

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

Xiaolu Hou^{1*}, Mingcheng Li^{1*}, Dingkang Yang^{1†}, Jiawei Chen¹, Ziyun Qian¹, Xiao Zhao¹, Yue Jiang¹, Jinjie Wei¹, Qingyao Xu¹, Lihua Zhang^{1,2,3,4,5†}

¹ Academy for Engineering and Technology, Fudan University

² Institute of Metaverse & Intelligent Medicine, Fudan University

³ Engineering Research Center of AI and Robotics, Ministry of Education

⁴ Jilin Provincial Key Laboratory of Intelligence Science and Engineering

⁵ Artificial Intelligence and Unmanned Systems Engineering Research Center of Jilin Province

xlh23@m.fudan.edu.cn, mingchengli21@m.fudan.edu.cn

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.

Code — <https://github.com/SparklingH/BloomScene>

Introduction

Currently, there is a growing demand for 3D content in virtual reality. However, creating 3D content is time-consuming and 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.

*These authors contributed equally.

†Corresponding authors.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

However, the shortage of annotated 3D 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) extract 2D priors from diffusion models through a time-consuming distillation process to optimize the generation of 3D content. However, these methods (Wang et al. 2024) have limitations when extended to fine-grained scenes with outward-facing viewpoints. Therefore, several methods (Höller et al. 2023; Ouyang et al. 2023; Hou et al. 2024) 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 3D scene generation.

Some methods (Höller et al. 2023; Fridman et al. 2024) generate 3D indoor scenes represented by mesh using a progressive framework but are prone to distorted or over-smoothing regions when applied to outdoor scenes. With the wide application of NeRF (Mildenhall et al. 2021) in novel view synthesis tasks, Text2NeRF (Zhang et al. 2024) generates 3D scenes represented by NeRF with a progressive framework. Although this method can generate high-quality scenes, the generation time is still quite 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 a progressive generation framework that follows the optimization goals of 3DGS to achieve domain-free 3D scene generation. Although previous 3DGS-based approaches have made some progress in 3D scene generation, they still suffer from the following limitations: (i) Rely only on photometric loss in the scene optimization process, lack sufficient regularization techniques, and are prone to artifacts and ambiguities. (ii) 3DGS requires millions of 3D Gaussians to represent each scene, resulting in high memory requirements, increasing storage costs, and end-device burden.

To address the above problems, we propose BloomScene, a lightweight structured 3D Gaussian splatting for high-quality crossmodal 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.

Related Work

Crossmodal 3D Scene Generation. Generating 3D content through language enables users to realize their demands without modeling skills. Existing methods (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 the prior knowledge of pre-trained models (Radford et al. 2021; Rombach et al. 2022). While progress has been made in the single-object generation, ensuring texture and structure consistency is still difficult when generating complex scenes with outward-facing viewpoints (Wang et al. 2024). The application of diffusion models in image inpainting has driven the development of 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 combining 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) achieve higher-quality scene generation results using 3DGS (Kerbl et al. 2023), 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 regularization of 3DGS.

Efficient 3D Scene Representation. In 3D content generation, choosing an appropriate 3D representation is crucial. The 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 representations (Mildenhall et al. 2021; Müller et al. 2022; Barron et al. 2022) improve expressiveness but still require a trade-off between speed and quality. 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) achieves fast rendering and high-quality output results, but high storage requirements impose an additional burden. For this reason, some methods focus on value (Fan et al. 2023; Navaneet et al. 2023) or structure representation (Lu et al. 2024) to reduce the computational burden. However, structural redundancy or anchor independence issues lead to lower compression efficiency. 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 μ and covariance matrix Σ , denoted as $G(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)}$. The covariance can be decomposed from a scaling matrix S and rotation matrix R , expressed as $\Sigma = R S S^T R^T$ with 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:

$$\hat{C} = \sum_i^N c_i \alpha_i \prod_j^{i-1} (1 - \alpha_j), \quad \hat{D} = \sum_i^N d_i \alpha_i \prod_j^{i-1} (1 - \alpha_j), \quad (1)$$

where c_i and α_i denote the color and opacity of the Gaussian, and d_i is the z-axis of the points by projecting the center of 3D Gaussians μ 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 when generating complex scenes with outward-facing viewpoints. To realize crossmodal 3D scene generation, we propose a crossmodal Progressive Scene Generation (PSG) framework to incrementally generate lightweight and high-quality scenes with reference to previous work (Ouyang et al. 2023; Chung et al. 2023). The main workflow of the proposed PSG is shown in Figure 1, which consists of four main phases: point cloud construction, 3D Gaussians initialization, hierarchical Depth Prior-based Regularization (DPR) mechanism to optimize the quality of the 3DGS-generated scenes, and Structured Context-guided Compression (SCC) for reducing the storage overhead. All these phases constitute the PSG framework for realizing cross-modal, high-quality 3D scenes.

