# DIScene: Object Decoupling and Interaction Modeling for Complex Scene Generation

XIAO-LEI LI, BNRist, Department of Computer Science and Technology, Tsinghua University, China
HAODONG LI, The Hong Kong University of Science and Technology (Guangzhou), China
HAO-XIANG CHEN, BNRist, Department of Computer Science and Technology, Tsinghua University, China
TAI-JIANG MU*, BNRist, Department of Computer Science and Technology, Tsinghua University, China
SHI-MIN HU, BNRist, Department of Computer Science and Technology, Tsinghua University, China


Fig. 1. **DIScene** bridges complex 3D scene generation with the industrial production pipeline. DIScene is capable of generating complex 3D scene with decoupled objects (red text) and clear interactions (light blue text) from natural language description or reference image. The generated scene can be edited successively by changing interactive objects or their attributes (green text). The generated content could be integrated into industrial production pipeline for downstream applications, like film, games and animations. Here we show two generation and editing results, as well as the effect of integrating several generated examples into Maxon Cinema 4D.

*Corresponding author: Tai-Jiang Mu (taijiang@tsinghua.edu.cn).

Authors' Contact Information: Xiao-Lei Li, BNRist, Department of Computer Science and Technology, Tsinghua University, China, li-xl23@mails.tsinghua.edu.cn; Haodong Li, The Hong Kong University of Science and Technology (Guangzhou), China, haodongli@zju.edu.cn; Hao-Xiang Chen, BNRist, Department of Computer Science and Technology, Tsinghua University, China, chx20@mails.tsinghua.edu.cn; Tai-Jiang Mu, BNRist, Department of Computer Science and Technology, Tsinghua University, China, taijiang@tsinghua.edu.cn; Shi-Min Hu, BNRist, Department of Computer Science and Technology, Tsinghua University, China, shimin@tsinghua.edu.cn.



*SA Conference Papers '24, December 03–06, 2024, Tokyo, Japan*

This paper reconsiders how to distill knowledge from pretrained 2D diffusion models to guide 3D asset generation, in particular to generate complex 3D scenes: it should accept varied inputs, i.e., texts or images, to allow for flexible expression of requirement; objects in the scene should be style-consistent and *decoupled* with clearly modeled *interactions*, benefiting downstream tasks. We propose *DIScene*, a novel method for this task. It represents the entire 3D scene with a learnable structured scene graph: each node explicitly models an object with its appearance, textual description, transformation, geometry as a mesh attached with surface-aligned Gaussians; the graph's edges model object interactions. With this new representation, objects are optimized in the canonical space and interactions between objects are optimized by object-aware rendering to avoid wrong back-propagation. Extensive experiments


ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687589

demonstrate the significant utility and superiority of our approach and that DIScene can greatly facilitate 3D content creation tasks.

CCS Concepts: • **Computing methodologies** → **Shape modeling**; **Rendering**; **Neural networks**.

Additional Key Words and Phrases: instance-aware surface gaussian splatting, 3d generation, diffusion models, scene relation graph

**ACM Reference Format:**
Xiao-Lei Li, Haodong Li, Hao-Xiang Chen, Tai-Jiang Mu, and Shi-Min Hu. 2024. DIScene: Object Decoupling and Interaction Modeling for Complex Scene Generation. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 03–06, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3680528.3687589

## 1 Introduction

The production of complex, high-quality 3D scene is an essential component in industries such as film, gaming and animation. Yet, this production process presents a tremendous challenge even for domain experts, due to the significant workload involved, e.g., creating the basic models with modeling softwares like Maya or Blender, enriching/sculpturing the models' details with tools like ZBrush or Mudbox, and generating high-quality texture UV maps for the models with Substance Painter or Photoshop. Recent image-diffusion-based methods, which enables directly generating 3D content from texts and/or images, hold the potential of accelerating or even revolutionizing the pipeline, such as (i) distilling knowledge from pre-trained 2D diffusion models, with approaches like Score Distillation Sampling (SDS), to guide the optimization of 3D representations [Chung et al. 2023; Li et al. 2024; Lin et al. 2023b; Poole et al. 2023; Raj et al. 2023; Wang et al. 2023], and ii) the direct generation of 3D content through a feedforward network [Hong et al. 2024; Tang et al. 2024a; Xu et al. 2024] based on 3D consistent multi-view images, video diffusion results [Voleti et al. 2024], or tri-planes [Hong et al. 2024; Shi et al. 2024; Wang and Shi 2023].

To meet real requirements, downstream tasks impose an instance-aware understanding and control on the objects. This would be even greater challenging when the scene contains contact-like interactions. However, existing methods either adopt the implicit representation [Gao et al. 2024], making it hard to be edited or processed, incapable of modeling the geometry at the boundaries between objects; or optimize the entire scene with a single shared representation for different objects, causing the objects to be coupled with each other. Though the very recent work, i.e., GALA3D [Zhou et al. 2024], attempts to decouple the objects with non-shared explicit representations, it still suffers from surface penetration and fails to generate accurate and clear contact-like interactions between objects due to the unsuitable modeling of object relationships.

