BloomScene: Lightweight Structured 3D Gaussian Splatting for Crossmodal Scene Generation

Anonymous submission

Abstract

With the widespread use of virtual reality applications, 3D scene generation has become a new challenging research frontier.3D scenes have highly complex structures and need to ensure that the output is dense, coherent, and contains all necessary structures. Many current 3D scene generation methods rely on pre-trained text-to-image diffusion models and monocular depth estimators. However, the generated scenes occupy large amounts of storage space and often lack effective regularisation methods, leading to geometric distortions. To this end, we propose BloomScene, a lightweight structured 3D Gaussian splatting for crossmodal scene generation, which creates diverse and high-quality 3D scenes from text or image inputs. Specifically, a crossmodal progressive scene generation framework is proposed to generate coherent scenes utilizing incremental point cloud reconstruction and 3D Gaussian splatting. Additionally, we propose a hierarchical depth prior-based regularization mechanism that utilizes multi-level constraints on depth accuracy and smoothness to enhance the realism and continuity of the generated scenes. Ultimately, we propose a structured context-guided compression mechanism that exploits structured hash grids to model the context of unorganized anchor attributes, which significantly eliminates structural redundancy and reduces storage overhead. Comprehensive experiments across multiple scenes demonstrate the significant potential and advantages of our framework compared with several baselines.

Introduction

Currently, there is a growing demand for 3D content in virtual reality. However, creating 3D content is not only time-consuming but also requires deep expertise, making 3D content generation a challenging frontier. In the 2D domain, sufficient annotated datasets have greatly contributed to the development of text-to-image generation models (Rombach et al. 2022), enabling users to generate images through natural language. However, the shortage of 3D annotated datasets limits the application of supervised learning in 3D content generation (Ouyang et al. 2023). To address this challenge, recent studies (Poole et al. 2022; Lin et al. 2023) have extracted 2D priors from diffusion models through a time-consuming distillation process to optimize 3D content generation. However, these approaches (Wang et al. 2024) have limitations when extended to fine-grained scenes with outward-facing viewpoints. Therefore, several methods (Höllein et al. 2023; Ouyang et al. 2023; Chung et al. 2023) that combine pre-trained text-to-image generation models (Rombach et al. 2022) with monocular depth estimators (Bhat et al. 2023; Ranftl et al. 2020) are receiving increasing attention due to their advantages in complex scene generation.

Previous work utilized NeRF (Mildenhall et al. 2021) for scene generation. Text2NeRF (Zhang et al. 2024) generates 3D scenes using an incremental framework, although it generates higher quality scenes, the generation time required is very long. Recently, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has been widely used for high-quality scene generation due to its excellent generation quality and real-time rendering capabilities. Among them, LucidDreamer (Chung et al. 2023) and Text2Immersion (Ouyang et al. 2023) use an incremental scene generation framework that follows the optimization goals in 3DGS to achieve domain-free scene generation. Although previous 3DGS-based approaches have made some progress in scene generation, they still suffer from the following limitations: (i) 3DGS requires millions of 3D Gaussians to represent each scene, leading to high memory requirements, increasing storage costs and end-device burdens; (ii) relying only on photometric loss in scene optimization, lacking sufficient regularisation techniques, which is prone to artifacts and ambiguities.

To address the above problems, we propose BloomScene, a lightweight structured 3D Gaussian splatting for crossmodal scene generation for high-quality 3D scene generation. BloomScene has the following three core contributions. (i) We propose a crossmodal progressive scene generation framework for generating 3D scenes via progressive point cloud reconstruction and 3D Gaussian splatting. (ii) Additionally, a hierarchical depth prior-based regularization mechanism is proposed to enhance the realism and continuity of the scene by implementing multi-level depth accuracy constraints and smoothness constraints. (iii) We propose a structured context-guided compression mechanism, which leverages a structured hash grid to model the context of unorganized anchor attributes, thus sufficiently compressing the model storage space. 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.