Point Cloud Construction. Given a text prompt y , our goal is to generate a 3D scene that matches 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 I_0 from text prompt y . If the input is an image without a corresponding text description, the pre-trained image-to-text generation model F_{i2t} is used to generate the corresponding text prompt y from I_0 , constituting an image-text pair. F_d is then used to obtain the depth map D_0 from I_0 .

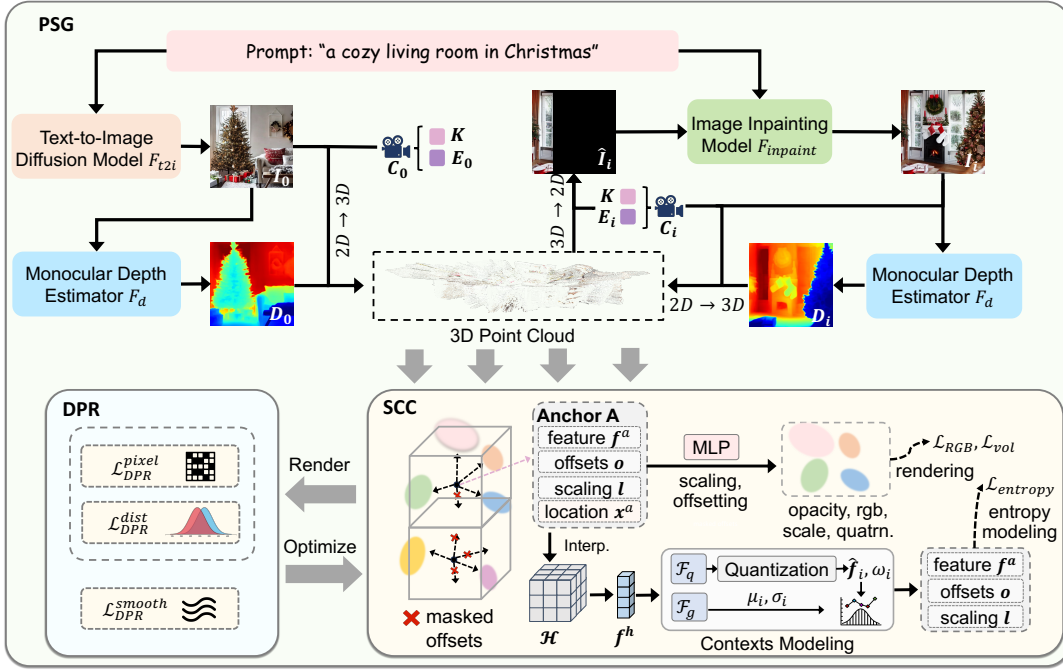


Figure 1: The overall framework of the proposed BloomScene. BloomScene utilizes the proposed crossmodal Progressive Scene Generation (PSG) framework to generate 3D scenes from the text prompts progressively. Moreover, the hierarchical Depth Prior-based Regularization (DPR) mechanism is applied to the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

Unlike previous work (Chung et al. 2023), we choose to set the camera to rotate sequentially from the initial position to both sides to minimize the cumulative error due to image inpainting during the progressive generation process. The predefined cameras $\{C_i\}_{i=0}^N$ are denoted by the extrinsic parameters $E_i \in \mathbb{R}^{3 \times 4}$ and the shared intrinsic parameter $K \in \mathbb{R}^{3 \times 3}$, where N denotes the number of cameras. Based on the initial camera C_0 , 2D pixels are transformed to 3D space to construct the initial point cloud P_0 through a series of geometric transformations \mathcal{T}_{unproj} :

$$P_0 = \mathcal{T}_{unproj}(I_0, D_0, E_0, K). \quad (2)$$

After obtaining the initial 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 P_{i-1} is projected into 2D space through a series of geometric transformations \mathcal{T}_{proj} . 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:

$$\hat{I}_i, \hat{M}_i = \mathcal{T}_{proj}(P_{i-1}, E_i, K). \quad (3)$$

Then $F_{inpaint}$ is used to generate the image I_i based on \hat{I}_i , \hat{M}_i , and y , followed by F_d to obtain the depth map D_i from I_i . Since there is some difference between the depth maps of two neighboring frames, D_i needs to be processed by minimizing the difference between the overlapping regions of the two point clouds to get the aligned depth D_i^a :

$$D_i^a = f_a(\mathcal{T}_{unproj}(I_i, D_i, E_i, K), P_{i-1}, \hat{M}_i = 1), \quad (4)$$

where the function $f_a(\cdot)$ minimizes the difference between the overlapping parts ($\hat{M}_i = 1$) of two point clouds. Then the inpainted pixels ($\hat{M}_i = 0$) of I_i need to be transformed to 3D space. The updated point cloud P_i is defined as:

$$P_i = f_u(P_{i-1}, \mathcal{T}_{unproj}(I_i, D_i^a, E_i, K), \hat{M}_i = 0), \quad (5)$$

where the function $f_u(\cdot)$ merges the new point cloud into the existing point cloud P_{i-1} . The above steps are repeated N times to obtain the final point cloud P_N .

3D Gaussians Initialization. We use P_N as the initial SfM (Schonberger and Frahm 2016) points to initialize 3DGS. Since the initial $(N + 1)$ views are not sufficient to train the 3DGS to produce reasonable outputs, we choose to add additional M support views to form the image training set $I_{i=0}^{N+M}$ of 3DGS. Unlike previous work (Chung et al. 2023), we choose to take the depth of the center of each depth map D_i as the radius of the spheres. The cameras are shifted $\pm 5^\circ$ along each sphere to get new cameras $\{C_i\}_{i=N+1}^{N+M}$. The image training set $I_{i=0}^{N+M}$ are obtained by reprojection from P_N using $\{C_i\}_{i=0}^{N+M}$:

$$I_i, \hat{M}_i = \mathcal{T}_{proj}(P_N, E_i, K), \quad (6)$$

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

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 artificial 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. In the following, the depth map \mathbf{D} is obtained by the monocular depth estimator F_d . 3DGS estimates the z-depth map $\hat{\mathbf{D}}$ 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 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:

$$\mathcal{L}_{DPR}^{pixel} = \begin{cases} g_{rgb} \frac{1}{|\hat{\mathbf{D}}|} \sum \|\mathbf{D} - \hat{\mathbf{D}}\|_1, & \text{if } \|\mathbf{D} - \hat{\mathbf{D}}\|_1 > \delta \\ g_{rgb} \frac{1}{|\hat{\mathbf{D}}|} \sum \frac{(\mathbf{D} - \hat{\mathbf{D}})^2 + \delta^2}{2\delta}, & \text{otherwise} \end{cases}, \quad (7)$$

where $g_{rgb} = \exp(-\nabla)$ and ∇ is the gradient of the current aligned RGB image, $\delta = 0.2 \max \|\mathbf{D} - \hat{\mathbf{D}}\|_1$, and $|\hat{\mathbf{D}}|$ indicates the total number of pixels in $\hat{\mathbf{D}}$. 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 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. For a random variable \mathbf{X} , the k -th central moment is given by: $\mu_k = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])^k]$, where $\mathbb{E}[\mathbf{X}]$ denotes the mean of \mathbf{X} . For two distributions \mathbf{P} and \mathbf{Q} , the CMD computes the discrepancy by summing the differences of their corresponding central moments up to order K :

$$\mathcal{D}_{CMD}^K(\mathbf{P}, \mathbf{Q}) = \sum_{k=1}^K \left\| \mu_k^{\mathbf{P}} - \mu_k^{\mathbf{Q}} \right\|_2, \quad (8)$$

where $\mu_k^{\mathbf{P}}$ and $\mu_k^{\mathbf{Q}}$ are the k -th central moments of distributions \mathbf{P} and \mathbf{Q} , respectively, and $\|\cdot\|_2$ represents the Euclidean norm. The CMD-based depth loss is expressed as:

$$\mathcal{L}_{DPR}^{dist} = \mathcal{D}_{CMD}^K(\mathbf{D}, \hat{\mathbf{D}}). \quad (9)$$