We thus reconsider how current complex 3D scene generation methods using 2D diffusion models can truly contribute to the industrial production pipeline. Firstly, it should accept varied inputs, i.e., texts or images, to allow for flexible expression of requirement. Secondly, objects in the scene should be *style-consistent* and *decoupled* with clearly modeled *interactions*, benefiting downstream tasks such as editing, animation, rendering, etc.

To achieve the above goals, we propose *DIScene*, a novel method for complex 3D scene generation, which represents the entire 3D scene with a *learnable structured scene graph*: each node explicitly models an object with its appearance, textual description, transformation, and geometry as an individual mesh attached with surface-aligned Gaussians in the canonical space; the graph's edges model object interactions. The former facilitates the natural decoupling and independent deformation of the object, allowing for penetration avoidance and significant scale differences in generated objects. The latter enhances the modeling of overall stylistic consistency and spatial coherence with the interaction. Besides, decoupling the inputs into individual objects and their relationships enables an easier and more accurate understanding of complex scenes. To effectively optimize our scene graph, we further propose an object-aware rendering which composes the final pixel according to the depth of objects, avoiding usually wrong gradient back-propagation in contact-like interaction.

As a result, *DIScene* supports for custom, complex, and flexible editing capabilities. For instance, transitioning from "holding a flower in the right hand" to "holding it in the left hand", or from "wearing a motorcycle helmet" to "wearing a top hat". Moreover, the generated scenes in mesh can be seamlessly integrated into industrial production pipelines, enabling individual animation and deformation, which greatly simplifies and accelerates existing 3D asset creation workflow.

In summary, our main contributions are as follows:

- A clear definition on how complex 3D scene generation with distilling knowledge from 2D diffusion models should function in real-world applications, addressing challenges in the industrial 3D content production workflow.
- A novel method, *DIScene*, utilizing a learnable structured scene graph to model the entire scene with complex inputs decomposition, object decoupling and interaction guidance.
- A novel object-aware rendering for optimization, avoiding wrong gradient back-propagation in contact-like interaction.

## 2 Related Work

### 2.1 Differentiable 3D Representations

Given a 3D representation, with trainable parameter $\theta$, one can optimize it to fit the condition (such as different kinds of images or texts) using a differentiable rendering function $g(\theta, c)$ to get an image in camera pose $c$ of that 3D representation. Previously, various differentiable 3D representations have been studied, such as volumetric representations [Brock et al. 2016; Gadelha et al. 2017; Li et al. 2019; Mu et al. 2023; Wu et al. 2016] and point clouds [Achlioptas et al. 2018; Pumarola et al. 2020]. Recently, NeRFs [Mildenhall et al. 2022] grows to be the most common representation in 3D generation tasks [Metzer et al. 2023; Poole et al. 2023; Wang et al. 2023]; but they struggle to generate high-resolution content due to the heavy volume rendering process needed by this implicit representation comparing with explicit ones. Textured meshes [Shen et al. 2021] offer efficient explicit rendering and high-quality surface geometry, but are hard to obtained from scratch. Thus, a coarse mesh initialization are commonly used for detailed 3D generation [Sun et al. 2024]. Recently, 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] gains increasing attentions due to its advantage in rendering efficiency. Also, recent works start to focus on improving the geometry of

3DGS while maintaining rendering quality [Guédon and Lepetit 2024; Huang et al. 2024; Yu et al. 2024].

### 2.2 3D Object Generation

With advances in diffusion models: [Chan et al. 2023; Nichol and Dhariwal 2021; Ramesh et al. 2021; Rombach et al. 2022a; Shi et al. 2024; Tang et al. 2023; Wang and Shi 2023; Xu et al. 2023], DreamFusion [Poole et al. 2023] first proposes to distil the knowledge from 2D diffusion models to guide 3D generation. This method does not require extensive 3D data to optimize the 3D representation, and thus has been widely explored in subsequent research. Specifically, some works proposes more sophisticated score distillation loss functions [Qian et al. 2024; Sun et al. 2024; Wang et al. 2023; Zhang et al. 2023], optimization strategies [Chen et al. 2023; Lin et al. 2023b; Sun et al. 2024; Tang et al. 2024b; Zhu et al. 2024], and better 3D representations [Chen et al. 2023, 2024a; Li et al. 2024; Sun et al. 2024; Tang et al. 2024b; Wang et al. 2023; Yi et al. 2024] to further improve the quality of the generated results. Although these methods can generate high-fidelity 3D assets, they are very time-consuming. In contrast, feed-forward 3D native methods [Hong et al. 2024; Tang et al. 2024a; Xu et al. 2024], after being trained on extensive 3D datasets [Deitke et al. 2023; Sun et al. 2023], can generate 3D assets within seconds. However, these forward methods struggle to decouple objects and precisely understand the complex input.