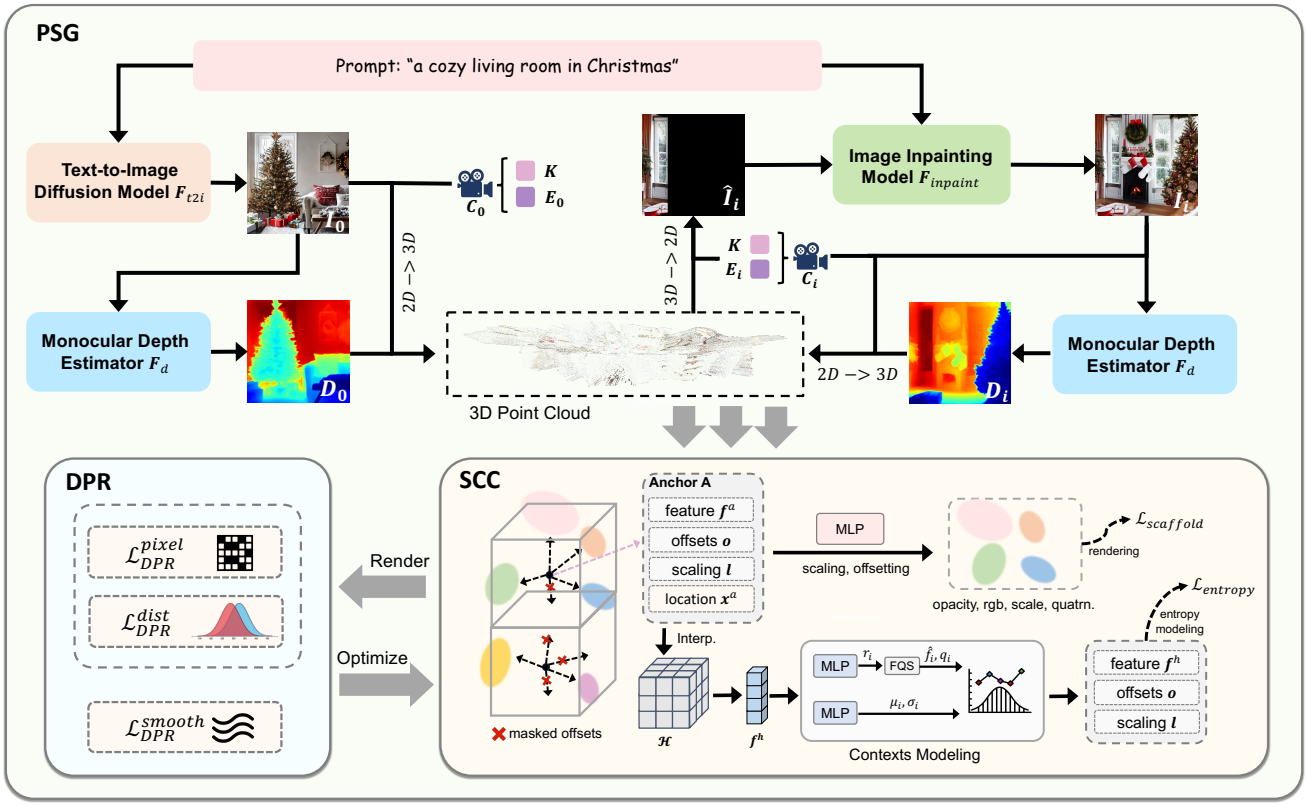

Figure 1: **The overall framework of the proposed BloomScene**. BloomScene utilizes the proposed crossmodal Progressive Scene Generation (PSG) framework to progressively generate 3D scenes from the text prompts. Moreover, the hierarchical Depth Prior-based Regularization (DPR) mechanism is applied to the 3DGS to enhance the realism and continuity of the generated scene. Eventually, Structured Context-guided Compression (SCC) is employed to mine structural correlations in 3DGS and reduce the number of parameters.

Related Work

Crossmodal 3D Scene Generation

Generating 3D models through language enables users to achieve requirements without modeling skills. Existing approaches (Mohammad Khalid et al. 2022; Lee and Chang 2022; Poole et al. 2022; Lin et al. 2023; Wang et al. 2024; Tang et al. 2023) optimize 3D content using a priori knowledge from pre-trained models (Radford et al. 2021; Rombach et al. 2022). Although these methods have made progress in single-object generation, it is difficult to ensure texture and structural coherence in complex outward-facing viewpoint scene generation (Wang et al. 2024). The application of diffusion models in image inpaint further advances progressive scene generation (Fridman et al. 2024; Höllein et al. 2023; Chung et al. 2023; Ouyang et al. 2023; Engstler et al. 2024; Yu et al. 2024), by incorporating a monocular depth estimator (Bhat et al. 2023; Ranftl et al. 2020) to update the scene. Among them, although LucidDreamer (Chung et al. 2023) and Text2Immersion (Ouyang et al. 2023) use 3DGS (Kerbl et al. 2023) for higher-quality scene generation, they are prone to artifacts and ambiguities due to their reliance on photometric loss alone. Therefore, we propose the hierarchical depth prior-based regularization mechanism for multi-level regularisation of 3DGS.

Efficient 3D Scene Representation.

In 3D content generation, the choice of an appropriate 3D representation is crucial. Classical explicit representations (Munkberg et al. 2022; Berger et al. 2014) provide intuitive geometric control and are suitable for simple scenes, but may face memory and rendering efficiency issues in complex scenes. Neural network-based implicit representation (Mildenhall et al. 2021; Müller et al. 2022; Barron et al. 2022) despite improved expressiveness, there is still a trade-off between speed and quality. 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) enables fast rendering and high-quality outputs, but high storage requirements impose an additional burden. For this reason, some methods focus on value (Fan et al. 2023a; Navaneet et al. 2023) or structure representation (Lu et al. 2024a) to reduce the computational burden. However, redundancy of structures or independence of anchors leads to more inefficient compression. To this end, we propose the structured context-guided compression mechanism, which utilizes a structured hash feature grid to achieve contextual modeling of anchor point attributes for further compression of 3DGS.

Methodology

Preliminaries

3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) introduces the 3D Gaussians as differential volumetric representations of radiance fields, allowing high-quality real-time novel view synthesis. A set of splats is initialized from the calibrated camera poses and the sparse point clouds produced through Structure-from-Motion (SfM) (Snavely, Seitz, and Szeliski 2006). Each Gaussian is represented by position $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, denoted as $G(\boldsymbol{x}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$. The covariance can be decomposed from a scaling matrix $\boldsymbol{S}$ and rotation matrix $\boldsymbol{R}$, expressed as $\boldsymbol{\Sigma} = \boldsymbol{RSS}^T\boldsymbol{R}^T$ with $\boldsymbol{S}$. To render the color, 3DGS further optimizes opacity and Spherical Harmonic (SH) coefficients, following the point-based differential rendering by rasterizing anisotropic splats with α-blending, denoted as:

$$\hat{\boldsymbol{C}} = \sum_i^N \mathbf{c}_i \boldsymbol{\alpha}_i \prod_j^{i-1}(1-\boldsymbol{\alpha}_j), \quad \hat{\boldsymbol{D}} = \sum_i^N \boldsymbol{d}_i \boldsymbol{\alpha}_i \prod_j^{i-1}(1-\boldsymbol{\alpha}_j), \tag{1}$$

where $\boldsymbol{c}_i$ and $\boldsymbol{\alpha}_i$ denote the color and opacity of the Gaussian, and $\boldsymbol{d}_i$ is the z-axis of the points by projecting the center of 3D Gaussians $\boldsymbol{\mu}$ to the camera coordinate.

Crossmodal Progressive Scene Generation

Previous methods (Wang et al. 2024) have made progress in the single-object generation, but it is difficult to ensure texture and structural coherence in complex outward-facing viewpoint scene generation.To realize crossmodal 3D scene generation, we propose a crossmodal Progressive Scene Generation (PSG) framework to incrementally generate high-quality scenes with reference to previous work (Ouyang et al. 2023). The main workflow of the proposed PSG is shown in Figure 1, which mainly consists of two phases: point cloud construction and supported view generation.

Point cloud construction. Given a text prompt y, our goal is to generate 3D scenes that match y in a crossmodal manner. We use a text-conditioned image inpainting model $F_{inpaint}$ and a monocular depth estimator F_d to progressively inpaint and update the scene. The pre-trained text-to-image diffusion model F_{t2i} is used to generate the initial image $\boldsymbol{I}_0$ from text prompt y. The user can also specify the initial image. At this point, the pre-trained image-to-text generation model F_{i2t} is used to generate a suitable caption y. F_d is then used to obtain the depth map $\boldsymbol{D}_0$ from $\boldsymbol{I}_0$.