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 p and 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:

$$\mathcal{L}_{DPR}^{smooth} = \frac{1}{|\mathcal{N}(p)|} \sum_{q \in \mathcal{N}(p)} \mathcal{G}_s(p, q) \cdot \mathcal{G}_c(p, q) \cdot (\hat{\mathbf{D}}_p - \hat{\mathbf{D}}_q)^2, \quad (10)$$

where $|\mathcal{N}(p)|$ is the number of pixels in the neighborhood of pixel p , $\hat{\mathbf{D}}_p$ is the depth value at pixel p , spatial kernel is denoted as $\mathcal{G}_s(p, q) = \exp(-\frac{(i-m)^2 + (j-n)^2}{2\sigma_s^2})$, and color kernel is denoted as $\mathcal{G}_c(p, q) = \exp(-\frac{\|\mathbf{D}_p - \mathbf{D}_q\|^2}{2\sigma_c^2})$. Consequently, the loss of DPR is expressed as:

$$\mathcal{L}_{DPR} = \lambda_1 \mathcal{L}_{DPR}^{pixel} + \lambda_2 \mathcal{L}_{DPR}^{dist} + \lambda_3 \mathcal{L}_{DPR}^{smooth}, \quad (11)$$

where 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. 2023; Navaneet et al. 2023), ignoring the structural correlation between their 3D Gaussians, resulting in a large amount of structural redundancy and inefficient compression. ScaffoldGS (Lu et al. 2024) introduces anchors to cluster nearby relevant 3D Gaussians and utilizes the anchors’ properties to predict the 3D Gaussians’ properties. Although ScaffoldGS 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. HAC (Chen et al. 2025) models the relationship among the anchors to some extent, but it insufficiently quantifies

the anchors, leading to sub-optimal storage compression results. To take full advantage of the correlation between unorganized anchors, inspired by Scaffold-GS and HAC, we propose a Structured Context-guided Compression (SCC) mechanism that utilizes a structured hash feature mesh to model the context of the anchor attributes.

Description of anchors. In Scaffold-GS, each anchor is composed of a location $\mathbf{x}^a \in \mathbb{R}^3$ and an anchor attribute $\mathcal{A} = \{\mathbf{f}^a \in \mathbb{R}^{D^a}, \mathbf{l} \in \mathbb{R}^6, \mathbf{o} \in \mathbb{R}^{3K}\}$, 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 \mathbf{x}^a and \mathbf{o} , where \mathbf{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, following the methodology of HAC, we utilize a structured hash grid to model the inherent spatial consistency of independent anchors. The core idea is to use the hash feature \mathbf{f}^h , obtained by implementing trilinear interpolation in the hash grid, to model the context of anchor attributes. There is rich mutual information between anchor feature \mathbf{f}^a and hash feature \mathbf{f}^h , thus maximizing the conditional probability of both can reduce the entropy of the feature and bit consumption (Chen et al. 2025).

Anchor feature modeling. To facilitate entropy coding, the values of \mathcal{A} must be quantized into a finite set. HAC utilizes uniform distribution-based noise and rounding operations to implement quantization during the training and testing phases, respectively, which are not sufficiently dynamic and smooth. Therefore, we propose a dynamic quantization strategy. Specifically, for the i -th anchor \mathbf{x}_i^a , we denote \mathbf{f}_i as any of its \mathcal{A}_i 's components: $\mathbf{f}_i \in \{\mathbf{f}_i^a, \mathbf{l}_i, \mathbf{o}_i\} \in \mathbb{R}^D$, where $D \in \{D^a, 6, 3K\}$ is its respective dimension. In the training phase, we construct a Gaussian noise to update the features, denoted as:

$$\hat{\mathbf{f}}_i = \mathbf{f}_i + \mathcal{N}(0, \omega_i^2), \quad (12)$$

where $\omega_i = \eta_i (1 + \text{Tanh}(\mathcal{F}_q(\mathbf{f}_i^h)))$ with $\eta_i \in \{2.5e - 1, 2.5e - 4, 5e - 2\}$, and \mathcal{F}_q is an MLP for generating factors to dynamically optimize quantization. In the inference phase, we utilize a semi-soft rounding operation to make the quantized results closer to the true values, but still retain some discretization, expressed as:

$$\hat{\mathbf{f}}_i = k \cdot \omega_i + \text{Tanh}\left(\frac{\mathbf{f}_i - k \cdot \omega_i}{\tau}\right) \cdot \omega_i, \quad (13)$$