### 2.3 Compositional Scene Generation

Earlier research [Niemeyer and Geiger 2021] has considered utilization of compositional neural radiance fields within an adversarial learning context to facilitate the generation of images, with 3D awareness. Recent methods [Cohen-Bar et al. 2023; Lin et al. 2023a; Po and Wetzstein 2024; Tertikas et al. 2023; Willis et al. 2022] incorporate additional contextual information, such as 3D layout data, representing a multi-object 3D scene in a compositional manner. However, creating the layout typically demands manual effort, which can be time-consuming and difficult for non-expert users. Gao et al. [2024] utilize a graph-based framework to only perform decoupling and structuring for semantics and leverage NeuS [Wang et al. 2021], an implicit field, as the object representation which is still coupled within the overall scene. Epstein et al. [2024] integrate layout learning within the optimization process, only attempt to decouple(not fully) the object representation but not decompose the semantics, nor do they model interactions. Concurrently, [Zhou et al. 2024] employ layout learning and explicit representation for decoupling but it's difficult to model complex interactions. Similarly, Chen et al. [2024b] learn a spatially-aware diffusion model to refine the positioning and interaction of objects but generate low-quality outputs.

DIScene has a clear definition of fully decoupling on both semantics and representations and precisely modeling the relations between objects.

## 3 Preliminaries

### 3.1 3D Gaussian Splatting and SuGaR

3D Gaussian splatting (3DGS)[Kerbl et al. 2023; Wu et al. 2024b] models a scene using a set of 3D Gaussians $\mathcal{H}$, where each Gaussian $h \in \mathcal{H}$ is described by its centroid $\mu_h \in \mathbb{R}^3$ and covariance matrix $\Sigma_h \in \mathbb{R}^{3\times3}$. The covariance $\Sigma_h$ is determined by a scaling vector $s_h \in \mathbb{R}^3$ and a rotation quaternion $q_h \in \mathbb{R}^4$. In addition, each Gaussian possesses an opacity $\alpha_h \in \mathbb{R}$ and color attributes $c_h$ which are used for rendering via the splatting technique. Although standard Gaussian splatting may struggle with accurate geometry representation, the SuGaR [Guédon and Lepetit 2024] method enhances this by integrating several regularization terms that promote flatness and proper alignment of 3D Gaussians with the object's surface. This enhancement aids in generating a mesh from the Gaussians via Poisson reconstruction [Kazhdan et al. 2006]. Additionally, SuGaR provides a mesh-Gaussian hybrid representation that ties Gaussians to mesh surfaces, enabling the concurrent optimization of texture and geometry via back-propagation.

During the rendering process, these 3D Gaussians are transformed into 2D Gaussians on the image plane, and their colors are blended using alpha compositing in a depth-sorted, front-to-back manner. We use a modified version proposed by LGM [Tang et al. 2024a] of the original 3DGS renderer, to support depth and alpha rendering, which is denoted as $x = g(\mathcal{H}, \mathbf{v})$, where $v$ is the camera matrix and $x = \{C, D, A\}$ for color map, depth map and alpha map, respectively. For vanilla Gaussians, we estimate the surface normal $N$ by taking the derivative of the depth. However, in contrast, we directly render the normal map with $g$ using mesh face normals as colors for mesh-Gaussian hybrid representation.

### 3.2 Score Distillation Sampling

Score Distillation Sampling (SDS) [Poole et al. 2023] leverages a pre-trained diffusion model as a prior to facilitate the creation of text-conditioned 3D assets. More precisely, SDS involves optimizing the parameters of a differentiable 3D representation (Gaussians in this paper) using the gradients of the loss function $\mathcal{L}_{\text{SDS}}$ with respect to $\theta$, guided by the pre-trained diffusion model $\epsilon_\phi$:

$$\nabla_{\mathcal{H}}\mathcal{L}_{\text{SDS}}(\phi, x) = \mathbb{E}_{t,\epsilon}\left[w(t)(\hat{\epsilon}_\phi - \epsilon)\frac{\partial g(\mathcal{H}, \mathbf{v})}{\partial \mathcal{H}}\right], \tag{1}$$

where $w(t)$ is a weighting function that varies with the timestep $t$. The core approach promotes adherence of the rendered 3D representation to the distribution learned by the diffusion model. Typically, the timestep $t$ and Gaussian noise $\epsilon$ are sampled randomly during each optimization step.

## 4 Complex 3D Scene Generation with Decoupled Objects and Interactions

Given a text $y$ or image $I$ containing multiple subjects and relationships, our focus is to generate a complex 3D scene of $M$ decoupled objects $O^c = \{o_i^c\}_{i=1}^M$ in canonical space with transformations $\mathcal{T} = \{T_i\}_{i=1}^M$. Firstly, we decouple the complex semantic information in the input through a learnable structured scene graph to provide better guidance for the optimization of individual objects and multi-object interactions (see Sec. 4.1). Furthermore, each individual object is defined and optimized in canonical space, decoupled using explicitly separable surface aligned gaussians and mesh-gaussian hybrid representation (see Sec. 4.2). More importantly, to better model their interactions, we utilize learnable transformation matrices within