The predefined cameras $\{\boldsymbol{C}_i\}_{i=0}^N$ are denoted by the extrinsic parameters $\boldsymbol{E}_i \in \mathbb{R}^{3\times4}$ and the shared intrinsic parameter $\boldsymbol{K} \in \mathbb{R}^{3\times3}$, where N denotes the number of cameras. Based on the initial camera $\boldsymbol{C}_0$, 2D pixels are lifted to 3D space to construct the initial point cloud $\boldsymbol{P}_0$ through a series of geometric transformations $f_{2\to3}$:

$$\boldsymbol{P}_0 = f_{2\to3}(\boldsymbol{I}_0, \boldsymbol{D}_0, \boldsymbol{E}_0, \boldsymbol{K}). \tag{2}$$

After obtaining the initial view's point cloud P_0, additional point clouds need to be merged into the existing ones at each camera pose. Specifically, at the i^{th} camera, the existing 3D point cloud $\boldsymbol{P}_{i-1}$ is projected into 2D space through a series of geometric transformations $f_{3\to2}$. Due to changes in camera pose, this projection produces a partial image $\hat{I}_i$ and a mask $\hat{M}_i$ indicating the area for inpainting:

$$\hat{\boldsymbol{I}}_i, \hat{\boldsymbol{M}}_i = f_{3\to2}(\boldsymbol{P}_{i-1}, \boldsymbol{E}_i, \boldsymbol{K}). \tag{3}$$

The image inpainting model $F_{inpaint}$ is used to inpaint an image $\boldsymbol{I_i}$ based on $\hat{\boldsymbol{I}}_i$, $\hat{\boldsymbol{M}}_i$ and y. The monocular depth estimator F_d is used to obtain the depth map $\boldsymbol{D}_i$. Since there is some difference between the depth maps of two neighboring frames, $\boldsymbol{D}_i$ needs to be processed by minimizing the difference between the overlapping regions of the two point clouds to get the aligned depth $\boldsymbol{D}_{i,align}$:

$$\boldsymbol{D}_{i,align} = align(f_{2\to3}(\boldsymbol{D}_i), \boldsymbol{P}_{i-1}, \hat{\boldsymbol{M}}_i = 1), \tag{4}$$

where the function $align(\cdot)$ minimizes the difference between the overlapping parts ($\hat{\boldsymbol{M}}_i = 1$) of two point clouds. Then the inpainted pixels of I_i need to be lifted to 3D space. The updated point cloud $\boldsymbol{P}_i$ can be defined as follows:

$$\boldsymbol{P}_i = update(\boldsymbol{P}_{i-1}, \hat{\boldsymbol{P}}_i), \tag{5}$$

$$\hat{\boldsymbol{P}}_i = f_{2\to3}(\boldsymbol{I}_i, \boldsymbol{D}_{i,align}, \boldsymbol{E}_i, \boldsymbol{K}, \hat{\boldsymbol{M}}_i = 0), \tag{6}$$

where the function $update(\cdot)$ merges the new point cloud $\hat{\boldsymbol{P}}_i$ into the existing point cloud $\boldsymbol{P}_{i-1}$, $\hat{\boldsymbol{M}}_i = 0$ means only the inpainted pixels will be lifted. The above steps are repeated N times to obtain the final point cloud $\boldsymbol{P}_N$.

Supported View Generation. After creating the final point cloud $\boldsymbol{P}_N$, we use $\boldsymbol{P}_N$ as the initial SfM (Schonberger and Frahm 2016) points to initialize 3DGS. Since the initial $(N+1)$ RGBD images are not sufficient to train the 3DGS to produce reasonable outputs, we choose to add additional M support images to form the 3DGS training set $\boldsymbol{I}_{i=0}^{N+M}$ and $\boldsymbol{D}_{i=0}^{N+M}$. We take the depth of the center of the initial depth map D_0 as the radius of the sphere. The cameras are shifted $\pm5^\circ$ along the sphere to get new cameras $\{\boldsymbol{C}_i\}_{i=N+1}^{N+M}$. The 3DGS training sets $\boldsymbol{I}_{i=0}^{N+M}$ and $\boldsymbol{D}_{i=0}^{N+M}$ are obtained by reprojection from $\boldsymbol{P}_N$ using $\{\boldsymbol{C}_i\}_{i=0}^{N+M}$:

$$\boldsymbol{D}_i, \boldsymbol{I}_i, \hat{\boldsymbol{M}}_i = f_{3\to2}(\boldsymbol{P}_N, \boldsymbol{E}_i, \boldsymbol{K}), \tag{7}$$

where $i \in \{0, ..., N+M\}$. When optimizing 3DGS, we only consider the valid image regions ($\hat{\boldsymbol{M}}_i = 1$) for the support images $\boldsymbol{I}_{i=N+1}^{N+M}$ and $\boldsymbol{D}_{i=N+1}^{N+M}$ to prevent 3DGS from learning the erroneous details of reprojection.

We propose hierarchical Depth Prior-based Regularization (DPR) and Structured Context-guided Compression (SCC) to optimize the quality of 3DGS-generated scenes and reduce storage space. The detailed descriptions are stated in the subsequent sections.

Hierarchical Depth Prior-based Regularization

3DGS represents the scene more realistically through numerous 3D Gaussians with geometric and appearance attributes. The scenes generated by 3DGS in the progressive scene generation framework tend to be ambiguous and artifactual since the scene contains millions of attributes of

Gaussian distributions optimized only via gradient descent based on photometric loss. Previous work (Yuan et al. 2024; Li et al. 2024) utilizes score distillation to achieve 3D scenes with consistency, which improves the quality of novel view synthesis to some extent. Despite their progress, some limitations remain: (1) lack of precise constraints on 3D cues and depth information in the optimization process. (2) Neglecting effective supervision of the visual and geometric smoothness of the scene. The above issues limit the realism and continuity of 3D scene generation. To this end, we propose a hierarchical Depth Prior-based Regularization (DPR) mechanism that implements multi-level regularization on the 3D Gaussians utilizing high-quality depth prior. Specifically, we implement joint constraints on the depth maps generated by 3DGS at the pixel level and distribution level by utilizing the Huber loss and Central Moment Discrepancy (CMD), respectively. Furthermore, the bilateral filter is leveraged to enhance the continuity of the depth information. Consequently, DPR generates high-quality 3D scenes that are elaborate and multi-view consistent.