where the smoothing hyperparameter τ is 1. To measure and reduce the bit consumption of \mathbf{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 (Chen et al. 2025). 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 an MLP \mathcal{F}_g from \mathbf{f}^h . The probability of $\hat{\mathbf{f}}_i$ is computed as:

$$p(\hat{\mathbf{f}}_i) = \int_{\hat{\mathbf{f}}_i - \frac{1}{2}\omega_i}^{\hat{\mathbf{f}}_i + \frac{1}{2}\omega_i} \phi_{\mu_i, \sigma_i}(x) dx, \quad (14)$$

where ϕ represents the probability density function and $\mu_i, \sigma_i = \mathcal{F}_g(\mathbf{f}_i^h)$. Ultimately, we define the entropy loss as the sum of the bit consumption of all $\hat{\mathbf{f}}_i$:

$$\mathcal{L}_{entropy} = \beta \sum_{\mathbf{f} \in \{\mathbf{f}^a, \mathbf{l}, \mathbf{o}\}} \sum_{i=1}^N \sum_{j=1}^D \left(-\log_2 p(\hat{\mathbf{f}}_i^j)\right), \quad (15)$$

where $\beta = \frac{1}{N(D^a + 6 + 3K)}$, N is the number of anchors, $D^a = 50$ is the anchor feature dimension, $K = 10$ is the number of learnable offsets and $\hat{\mathbf{f}}_i^j$ means the j -th dimension value of $\hat{\mathbf{f}}_i$. Minimizing the entropy loss achieves a high probability estimation of $p(\hat{\mathbf{f}}_i)$ that guides the learning of the contextual model. The SCC loss is denoted as:

$$\mathcal{L}_{SCC} = \lambda_4 \mathcal{L}_{vol} + \lambda_5 \mathcal{L}_{entropy}, \quad (16)$$

where λ_4 and λ_5 are set to $1e - 2$ and $2e - 3$, \mathcal{L}_{vol} is the regularization term defined in (Lu et al. 2024).

Optimization Objectives

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

$$\mathcal{L} = \mathcal{L}_{RGB} + \mathcal{L}_{DPR} + \mathcal{L}_{SCC}, \quad (17)$$

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

Experiments

Text prompts and Evaluation Metrics

Text Prompts. To achieve a fair and comprehensive comparison, we select 9 text prompts describing the indoor, outdoor, and artistic style scenes: (1) A living room with a lit furnace, couch and cozy curtains, bright lamps that make the room look well-lit. (2) A cozy living room in Christmas. (3) A small cabin on top of a snowy mountain, Disney style. (4) A suburban street in North Carolina on a bright, sunny day. (5) Simple museum, pictures, paintings, artistic, best quality, dimly lit. (6) A children's room filled with toys and books. (7) A sunroom with floor-to-ceiling windows overlooking the garden, comfortable chairs, and a coffee table inside. (8) A sunny beach with fine sand and blue water, with a backdrop of blue sky and white clouds. (9) A winter snow scene with snow-covered trees and houses.

Evaluation Metric. Previous reference-based metrics (*e.g.*, PSNR and LPIPS (Zhang et al. 2018)) are not suitable for this generation task due to the lack of 3D scenes related to text prompts as reference. Therefore, six 2D metrics are used 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 the alignment between the rendered images and the input text prompts. In addition, the appearance and feel of the images are evaluated in a way that is more closely aligned with human perception through the Colorful, Quality, and Sharp metrics of CLIP-IQA (Wang, Chan, and Loy 2023). Moreover, we use the size of the model to measure the storage overhead.



Figure 2: Qualitative comparison results. The input text prompt is: “A cozy living room in Christmas”



Figure 3: Qualitative comparison results. The input text prompt is: “A small cabin on top of a snowy mountain, Disney style”

Implementation Details

To maximize the generalization ability of the proposed BloomScene, we use pre-trained models to build the entire architecture. Specifically, Stable Diffusion v1.5 (Rombach et al. 2022) is used to generate the initial image from the text prompt. If the input is an image without a corresponding text description, LLaVa (Contributors 2023) is used to generate the corresponding text prompt from the image, constituting an image-text pair. We use the Stable Diffusion v1.5 Inpainting model (Rombach et al. 2022) as the text-conditioned image inpainting model. We use ZoeDepth (Bhat et al. 2023) as the monocular depth estimator. To generate 3D scenes, we move the camera with a rotation of 0.63 radians. 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 (Höller et al. 2023), Invisible-stitch (Engstler et al. 2024) and LucidDreamer (Chung et al. 2023), 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 rendered RGB images of our method and baseline methods in the 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, leading to distorted and over-smoothing areas in the scene. (iii) LucidDreamer is currently the most visually effective progressive scene generation method 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 rendered results.