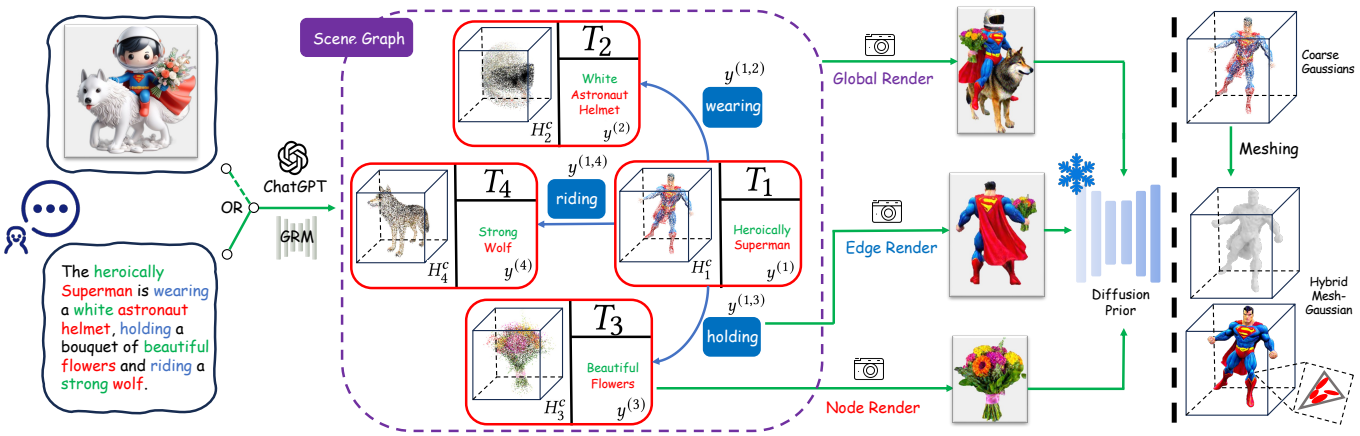



Fig. 2. **Overview of DIScene.** Given flexible input (image or text), our method initializes the scene graph using Large Language Models (LLMs) and Gaussain Reconstruction Model (GRM) [Xu et al. 2024]. The nodes and edges of the scene graph are rendered using different strategies and optimized through score distillation sampling guidance. The training pipeline comprises two stages with distinct object representations. In the first stage, coarse 3D Gaussians are used. In the second stage, a Hybrid Mesh-Gaussian is employed, which is extracted from the coarse Gaussians of the first stage.

the scene graph to transform corresponding objects from canonical space to global space and directly supervise their interactions (see Sec. 4.3).

Based on this concept, we present DIScene in Fig. 2 with an example of a four-object scene to aid understanding.

## 4.1 Structured Scene Graph

*Scene Graph Initialization.* Upon receipt of user text input $y$, we initialize the scene graph $\mathcal{G}(\mathcal{O}, \mathcal{E})$ with the aid of a Large Language Models (LLM), such as GPT-4(V) [OpenAI 2023], in which the nodes are $\mathcal{O} = \{H_i^c, T_i, y^{(i)}\}_{i=1}^{M}$ where M is the number of objects, the edges are $\mathcal{E} = \{e_{i,j}, y^{(i,j)} | i, j \in [1, M]\}$. Specifically, $H_i^c \in \mathcal{H}^c$ represents the independent object representation of node $o_i$ in canonical space, $T_i \in \mathcal{T}$ is the transformation matrix that transforms $H_i^c$ from the canonical space to the global space, and $y^{(i)}$ is the prompt corresponding to node $o_i$. The relationship, such as riding and holding, between two interacting nodes $o_i$ and $o_j$ is represented as $e_{i,j}$, and $y^{(i,j)}$ is the prompt corresponding to this interaction. The detailed explanation of the scene graph is as follows.

*Node Attributes.* Each node $o_i \in \mathcal{O}$ contains the independent Gaussian representation $H_i^c$ of the corresponding object in canonical space, a description $y^{(i)}$, and a transformation matrix $T_i \in \mathcal{T}$. The description $y^{(i)}$ encapsulates the intrinsic attributes of the object, such as color and emotional state (e.g., "cheerful green dinosaur", "a Lamborghini race car painted a vibrant shade of red"). For composing objects to model the interactions, the object $o_i^c$ are transformed to global space with a transformation matrix:

$$T_i = \begin{pmatrix} R_i & t_i \\ \mathbf{0} & s_i \end{pmatrix}, R_i \in SO(3), t_i \in \mathbb{R}^3, s_i \in \mathbb{R}, \tag{2}$$

where $R_i$ is the rotation matrix, $t_i$ is the translation vector, and $s_i$ is the scaling factor. And the attributes of each transformed Gaussians are computed by:

$$\mu_h' = s_i R \cdot \mu_h + t_i; \quad q_h' = q_h \otimes \text{quat}(R_i); \quad s_h' = s_h \cdot s_i, \tag{3}$$

where $\text{quat}(\cdot)$ represents rotation matrix to unit quaternion conversion and $\otimes$ denotes quaternion product.

