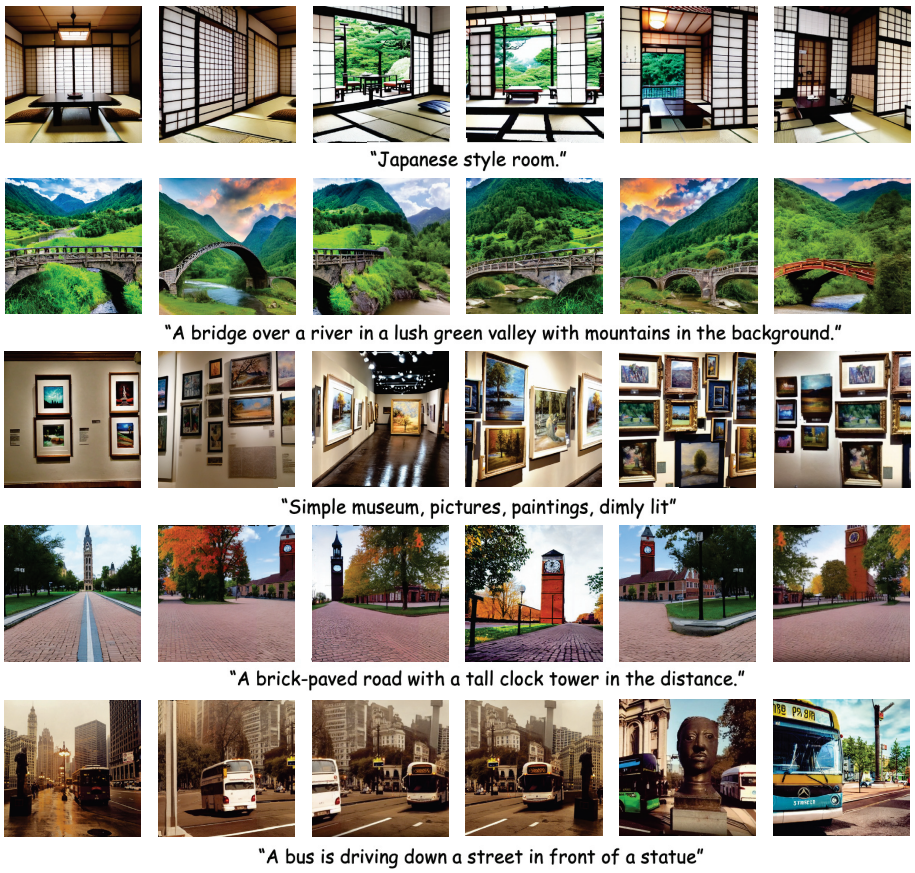

Figure 5: 3D Scenes Generated from Diverse Prompts or Images by SceneWeaver. Our approach contains the necessary scene structure and yields high-quality rendered results with realistic details.

Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) (Mittal et al., 2012a) and Natural Image Quality Evaluator (NIQE) (Mittal et al., 2012b) for reference-free quality evaluation of the rendered images of 3D scenes. CLIP score (Hessel et al., 2021) is used to measure the alignment of the rendered images and text prompts. We also assess the appearance and feel of the rendered images in a way that is more relevant to human perception by using the Colorful, Quality, and Sharp metrics in CLIP-IQA (Wang et al., 2023b).

5.2. Quantitative Results

We show the average quantitative results across multiple scenes in Table.1. All methods compute all 2D metrics by applying Gaussian noises to the rotation matrices and translation vectors of the training cameras and render 50 images of each scene from new viewpoints. We conclude from the experimental results in Table.1 as follows: (i) Overall, Our method outperforms baselines by generating higher quality 3D scenes, obtaining lower BRISQUE and NIQE values, and higher CLIP-Score and CLIP-IQA values. (ii) Our method obtains a BRISQUE score of 20.33 and a NIQE score of 2.60, which are 11.9% and 8.8% lower

Figure 6: 3D Scenes Generated from Same Prompt by SceneWeaver.

Figure 7: 3D Scenes in Artistic Styles by SceneWeaver.

compared to the former optimal scores. The optimal scores are achieved for the CLIP-IQA metrics that assess the appearance and perception of images. This demonstrates that our proposed key components effectively utilize the geometric information of the scene. This results in a significant reduction of distortion in the new perspective images rendered from the scene, highlighting the advantages of our method in terms of image quality, colorfulness, and sharpness. (iii) Our method achieves optimal results in CLIP-Score. This demonstrates that our method effectively reduces geometric distortions in the scene, which in turn enhances the alignment of the rendered image with the input text prompts for the new perspective.