Quantitative Results. Table 1 shows the average quantitative results for multiple scenes. We can conclude the

Models	Size (MB) ↓	CLIP-Score ↑	CLIP-IQA ↑			BRISQUE ↓	NIQE ↓
			Quality	Colorful	Sharp		
Text2Room (Höllerlein 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)	—	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)	99.22	31.78	0.70	0.79	0.45	20.16	2.92

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

Models	Size (MB) ↓	CLIP-Score ↑	CLIP-IQA ↑			BRISQUE ↓	NIQE ↓
			Quality	Colorful	Sharp		
BloomScene (full)	99.22	31.78	0.70	0.79	0.45	20.16	2.92
w/o DPR	101.35	31.38	0.67	0.77	0.40	22.15	3.03
w/o SCC	569.33	31.60	0.68	0.78	0.44	22.46	2.96
w/o $\mathcal{L}_{DPR}^{smooth}$	101.11	31.68	0.67	0.78	0.42	20.37	2.93
w/o \mathcal{L}_{DPR}^{dist}	100.79	31.72	0.67	0.77	0.42	20.76	2.94
w/o $\mathcal{L}_{DPR}^{pixel}$	101.19	31.54	0.66	0.77	0.42	20.57	2.95

Table 2: Ablation results of different components.

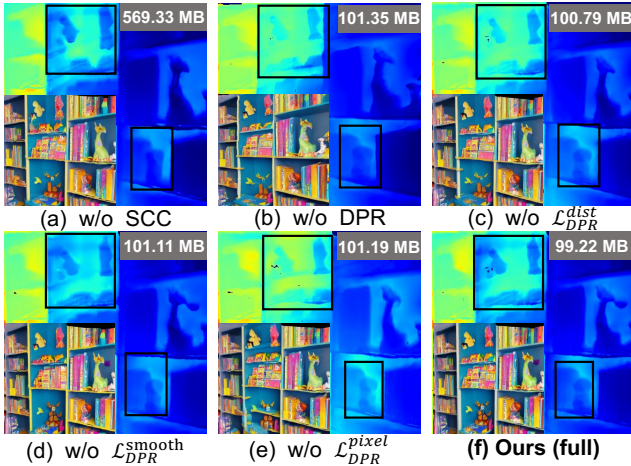


Figure 4: Visualization of ablation results.

following points: (i) Overall, our method generates much higher quality 3D scenes with significantly reduced storage overhead, significantly outperforming the baseline models. (ii) The storage overhead of our generated scenes is 4.3x and 5.8x lower than Invisible-stitch (Engstler et al. 2024) and LucidDreamer (Chung et al. 2023) using 3DGS. The storage overhead is also significantly reduced compared to Text2Room (Höllerlein et al. 2023) and SceneScape (Fridman et al. 2024) using mesh. (iii) We achieve the best performance on all the 2D metrics. In addition, the BRISQUE and NIQE scores are 20.16 and 2.92, respectively, which are reduced by 16.2% and 4.3% compared to the optimal scores. This indicates that our method fully utilizes the geometric information of the scene, effectively reduces distortions of the scene, and enhances the alignment of the scenes with the input text prompts.

Ablation Studies

To verify the necessity of the different components, we perform comprehensive ablation experiments using the same set of text prompts. Figure 4 shows the rendered results and Table 2 shows the average quantitative results for multiple scenes. (i) Firstly, DPR is removed from BloomScene. The decreased performance and the worse depth rendered results 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. 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.

Acknowledgements

This work was supported by National Key R&D Program of China 2021ZD0113502.

References

- 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*.
- Chen, Y.; Wu, Q.; Lin, W.; Harandi, M.; and Cai, J. 2025. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, 422–438. Springer.
- 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>.
- 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. 2023. 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.
- Hou, X.; Li, M.; Chen, J.; Yang, D.; Qian, Z.; and Zhang, L. 2024. SceneWeaver: Text-Driven Scene Generation with Geometry-aware Gaussian Splatting. In *The 16th Asian Conference on Machine Learning (Conference Track)*.
- 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. 2024. 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. Dream-Scape: 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.