*Edge Formulation.* In addition to completely decoupling the objects, our goal also includes modeling the interactions between them. Therefore, we store the relation prompt $y^{(i,j)}$ in the edge $e_{i,j}$, which provides guidance for interaction modeling after pair-wise object transformation and composition. It is important to note that not every pair of objects is necessarily connected by an edge (e.g. an astronaut helmet and a flower). We denote the number of edges in $\mathcal{E}$ as $K$, where $K \leq C_2^M$ (the maximum possible number of edges).

One can flexibly modify $y^{(i)}$ and $y^{(i,j)}$ as needed and we also have a global prompt $y$. With all of these, we could obtain a set of $(1+M+K)$ prompts, $M$ individual representations $H^c$ and $M$ transformations $\mathcal{T}$, which are used to guide scene generation and editing from the perspective of both decoupled individual objects in canonical space and pairwise or global relationships and interactions.

## 4.2 Subject Decoupling

*Canonical Objects Initialization.* For a complex 3D scene of $M$ objects, to decouple them and provide better, fairer supervision for objects with significant scale differences in the scene, we model them with a compositional instance-aware surface-aligned Gaussian representation. Specifically, for each node $o_i \in \mathcal{O}$, we first feed $y^{(i)}$ to a pretrained Gaussian generation model [Xu et al. 2024] to obtain $M$ initial vanilla 3DGS representations [Kerbl et al. 2023] $\mathcal{H}^c$ in canonical space, which are totally decoupled from each other.

*Surface-aligned Gaussians.* We assign an object label $i$ to the Gaussians in $\mathcal{H}_i^c$. Additionally, we develop an algorithm for updating Gaussian attributes that is attuned to this instance-aware representation. To further ensure that Gaussians at the interfaces between objects do not contribute to both objects simultaneously, which could have a detrimental effect on the geometry and texture optimization of interactions of decoupled objects (e.g. the hand of Superman in "Superman is holding a bunch of tulip flowers"), we

apply a surface regulation term denoted $\mathcal{L}_{\text{reg}}$ from SuGaR [Guédon and Lepetit 2024] for constraining $\mathcal{H}_i^c$ to the surface of the corresponding object $o_i^c$, including opacity binary entropy loss, density and normal regulation. For detailed descriptions, please refer to the supplementary materials. Building on this foundation, for the object $o_i^c \in O^c$, we query its description $y^{(i)}$ from the structured scene graph and we simultaneously distill diffusion prior from MVDream [Shi et al. 2024] and DeepFloyd-IF [Alex et al. 2023] for better coarse geometry and to avoid the Janus (multi-head) problem. Note that for MVDream we feed only an RGB image while for DeepFloyd-IF, we feed both an RGB image and a normal image to get better geometry, following [Chen et al. 2023]. Thus we can define the gradient for $\mathcal{H}_i^c$ with SDS following Eq. 1 as:

$$\begin{aligned}\nabla_{\mathcal{H}_i^c}\mathcal{L}_{\text{SDS}}^{(i)} &= \nabla\mathcal{L}(\mathcal{H}_i^c, g, \mathbf{v}_*, \eta, \mathbf{c}, y^{(i)}) \\ &= \lambda_{IF}\nabla_{\mathcal{H}_i^c}\mathcal{L}_{\text{SDS}}^{IF}\{x_*^{(i)} = g(\mathcal{H}_i^c, \mathbf{v}_*); \eta, y^{(i)}\} \\ &\quad + \lambda_{MV}\nabla_{\mathcal{H}_i^c}\mathcal{L}_{\text{SDS}}^{MV}\{x_*^{(i)} = g(\mathcal{H}_i^c, \mathbf{v}_*); \eta, \mathbf{c}, y^{(i)}\},\end{aligned} \tag{4}$$

where $\mathbf{v}_* \in \mathbb{R}^{4\times4\times4}$ corresponds to four orthogonal camera matrices; $\eta$ is the time step for optimization; $\mathbf{c}$ is the camera information required in MVDream; $\lambda_{IF}$ and $\lambda_{MV}$ are the strengths of DeepFloyd-IF and MVDream priors respectively.

*Mesh-Gaussian Hybrid Representation.* For better optimization of geometry and texture, after getting coarse surface-aligned Gaussians as above, we extract a mesh and bind a set of new flat Gaussians for each triangle face [Guédon and Lepetit 2024]. In the initial setup, the Gaussians are assigned colors based on the vertex colors of the triangles. Their positions are determined using predefined barycentric coordinates, and their rotations are controlled using complex numbers in two dimensions, ensuring they align properly within their respective triangles. We leverage Stable Diffusion [Rombach et al. 2022b] as guidance to achieve optimization at high resolutions for this hybrid representation with SDS, following Eq. 1:

$$\begin{aligned}\nabla_{\mathcal{H}_i^c}\mathcal{L}_{\text{SDS}}^{(i)} &= \nabla\mathcal{L}'(\mathcal{H}_i^c, g, \mathbf{v}_r, \eta, y^{(i)}) \\ &= \nabla_{\mathcal{H}_i^c}\mathcal{L}_{\text{SDS}}^{SD}\{x_r^{(i)} = g(\mathcal{H}_i^c, \mathbf{v}_r); \eta, y^{(i)}\},\end{aligned} \tag{5}$$

where $\mathbf{v}_r$ corresponds to $B$ randomly sampled views and $\mathcal{H}_i^c, \eta, y^{(i)}$ remains the meaning of Eq. 4. Additionally, a regularization term $\mathcal{L}'_{\text{reg}}$ is applied to the mesh, which includes normal consistency and Laplacian smoothing.