5.3. Qualitative Results

In addition to the quantitative evaluation, we perform an intuitive qualitative evaluation. We show the rendered RGB images of our method and the 5 baselines in Figure.3 and Figure.4. We can conclude the following points: (i) SceneScape (Fridman et al., 2024), WonderJourney (Yu et al., 2024), and Invisible-stitch (Engstler et al., 2024) all create relatively complete scene structures and appear to be coherently connected at particular viewpoints. However, when the images are rendered from new viewpoints, clear breaks can be observed in the boxed regions of the rendered images, and geometric distortions are evident. (ii) Text2Room (Höllein et al., 2023) uses polygonal meshes to represent scenes. It proposes a threshold filtering scheme for mesh fusion, which leads to the possibility that not all mesh stretching regions can be detected. This results in a large number of distorted and oversmoothed regions observable in the rendered images, which is particularly noticeable in outdoor scenes. (iii) LucidDreamer (Chung et al., 2023) is currently the most

visually appealing progressive scene generation framework, but we can observe artifacts and geometric distortions in the boxed parts of the rendered images. (iv) Compared to baselines, our approach contains the necessary scene structure and yields high-quality rendered results with realistic details, significantly reducing artifacts and geometric distortions.

Furthermore, we show more rendered results of indoor and outdoor scenes generated by SceneWeaver in Figure.5. It is worth noting that our method can generate scenes from text prompts, synthetic images, or real photos. The initial images in the first four rows are synthetic images generated by F_{t2i} based on text prompts. The initial image in the fifth row is a real photo, using the text prompt generated by F_{i2t} for text-conditioned image inpainting. Moreover, SceneWeaver can generate diverse scenes from a single text prompt (Figure.6), showcasing its flexibility and versatility. Additionally, it supports the creation of artistic style scenes (Figure.7), allowing for a wide range of creative and stylistic possibilities. This demonstrates the considerable potential and numerous advantages of our approach in generating complex and varied 3D scenes. It makes SceneWeaver a powerful tool for both practical applications and artistic endeavors.

5.4. Ablations Studies

To validate the effectiveness and necessity of the proposed components in SceneWeaver, the results of the ablation studies are shown in Table.1, Figure.8 and Figure.9. We use the six 2D metrics mentioned in Sec. 5.1 to evaluate each key component of the model. The quantitative evaluation results shown in Table.1 demonstrate that each key component of our method plays a very important role in the final scene generation.

Ablation on MDR. We evaluated the effectiveness of MDR (Figure.8 Left). Without applying MDR (Figure.8 (a)), it can be observed that the boundaries of the objects are not distinct enough, especially when dealing with detail-rich objects such as trees, window prisms, and picture frames. This boundary-blurring problem may affect the quality of the subsequent rendering. However, after MDR processing (Figure.8 (b)), the object boundaries are significantly enhanced, which shows that MDR has a significant advantage in improving the quality of the depth map. By enhancing the boundaries of objects, MDR provides a more accurate basis for subsequent 3D scene generation and rendering.

We also compare various methods for obtaining edge maps (Figure.8 Right). The traditional Canny algorithm detects edges using local image gradients and orientations (Figure.8 (c)). This results in edges that often include excessive texture and noise, lacking a holistic semantic understanding of the image. Directly extracting edge maps from the depth map $\boldsymbol{D}_i^m$ using the Sobel operator (Figure.8 (d)) loses significant object edge details. When using the Sobel operator on $\{seg_j\}_{j=1}^{N_s}$ (Figure.8 (e)), object edges exhibit considerable discontinuities or fragmentation. In contrast, our method (Figure.8 (f)) prioritizes object edge features, resulting in clearer and more continuous edge maps.

Ablation on GAO. The quantitative results presented in Table.1 demonstrate that depth and normal supervision effectively leverage the geometric information within the scene. This mitigates geometric distortions in the generated scenes and notably enhances image quality assessment metrics. In Figure.9, we showcase the rendered results of depth maps and RGB images corresponding to various regularization terms. Notably, the introduction of depth