The depth map $\boldsymbol{D}$ is obtained by the Equation (7). 3DGS estimates the z-depth map $\hat{\boldsymbol{D}}$ of all pixel by the Equation (1).

Depth estimation accuracy constraints. We utilize a multi-scale constraint paradigm at the pixel level and distribution level to achieve accurate estimation of depth information. The depth of object edges is often difficult to estimate and inaccurate in depth maps. The edges of objects tend to be regions with large image gradients. Thus, to apply more attention to the edges, we design a gradient-aware Huber-based depth loss for implementing pixel-level depth constraints and adaptive depth regularization, denoted as follows:

$$\mathcal{L}_{DPR}^{pixel} = \begin{cases} g_{\text{rgb}} \frac{1}{|\hat{\boldsymbol{D}}|} \sum \|\boldsymbol{D} - \hat{\boldsymbol{D}}\|_1, \text{ if } \|\boldsymbol{D} - \hat{\boldsymbol{D}}\|_1 > \delta \\ g_{\text{rgb}} \frac{1}{|\boldsymbol{D}|} \sum \frac{(\boldsymbol{D} - \hat{\boldsymbol{D}})^2 + \delta^2}{2\delta}, \text{ otherwise} \end{cases}, \tag{8}$$

where $g_{rgb} = exp(-\nabla)$ and ∇ is the gradient of the current aligned RGB image, $\delta = 0.2 \max \|\boldsymbol{D} - \hat{\boldsymbol{D}}\|_1$ and $|\hat{\boldsymbol{D}}|$ indicates the total number of pixels in $\hat{\boldsymbol{D}}$. Image edges with larger gradients are dynamically assigned smaller learning weights. Constraining two depth maps only at the pixel level ignores the discrepancy of their distributions. Therefore, we implement distribution-level alignment between depth maps based on Central Moment Discrepancy (CMD), which has been widely used in domain adaptation to estimate the discrepancy between two domains (Zellinger et al. 2019). CMD can utilize higher-order moments to effectively capture higher-order statistical information without kernel function dependence. Let X and Y be bounded random samples with respective probability distributions p and q on the interval $[a, b]^N$. The central moment discrepancy $\mathcal{D}_{CMD}^K$ is defined as an empirical estimate of the CMD metric:

$$\begin{aligned} \mathcal{D}_{CMD}^K(\boldsymbol{X}, \boldsymbol{Y}) = & \frac{1}{|b-a|} \|\mathbf{E}(\boldsymbol{X}) - \mathbf{E}(\boldsymbol{Y})\|_2 \\ & + \sum_{k=2}^{K} \frac{1}{|b-a|^k} \|C_k(\boldsymbol{X}) - C_k(\boldsymbol{Y})\|_2, \end{aligned} \tag{9}$$

where $\mathrm{E}(\boldsymbol{X}) = \frac{1}{|\boldsymbol{X}|} \sum_{x \in \boldsymbol{X}} x$ is the empirical expectation vector of sample X and $C_k(X) = \mathrm{E}\left((x - \mathrm{E}(\boldsymbol{X}))^k\right)$ is the vector of all k^{th} order sample central moments of the coordinates of $\boldsymbol{X}$. The CMD-based depth loss is expressed as:

$$\mathcal{L}_{DPR}^{dist} = \mathcal{D}_{CMD}^K(\boldsymbol{D}, \hat{\boldsymbol{D}}). \tag{10}$$

Depth smoothness constraints. To address the problem that object boundaries in 3DGS-rendered images often appear to have nonsmooth edges, we propose a depth loss based on the bilateral filter (Tomasi and Manduchi 1998). Bilateral filtering is a typical nonlinear filtering method that simultaneously considers both the space and value domain information, allowing the removal of depth noise while preserving the boundaries and details of the image. Given two pixels $\boldsymbol{p}$ and $\boldsymbol{q}$ in the depth map with coordinates (i, j) and (m, n) respectively. The spatial kernel and color kernel of bilateral filtering are denoted as:

$$\mathcal{L}_{DPR}^{smooth} = \frac{1}{|\boldsymbol{\mathcal{N}}(\boldsymbol{p})|} \sum_{\boldsymbol{q} \in \boldsymbol{\mathcal{N}}(\boldsymbol{p})} \boldsymbol{\mathcal{G}}_s(\boldsymbol{p}, \boldsymbol{q}) \cdot \boldsymbol{\mathcal{G}}_r(\boldsymbol{p}, \boldsymbol{q}) \cdot (\hat{\boldsymbol{D}}_{\boldsymbol{p}} - \hat{\boldsymbol{D}}_{\boldsymbol{q}})^2, \tag{11}$$

where $|\boldsymbol{\mathcal{N}}(\boldsymbol{p})|$ is the number of pixels in the neighborhood of pixel $\boldsymbol{p}$, $\hat{\boldsymbol{D}}_{\boldsymbol{p}}$ is the depth value at pixel $\boldsymbol{p}$, spatial kernel $\boldsymbol{\mathcal{G}}_s(\boldsymbol{p}, \boldsymbol{q}) = \exp(-\frac{(i-m)^2 + (j-n)^2}{2\sigma_s^2})$, and color kernel $\boldsymbol{\mathcal{G}}_r(\boldsymbol{p}, \boldsymbol{q}) = \exp(-\frac{\|\hat{\boldsymbol{D}}(\boldsymbol{p}) - \hat{\boldsymbol{D}}(\boldsymbol{q})\|^2}{2\sigma_r^2})$.

Consequently, the loss of DRP is expressed as:

$$\mathcal{L}_{DPR} = \lambda_1 \mathcal{L}_{DPR}^{pixcel} + \lambda_2 \mathcal{L}_{DPR}^{dist} + \lambda_3 \mathcal{L}_{DPR}^{smooth}. \tag{12}$$

The trade-off hyperparameters λ_1, λ_2, and λ_3 are set to 0.7, 0.1 and 1.0 respectively.