Notably, since $o_i^c$ is in the canonical space, $x_*^{(i)}$ in Eq. 4 or $x_r^{(i)}$ in Eq. 5 centers around the object and features rich geometric and textural details, which will enhance the optimization for small objects in scenes with significant scale differences. Additionally, through this representation, the Gaussians $\mathcal{H}$ are constrained to the surfaces of $o_i$, ensuring that Gaussians from different objects do not interfere with each other and allowing for the extraction of textured meshes to be further processed.

## 4.3 Interaction Modeling

Building on the structured scene graph discussed in Sec 4.1 and decoupled object representation in Sec. 4.2, we propose a novel compositional optimization to model the complex relations and interactions of pairwise objects and global scene. It contains following three key parts:

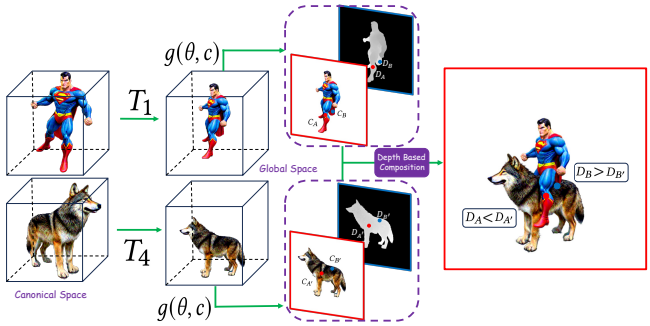



Fig. 3. **Object-aware rendering.** The objects are transformed into global space and rendered separately. Then the final render result is the composition of each object based on their depth orders.

*Object-aware Rendering.* For better modeling of interfaces and interactions between objects (connected by graph edges), we propose a novel multi-object rendering method called *Object-aware Rendering*. Even though the Gaussians of different objects are separated in global space through the above representation, the challenge of using traditional Gaussian rendering, when multiple interacting or occluding objects are rendered into image space, lies in that they still influence each other's gradient propagation. This influence arises from alpha blending at the Gaussian level, especially when Gaussians from different objects mix with each other. Therefore, we avoid alpha blending at the Gaussian level and instead perform it in pixel space as shown in Fig. 3. For example, to optimize the interaction between objects $o_i^c$ and $o_j^c$ connected by $e_{i,j}$, we first transform $o_i^c$ and $o_j^c$ from the canonical space to global space using transformations $T_i$ and $T_j$ respectively:

$$H_i = T_i(H_i^c); H_j = T_j(H_j^c). \tag{6}$$

Our object-aware renderer $g_{\mathcal{P}}$ renders each individual object in global space and composites the color images according to their depth. In detail, each object $o_i$ is rendered by $g(H_i, c)$, resulting in color image $C^i$, and depth image $D^i$. Then the color images are composited according to depth images:

$$\begin{aligned}U_{i,j} &= \arg\min_k D_{i,j}^k, k \in [1, \ldots, N], \\ C_{i,j} &= C_{i,j}^k, \text{where } k = U_{i,j},\end{aligned} \tag{7}$$

where $U$ is an index map which stores the image index of minimum depth value at each pixel $(i, j)$.

*Pairwise interaction modeling.* We design an optimization strategy that focuses on the details of pairwise objects interactions based on object-aware rendering $g_{\mathcal{P}}$. To be specific, for object $o_i^c$ and $o_j^c$ connected by $e_{i,j}$, we query relationship description $y^{(i,j)}$ in the structured scene graph. Then use the transformation matrices $T_i, T_j$ to transform $o_i^c$ and $o_j^c$ from canonical space to global space and do object-aware rendering on them. For surface-aligned Gaussians:

$$\nabla_{\mathcal{H}_{i,j}^c}\mathcal{L}_{\text{SDS}}^{(i,j)} = \nabla\mathcal{L}(\mathcal{H}_i, \mathcal{H}_j, g_{\mathcal{P}}, \mathbf{v}_*, \eta, \mathbf{c}, y^{(i,j)}), \tag{8}$$


Fig. 4. **Qualitative comparison with baseline approaches.** DIScene generates scenes with all composing objects being separable.