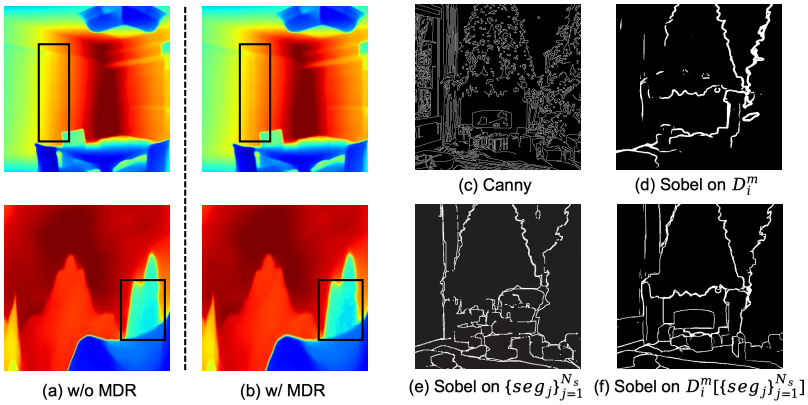

Figure 8: **Left: Ablation on MDR.** With MDR, the boundaries of objects in the boxed regions of the depth map become clearer and the details are more distinct. **Right: Ablation on Edge Detection Methods.** Our method (d) focuses more on the edge features of objects and obtains a clearer and more continuous edge map.

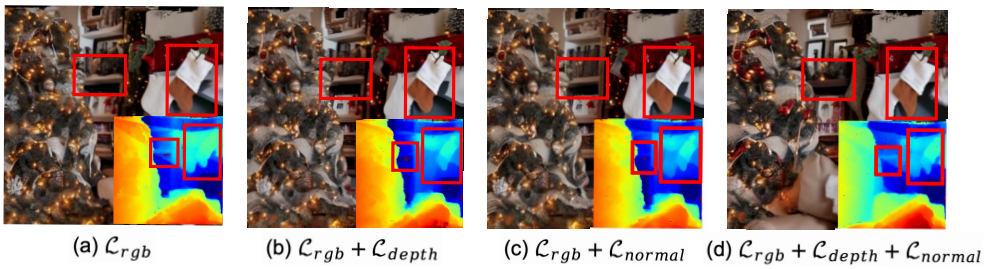

Figure 9: **Ablation on GAO.** (a) The depth map rendered by 3D Gaussians is not accurate. Applying (b) depth and (c) normal regularization terms effectively reduces the depth inaccuracy. (d) Combining the two regularizations, the optimal result can be seen in the boxed regions of the depth map.

and normal regularization terms visibly reduces depth inaccuracies within the boxed regions of the scene. Combining these with $\mathcal{L}_{rgb}$ term leads to more accurate geometry in the scene.

6. Conclusion

We present SceneWeaver, a geometry-aware progressive text-driven 3D scene generation framework. SceneWeaver uses a strategy incorporating image inpainting, a multi-level depth refinement mechanism, and point cloud updating, with regularization terms to capture rich appearance and geometric information for geometry-aware scene optimization. Comprehensive evaluations of existing open-source 3D scene generation methods on visual results and multiple metrics demonstrate our approach's great potential and advantages for complex 3D scene generation, which opens up more possibilities for future virtual reality applications.

Acknowledgments

This work is supported by the National Key R&D Program of China (2021ZD0113502).

References

Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9535–9545, 2024.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023.

XTuner Contributors. Xtuner: A toolkit for efficiently fine-tuning llm. https://github.com/InternLM/xtuner, 2023.

Joel Daniels, Cláudio T Silva, Jason Shepherd, and Elaine Cohen. Quadrilateral mesh simplification. *ACM transactions on graphics (TOG)*, 27(5):1–9, 2008.

Paul Engstler, Andrea Vedaldi, Iro Laina, and Christian Rupprecht. Invisible stitch: Generating smooth 3d scenes with depth inpainting. *arXiv preprint arXiv:2404.19758*, 2024.

Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. Scenescape: Text-driven consistent scene generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7909–7920, 2023.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024.

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

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

Han-Hung Lee and Angel X Chang. Understanding pure clip guidance for voxel grid nerf models. *arXiv preprint arXiv:2209.15172*, 2022.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

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

Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012a.

Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012b.

Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 conference papers*, pages 1–8, 2022.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022.

Hao Ouyang, Kathryn Heal, Stephen Lombardi, and Tiancheng Sun. Text2immersion: Generative immersive scene with 3d gaussians. *arXiv preprint arXiv:2312.09242*, 2023.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

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

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. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *arXiv preprint arXiv:2403.17822*, 2024.

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, 2023a.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2555–2563, 2023b.

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. *Advances in Neural Information Processing Systems*, 36, 2024.

Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019.

Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. Wonderjourney: Going from anywhere to everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6658–6667, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.