Structured Context-guided Compression

The microscopic 3D Gaussians with optimizable geometric and appearance attributes in 3DGS make it a powerful advantage for rendering a variety of scenes. Nevertheless, a complex and larger-scale scene often requires a prohibitively large number of 3D Gaussians for fine-grained representation, resulting in significant storage overhead. Furthermore, in real-world applications, low-cost and lightweight models are more conducive to deployment and rapid scene generation. Due to the unorganized and sparse properties of 3D Gaussians (Chen and Wang 2024), compressing 3D Gaussians is a challenging task. Mainstream 3DGS compression methods mostly focus only on the "values" (Fan et al. 2023b; Navaneet et al. 2023), ignoring the structural correlation between their 3D Gaussians, resulting in a large amount of structural redundancy and inefficient compression. Scaffold-GS (Lu et al. 2024b) introduces anchors to cluster nearby relevant 3D Gaussians and utilizes the anchors' properties to predict the 3D Gaussians' properties. Although Scaffold-GS exploits the spatial correlations among 3D Gaussians, the independence of anchors leads to a large number of sparse and disordered anchors that are difficult to compress. In order to take full advantage of the correlation between unorganized anchors, we propose a Structured Context-guided Compression (SCC) mechanism based on Scaffold-GS that utilizes a structured hash feature mesh to model the context of the anchor attributes, thus further compressing the Scaffold-GS.

In Scaffold-GS, each anchor is composed of a location $\boldsymbol{x}^a \in \mathbb{R}^3$ and an anchor attribute $\mathcal{A} = \{\boldsymbol{f}^a \in \mathbb{R}^{D^a}, \boldsymbol{l} \in \mathbb{R}^6, \boldsymbol{o} \in \mathbb{R}^{3K}\}$, where each component represents anchor feature, scaling, and offsets, respectively. During the rendering phase, the anchor feature is fed into the MLPs to generate attributes for 3D Gaussians, whose locations are determined by adding $\boldsymbol{x}_a$ and $\boldsymbol{o}$, where $\boldsymbol{l}$ is utilized to regularize both locations and shapes of the Gaussians. The attributes inferred from the anchor attributes by neighboring 3D Gaussians should be similar. Thus, we utilize a structured hash grid to model the inherent spatial consistency of independent anchors. The core idea is to replace the anchor feature $\boldsymbol{f}^a$ with the feature $\boldsymbol{f}^h$ obtained by implementing the trilinear interpolation in the hash grid. Assuming that the two features have a strong correlation, we try to use the hash feature to model the context of the anchor attribute $\boldsymbol{\mathcal{A}}$, denoted as conditional probability:

$$\begin{aligned} p\left(\mathcal{A}, \boldsymbol{x}^a, \mathcal{H}\right) &= p\left(\mathcal{A} \mid \boldsymbol{x}^a, \mathcal{H}\right) \times p\left(\boldsymbol{x}^a, \mathcal{H}\right) \\ &\sim p(\mathcal{A} \mid \boldsymbol{f}^h) \times p(\mathcal{H}). \end{aligned} \tag{13}$$

According to information theory (Cover 1999), the higher the probability, the lower the entropy and the lower the bit consumption. Therefore, the main goal of SCC is to minimize the entropy of the anchor attribute $\boldsymbol{\mathcal{A}}$ with the help of the hash feature $\boldsymbol{f}^h$.

Feature quantization strategy. To facilitate entropy coding, the values of $\boldsymbol{\mathcal{A}}$ must be quantized into a finite set. Following (Ballé et al. 2018), we propose a Feature Quantization Strategy (FQS) that uses "noise addition" and "rounding" operations in the training and testing phases, respectively. for the i-th anchor $\boldsymbol{x}_i^a$, we denote $\boldsymbol{f}_i$ as any of its $\boldsymbol{\mathcal{A}}_i$'s components: $\boldsymbol{f}_i \in \{\boldsymbol{f}_i^a, \boldsymbol{l}_i, \boldsymbol{o}_i\} \in \mathbb{R}^D$, where $D \in \{D_a, 6, 3K\}$ is its respective dimension. The FQS process is represented as:

$$\begin{aligned} \hat{\boldsymbol{f}}_i &= \boldsymbol{f}_i + \mathcal{U}(-\frac{1}{2}, \frac{1}{2}) \times \boldsymbol{q}_i, \quad \text{for training} \\ &= \operatorname{Round}\left(\boldsymbol{f}_i / \boldsymbol{q}_i\right) \times \boldsymbol{q}_i, \quad \text{for testing} \end{aligned} \tag{14}$$

where $\boldsymbol{q}_i = Q_0 \times (1 + \operatorname{Tanh}(\boldsymbol{r}_i))$ and $\boldsymbol{r}_i = \mathrm{MLP}_{\mathrm{q}}(\boldsymbol{f}_i^h)$. We input $\boldsymbol{f}^h$ into MLP to obtain the factor r for dynamically adjusting the predefined Q. Obviously, the quantization step q is in the range of $(0, 2Q_0)$, which makes $\hat{\boldsymbol{f}}_i$ very close to the original feature $\boldsymbol{f}_i$ and maintains a high fidelity.

Gaussian Distribution Modeling. To measure and reduce the bit consumption of $\boldsymbol{f}_i$ during training, we need to estimate its probability in a microscopic manner. All three attributes of the anchors exhibit statistical tendencies of Gaussian distributions(Lu et al. 2024b). Thus, based on the independence of the anchor attributes, we construct Gaussian distributions for all anchor attributes, with μ and σ in the respective distributions estimated by a contextual modeling module from f^h. The probability of $\hat{\boldsymbol{f}}_i$ is computed as:

$$\begin{aligned} p\left(\hat{\boldsymbol{f}}_i\right) &= \int_{\hat{\boldsymbol{f}}_i - \frac{1}{2}\boldsymbol{q}_i}^{\hat{\boldsymbol{f}}_i + \frac{1}{2}\boldsymbol{q}_i} \phi_{\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i}(x) dx \\ &= \Phi_{\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i}\left(\hat{\boldsymbol{f}}_i + \frac{1}{2}\boldsymbol{q}_i\right) - \Phi_{\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i}\left(\hat{\boldsymbol{f}}_i - \frac{1}{2}\boldsymbol{q}_i\right) \\ \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i &= \mathrm{MLP}_{\mathrm{c}}\left(\boldsymbol{f}_i^h\right). \end{aligned} \tag{15}$$

where ϕ and Φ represent the probability density function and the cumulative distribution function, respectively. Ultimately, we define the entropy loss as the sum of the bit consumption of all $\hat{\boldsymbol{f}}_i$:

$$\mathcal{L}_{entropy} = \frac{1}{N(D^a + 6 + 3K)} \sum_{\boldsymbol{f} \in \{\boldsymbol{f}^a, \boldsymbol{l}, \boldsymbol{o}\}} \sum_{i=1}^{N} \sum_{j=1}^{D} \left(-\log_2 p\left(\hat{\boldsymbol{f}}_{i,j}\right)\right), \tag{16}$$

where N is the number of anchors and $\boldsymbol{f}_{i,j}$ means the j-th dimension value of $\boldsymbol{f}_i$. Minimizing the entropy loss achieves a high probability estimation of $p(\boldsymbol{f}_i)$ that guides the learning of the contextual model. The SCC loss is denoted as:

$$\mathcal{L}_{SCC} = \mathcal{L}_{Scaffold} + \mathcal{L}_{entropy}. \tag{17}$$

where $\mathcal{L}_{Scaffold}$ represents the rendering loss as defined in (Lu et al. 2024a), which includes two fidelity penalty loss terms and one regularization term for the scaling $\boldsymbol{l}$.