and for mesh-Gaussian hybrid representation:

$$\nabla_{\mathcal{H}^c_{i,j}} \mathcal{L}^{(i,j)}_{\text{SDS}} = \nabla \mathcal{L}'(\mathcal{H}_i, \mathcal{H}_j, g_{\mathcal{P}}, \mathbf{v}_r, \eta, y^{(i,j)}). \tag{9}$$

In addition, the intersection area can be easily computed for mesh-Gaussian hybrid representation, so we incorporate a penetration loss to refine the transformation matrices $\mathcal{T}$ and local geometry to better model the intersecting boundaries. Specifically, in each step, for the currently selected object $o_i$ and any object $o_j$ that might intersect $o_i$, we aim to minimize the distance between intersecting $o_i$ points and the surface of $o_j$. Thus, we define the penetration loss as:

$$\mathcal{L}_{\text{penetr}} = \sum_{i=1}^{n} \left( \left\| \mu'_j - v_i \right\|_2 \max\{\text{sgn}((\mu'_j - v_i) \cdot n'_j), 0\} \right), \tag{10}$$

where $v_i \in \mathcal{V}_i$ is a vertex on the mesh of object $o_i$, $\mu'_j$ is the centroid of Gaussian $h'_j$ on object $o_j$ which is closest to $v_i$, $\text{sgn}(\cdot)$ is the sign function which outputs $\{1, -1\}$, and $n'_j$ is the normal of $h'_j$.

*Global style optimization.* Further, to ensure that the entire scene maintains rational spatial relationships and a consistent style, we need to optimize the global scene. Here, we directly utilize an extension of pairwise interaction modeling, transitioning from focusing on pairwise objects to encompassing all objects $O^c$ in the scene and using the global prompt $y_g$, For surface-aligned Gaussians:

$$\nabla_{\mathcal{H}^c} \mathcal{L}^{\mathcal{G}}_{\text{SDS}} = \nabla \mathcal{L}(\mathcal{H}, g_{\mathcal{P}}, \mathbf{v}_*, \eta, \mathbf{c}, y), \tag{11}$$

and for mesh-Gaussian hybrid representation:

$$\nabla_{\mathcal{H}^c} \mathcal{L}^{\mathcal{G}}_{\text{SDS}} = \nabla \mathcal{L}'(\mathcal{H}, g_{\mathcal{P}}, \mathbf{v}_r, \eta, y). \tag{12}$$

*Training Objectives.* Instead of rendering all objects and edges in the scene-graph in every step, which is quite memory-intensive, we randomly choose one object $o^c_i$ and an edge $e_{i,j}$ linked to it and optimize them per step. The global optimization is performed every $M$ steps. Thus, the total SDS loss is:

$$\nabla_{\mathcal{H}^c} \mathcal{L}_{\text{SDS}} = \begin{cases} \lambda_O \nabla_{\mathcal{H}^c_i} \mathcal{L}^{(i)}_{\text{SDS}} + \lambda_{\mathcal{E}} \nabla_{\mathcal{H}^c_{i,j}} \mathcal{L}^{(i,j)}_{\text{SDS}}, & \text{if } s\%(M+1) = 0 \\ \lambda_{\mathcal{G}} \nabla_{\mathcal{H}^c} \mathcal{L}^{\mathcal{G}}_{\text{SDS}}, & \text{otherwise} \end{cases} \tag{13}$$

where $s$ indexes the current training step, $j$ refers to any other node which has an edge connected to $o^c_i$, and $\lambda_O$, $\lambda_{\mathcal{E}}$, $\lambda_{\mathcal{G}}$ are wights for canonical, interaction and global style optimization respectively.

To ensure the smoothness of generated geometry, we incorporate total variation (TV) regularization terms [Rudin and Osher 1994] on depth and normal map, denoted as $\mathcal{L}^d_{\text{TV}}$ and $\mathcal{L}^n_{\text{TV}}$.

Our final loss used for training DIScene is as follows, for stage 1 optimization on surface-aligned Gaussians:

$$\mathcal{L}^{(1)}_{\mathcal{H}^c} = \mathcal{L}_{\text{SDS}} + \beta^{(1)}_1 \mathcal{L}^d_{\text{TV}} + \beta^{(1)}_2 \mathcal{L}^n_{\text{TV}} + \beta^{(1)}_3 \mathcal{L}_{\text{reg}}, \tag{14}$$

and for stage 2 optimization on the hybrid mesh-Gaussian representation:

$$\mathcal{L}^{(2)}_{\mathcal{H}_c} = \mathcal{L}_{\text{SDS}} + \beta^{(2)}_1 \mathcal{L}^d_{\text{TV}} + \beta^{(2)}_2 \mathcal{L}^n_{\text{TV}} + \beta^{(2)}_3 \mathcal{L}'_{\text{reg}} + \beta^{(2)}_4 \mathcal{L}_{penetr}, \tag{15}$$

where $\beta^{(1)}_{k=1,2,3}$ and $\beta^{(2)}_{k=1,2,3,4}$ are weights for stage 1 and 2 respectively.