Optimization Objectives

The final loss we use for optimization is defined as follows:

$$\mathcal{L} = \mathcal{L}_{RGB} + \mathcal{L}_{DPR} + \mathcal{L}_{SCC}, \tag{18}$$

where $\mathcal{L}_{RGB}$ is the original photometric loss proposed in (Kerbl et al. 2023).

Experiments

Datasets and Evaluation Metrics

BloomScene is optimized for each input without requiring training datasets. Due to the lack of 3D scene generated related to text prompt as a reference, previous reference-based metrics (e.g., PSNR and LPIPS (Zhang et al. 2018)) are unsuitable for this generation task. Therefore, we used six 2D metrics to assess the quality of the generated scenes comprehensively. We use BRISQUE (Mittal, Moorthy, and Bovik 2012) and NIQE (Mittal, Soundararajan, and Bovik 2012) for reference-free image quality assessment, and CLIP score (Hessel et al. 2021) to measure how well the rendered images are aligned to the input text prompt. Moreover, the colorful, quality, and sharp metrics of CLIP-IQA (Wang, Chan, and Loy 2023) are used to assess the appearance and feel of the image in a way that is closer to human perception.

Implementation Details

To maximize the generalization ability of the network, we employ pre-trained large-scale models to build the entire network architecture. Specifically, we use Stable Diffusion V1.5 (Rombach et al. 2022) as the text-to-image generation

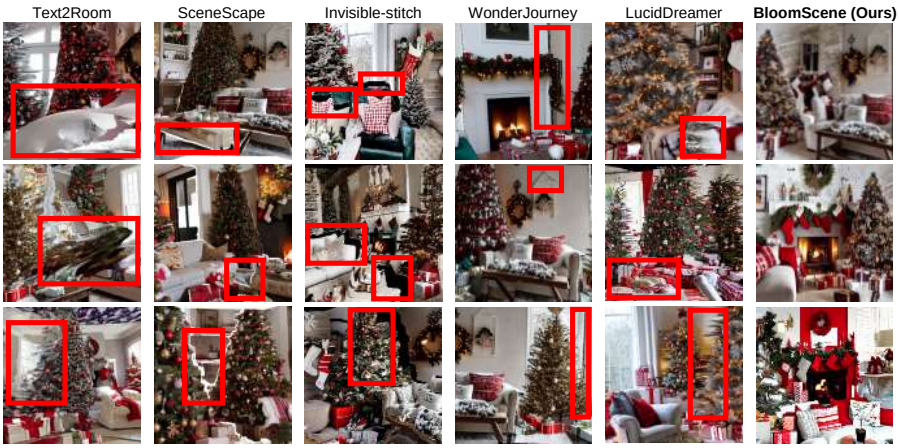

Figure 2: **Qualitative comparison of our method and baselines.** "A cozy living room in Christmas"

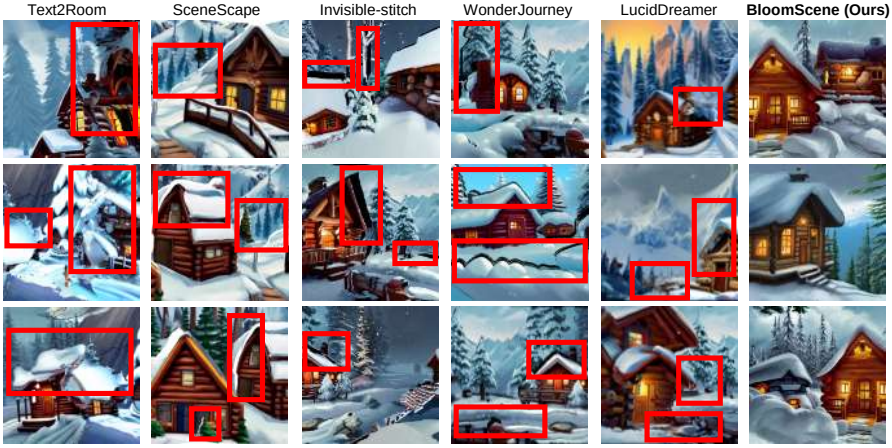

Figure 3: **Qualitative comparison of our method and baselines.** "A small cabin on top of a snowy mountain, Disney style"

model. The text-conditioned image inpainting model (Rombach et al. 2022) is used to inpaint the masked images. If the input is an image without text, LLaVa (Contributors 2023) is used for image-to-text pairing for the image inpainting model. We use ZoeDepth (Bhat et al. 2023) as the monocular depth estimator. To generate visual scenes, we move the camera with a rotation of 0.63 radians. We use text prompts describing indoor, outdoor, and artistic scenes. All experiments are done on a single NVIDIA A800 GPU. All experimental results are averaged over multiple experiments using five different random seeds.

Comparison with State-of-the-Art Methods

We compare the proposed BloomScene with five representative and reproducible methods, including progressive 3D scene generation methods: text2Room (Höllein et al. 2023), 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). We use the open-source codebase of the above models and modify the inputs to start from the same initial images and text prompts.

Qualitative Results We perform an intuitive qualitative analysis. We show the RGB rendering results of our method and baseline methods in new viewpoints in Figure 2 and Figure 3. We have the following observations: (i) SceneScape, WonderJourney, and Invisible-stitch generate relatively complete scene content, but clear breaks and geometric distortions can be observed in boxed areas. (ii) Text2Room uses a polygonal mesh to represent the scene, but its mesh fusion threshold filtering scheme results in incomplete detection of stretched regions. This leads to a large number of distorted and oversmoothed regions in the scene. (iii) LucidDreamer is currently the more visually appealing method for progressive scene generation but suffers from artifacts and geometric distortions in boxed areas. (iv) In contrast, our method preserves the necessary scene structures, significantly reduces artifacts and geometric distortions, and provides high-quality and realistic rendering results.

Quantitative Results Table 1 shows the average quantitative results for several scenes. We can conclude the following points: (i) Overall, our method generates higher quality 3D scenes with significantly reduced storage overhead, which is significantly better than the baseline models. (ii) The storage overhead of our generated scenes is 6.9x and 9.2x lower than Invisible-stitch (Engstler et al. 2024) and

Models	Size (MB) ↓	CLIP-Score ↑	CLIP-IQA ↑			BRISQUE ↓	NIQE ↓
			Quality	Colorful	Sharp		
Text2Room (Höllein et al. 2023)	204.41	29.45	0.60	0.77	0.34	27.24	3.43
SceneScape (Fridman et al. 2024)	189.00	30.97	0.56	0.76	0.32	31.98	3.95
Invisible-stitch (Engstler et al. 2024)	430.55	31.16	0.63	0.68	0.41	26.19	3.56
WonderJourney (Yu et al. 2024)	145.95	30.76	0.58	0.77	0.38	27.46	3.47
LucidDreamer (Chung et al. 2023)	571.63	31.19	0.66	0.77	0.42	24.07	3.05
BloomScene (Ours)	**62.36**	**31.94**	**0.72**	**0.79**	**0.45**	**20.40**	**2.91**

Table 1: Performance comparison among the proposed framework and baselines. Our approach achieves the best results.

Models	Size (MB) ↓	CLIP-Score ↑	CLIP-IQA ↑			BRISQUE ↓	NIQE ↓
			Quality	Colorful	Sharp		
w/o DPR	64.48	31.54	0.69	0.77	0.40	22.40	3.03
w/o SCC	569.33	31.76	0.70	0.78	0.44	22.71	2.96
w/o $\mathcal{L}_{DPR}^{smooth}$	64.24	31.84	0.69	0.78	0.42	20.62	2.93
w/o $\mathcal{L}_{DPR}^{dist}$	63.92	31.89	0.69	0.78	0.42	21.01	2.93
w/o $\mathcal{L}_{DPR}^{pixel}$	64.33	31.70	0.68	0.77	0.42	20.82	2.95
BloomScene (Full)	**62.36**	**31.94**	**0.72**	**0.79**	**0.45**	**20.40**	**2.91**

Table 2: Ablation results of different components.

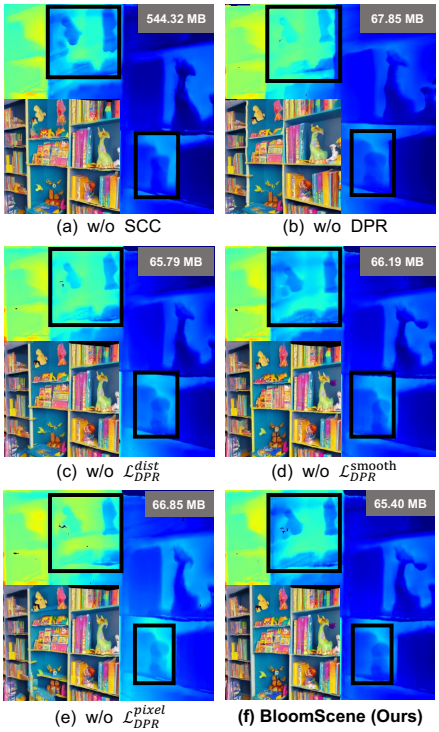

Figure 4: Visualization of ablation results.

LucidDreamer (Chung et al. 2023) using 3DGS. It is also significantly reduced compared to Text2Room (Höllein et al. 2023) and SceneScape (Fridman et al. 2024) using mesh or WonderJourney (Yu et al. 2024) using point clouds. (iii) Our BRISQUE and NIQE scores are 20.40 and 2.91, which are 15.3% and 4.6% lower compared to the optimal scores. Additionally, the best performance is achieved in the CLIP-Score and CLIP-IQA metrics. It is demonstrated that the key components effectively utilize the scene geometry information to reduce the distortion of the rendered images and enhance their alignment with the text prompts.

Ablation Studies

To verify the necessity of the different components, we performed comprehensive ablation experiments using the same set of text prompts. The results are shown in Figure 4 and Table 2. (i) Firstly, DPR is removed from BloomScene. The decreased performance and worse depth rendering result indicate that effective supervision of depth information and smoothness during optimization is crucial in the realism and continuity of 3D scenes. (ii) Moreover, we replace SCC with the original 3DGS. The dramatic increase in scene storage overhead indicates that compression for complex and larger-scale scenes is very necessary. (iii) Eventually, we remove the loss terms from DPR. The degraded and worse performance in depth map smoothness and accuracy indicate that the various loss items of DPR are necessary.

Conclusion

In this paper, we propose BloomScene, a lightweight structured 3D Gaussian splatting for crossmodal scene generation, which creates diverse and high-quality 3D scenes from text or image inputs. Specifically, a crossmodal progressive scene generation framework is proposed to incrementally generate coherent scenes. Furthermore, we propose a hierarchical depth prior-based regularization mechanism that utilizes multi-level constraints on depth accuracy and smoothness to enhance the realism and continuity of the generated scenes. Finally, we propose a structured context-guided compression mechanism that utilizes structured hash grids to model the context of unorganized anchor attributes, thus significantly reducing storage overhead. Comprehensive qualitative and quantitative experiments across multiple scenarios show that the proposed framework has significant advantages over several baselines. Our framework opens up more possibilities for future virtual reality applications.

References

Ballé, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5470–5479.

Berger, M.; Tagliasacchi, A.; Seversky, L. M.; Alliez, P.; Levine, J. A.; Sharf, A.; and Silva, C. T. 2014. State of the art in surface reconstruction from point clouds. In *35th Annual Conference of the European Association for Computer Graphics, Eurographics 2014-State of the Art Reports*. The Eurographics Association.

Bhat, S. F.; Birkl, R.; Wofk, D.; Wonka, P.; and Müller, M. 2023. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*.

Chen, G.; and Wang, W. 2024. A survey on 3d gaussian splatting. *arXiv preprint arXiv:2401.03890*.

Chung, J.; Lee, S.; Nam, H.; Lee, J.; and Lee, K. M. 2023. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*.

Contributors, X. 2023. XTuner: A Toolkit for Efficiently Fine-tuning LLM. https://github.com/InternLM/xtuner.

Cover, T. M. 1999. *Elements of information theory*. John Wiley & Sons.

Engstler, P.; Vedaldi, A.; Laina, I.; and Rupprecht, C. 2024. Invisible Stitch: Generating Smooth 3D Scenes with Depth Inpainting. *arXiv preprint arXiv:2404.19758*.

Fan, Z.; Wang, K.; Wen, K.; Zhu, Z.; Xu, D.; and Wang, Z. 2023a. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *arXiv preprint arXiv:2311.17245*.

Fan, Z.; Wang, K.; Wen, K.; Zhu, Z.; Xu, D.; and Wang, Z. 2023b. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *arXiv preprint arXiv:2311.17245*.

Fridman, R.; Abecasis, A.; Kasten, Y.; and Dekel, T. 2024. Scenescape: Text-driven consistent scene generation. *Advances in Neural Information Processing Systems*, 36.

Hessel, J.; Holtzman, A.; Forbes, M.; Bras, R. L.; and Choi, Y. 2021. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*.

Höllein, L.; Cao, A.; Owens, A.; Johnson, J.; and Nießner, M. 2023. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7909–7920.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Lee, H.-H.; and Chang, A. X. 2022. Understanding pure clip guidance for voxel grid nerf models. *arXiv preprint arXiv:2209.15172*.

Li, H.; Shi, H.; Zhang, W.; Wu, W.; Liao, Y.; Wang, L.; Lee, L.-h.; and Zhou, P. 2024. DreamScene: 3D Gaussian-based Text-to-3D Scene Generation via Formation Pattern Sampling. *arXiv preprint arXiv:2404.03575*.

Lin, C.-H.; Gao, J.; Tang, L.; Takikawa, T.; Zeng, X.; Huang, X.; Kreis, K.; Fidler, S.; Liu, M.-Y.; and Lin, T.-Y. 2023. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 300–309.

Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; and Dai, B. 2024a. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20654–20664.

Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; and Dai, B. 2024b. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20654–20664.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Mittal, A.; Moorthy, A. K.; and Bovik, A. C. 2012. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12): 4695–4708.

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

Mohammad Khalid, N.; Xie, T.; Belilovsky, E.; and Popa, T. 2022. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 conference papers*, 1–8.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4): 1–15.

Munkberg, J.; Hasselgren, J.; Shen, T.; Gao, J.; Chen, W.; Evans, A.; Müller, T.; and Fidler, S. 2022. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8280–8290.

Navaneet, K.; Meibodi, K. P.; Koohpayegani, S. A.; and Pirsiavash, H. 2023. Compact3d: Compressing gaussian splat radiance field models with vector quantization. *arXiv preprint arXiv:2311.18159*.

Ouyang, H.; Heal, K.; Lombardi, S.; and Sun, T. 2023. Text2immersion: Generative immersive scene with 3d gaussians. *arXiv preprint arXiv:2312.09242*.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.;

et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Ranftl, R.; Lasinger, K.; Hafner, D.; Schindler, K.; and Koltun, V. 2020. 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.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.

Snavely, N.; Seitz, S. M.; and Szeliski, R. 2006. Photo tourism: exploring photo collections in 3D. In *ACM siggraph 2006 papers*, 835–846.

Tang, J.; Ren, J.; Zhou, H.; Liu, Z.; and Zeng, G. 2023. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*.

Tomasi, C.; and Manduchi, R. 1998. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, 839–846. IEEE.

Wang, J.; Chan, K. C.; and Loy, C. C. 2023. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2555–2563.

Wang, Z.; Lu, C.; Wang, Y.; Bao, F.; Li, C.; Su, H.; and Zhu, J. 2024. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36.

Yu, H.-X.; Duan, H.; Hur, J.; Sargent, K.; Rubinstein, M.; Freeman, W. T.; Cole, F.; Sun, D.; Snavely, N.; Wu, J.; et al. 2024. Wonderjourney: Going from anywhere to everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6658–6667.

Yuan, X.; Yang, H.; Zhao, Y.; and Huang, D. 2024. DreamScape: 3D Scene Creation via Gaussian Splatting joint Correlation Modeling. *arXiv preprint arXiv:2404.09227*.

Zellinger, W.; Moser, B. A.; Grubinger, T.; Lughofer, E.; Natschläger, T.; and Saminger-Platz, S. 2019. Robust unsupervised domain adaptation for neural networks via moment alignment. *Information Sciences*, 483: 174–191.

Zhang, J.; Li, X.; Wan, Z.; Wang, C.; and Liao, J. 2024. Text2nerf: Text-driven 3d scene generation with neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Reproducibility Checklist

This paper:

- Includes a conceptual outline and/or pseudocode description of AI methods introduced (**yes**)
- Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (**yes**)
- Provides well marked pedagogical references for less-familiare readers to gain background necessary to replicate the paper (**yes**)

Does this paper make theoretical contributions? (**yes**)
If yes, please complete the list below.

- All assumptions and restrictions are stated clearly and formally. (**yes**)
- All novel claims are stated formally (e.g., in theorem statements). (**yes**)
- Proofs of all novel claims are included. (**yes**)
- Proof sketches or intuitions are given for complex and/or novel results. (**partial**)
- Appropriate citations to theoretical tools used are given. (**yes**)
- All theoretical claims are demonstrated empirically to hold. (**yes**)
- All experimental code used to eliminate or disprove claims is included. (**NA**)

Does this paper rely on one or more datasets? (**yes**)
If yes, please complete the list below.

- A motivation is given for why the experiments are conducted on the selected datasets. (**yes**)
- All novel datasets introduced in this paper are included in a data appendix. (**NA**)
- All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (**NA**)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. (**yes**)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. (**yes**)
- All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing. (**NA**)

Does this paper include computational experiments? (**yes**)
If yes, please complete the list below.

- Any code required for pre-processing data is included in the appendix. (**yes**).
- All source code required for conducting and analyzing the experiments is included in a code appendix. (**yes**)
- All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (**yes**)

- All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from. (**partial**)
- If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. (**NA**)
- This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. (**yes**)
- This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. (**yes**)
- This paper states the number of algorithm runs used to compute each reported result. (**yes**)
- Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. (**yes**)
- The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). (**yes**)
- This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. (**yes**)
- This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. (**no**)