# 5 Experiments Results

## 5.1 Settings of Experiments

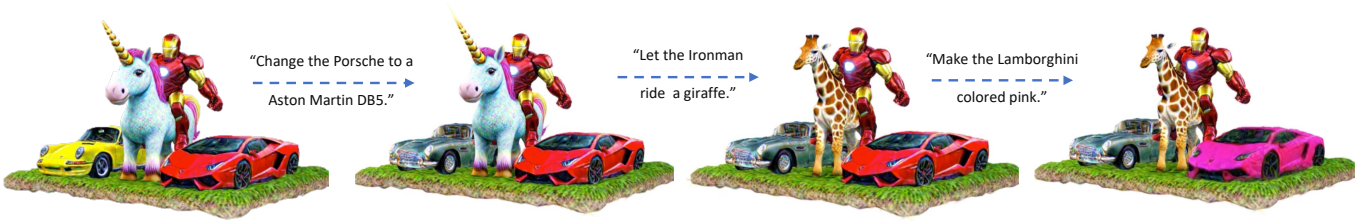
*Implementation details.* We employed GPT-4V [OpenAI 2023] and the official code and weights of GRM [Xu et al. 2024] for initialization. The entire optimization process followed a coarse-to-fine strategy with $\lambda_O = 0.01$, $\lambda_{\mathcal{E}} = 0.02$, $\lambda_{\mathcal{G}} = 0.01$, respectively. The coarse optimization stage comprises a total of 2000 steps. In the initial 1000 steps, we utilized the original Gaussian optimization strategy without densification and pruning. In next 1k steps, we enabled them and introduced SuGaR regularization terms $\mathcal{L}_{\text{reg}}$ to constrain the Gaussians to the surface. We use $\lambda_{IF} = 0.1$ and $\lambda_{MV} = 0.01$ to balance DeepFloyd [Alex et al. 2023] and MVDream [Shi et al. 2024] diffusion guidance with SDS. The $\beta^{(1)}_{k=1,2,3}$ and $\beta^{(2)}_{k=1,2,3}$ are all equal to 1.0, special for $\beta^{(2)}_4 = 0.1$. In fine stage of 20000 steps, hybrid Mesh-Gaussian representation are used and we use Screened Poisson Surface Reconstruction algorithm for extracting surface from Gaussians and bind 6 Gaussians on each face. For different diffusion guidance, we render images with different resolution: 256× 256 for MVDream, $64 \times 64$ for DeepFloyd and $512 \times 512$ for Stable Diffusion [Rombach et al. 2022b]. The entire optimization process was conducted on a single NVIDIA L20 GPU.

*Baseline approaches.* We compared ours against following state-of-the-art (SOTA) approaches: (i)*GraphDreamer* [Gao et al. 2024] and *Set-the-Scene* [Cohen-Bar et al. 2023] are two representative compositional 3D generation methods similar to ours; (ii)high-quality generation methods without compositional manner: MVDream [Shi et al. 2024], *LucidDreamer* [Liang et al. 2024] and *DreamCraft3D* [Sun et al. 2024]. We followed the official implement for training all baselines. Specifically, we use bounding boxes of initialized objects in global space as the shape proxies for Set-the-Scene; when the input is text, we use Stable Diffusion [Rombach et al. 2022b] to generate the corresponding images as input for DreamCraft3D.

*Evaluation metrics.* We report the CLIP Score (text-image alignment) [Radford et al. 2021] and GPTEval3D ELO Scores [Wu et al. 2024a] in quantitative comparison with baseline models. CLIP Score measures the similarity between a text prompt $y$ and a rendered image $C$: $\text{CLIP}(C, y) = \cos\langle E_C(C), E_Y(y)\rangle$, with $E_C(C)$ the visual embedding and $E_Y(y)$ the textual embedding. GPTEval3D employes GPT-4(V) [OpenAI 2023] to compare two 3D assets according to user-defined criteria. Then, we can use these pairwise comparison results to assign these models ELO ratings. We use OpenCLIP ViT-B/32 for visual and textual encoders, and the `GPT-4-Vision-Preview` API for computing the ELO Scores from 600 pairwise comparisons. The criteria are: 1) Text-Asset Alignment (TA-A), 2) 3D Plausibility (3D-P), 3) Text-Geometry Alignment (TG-A), 4) Texture Details (T-D), and 5) Geometry Details (G-D). A total of 10 complex scenes are generated, each containing an average of 4.25 objects, and a total of 16 images are rendered from each 3D scene for evaluation.

## 5.2 Comparisons

We report the comparison of DIScene with baselines, including the qualitative comparison in Fig. 4 and the quantitative comparison in Table 1 and Table 2. GraphDreamer [Gao et al. 2024], cannot fully




"The Iron Man rides A cute Unicorn on the grass with a red Lamborghini on his left and a yellow Porsche on his right."


"Three beautiful angels, are gathering around and looking at a cute cat with wings that is flying above a exquisite table together."


The Superman is wearing a white astronaut helmet, holding a bouquet of flowers and riding a strong wolf.


a Wizard standing before a Wooden Desk, gazing into a Crystal Ball perched atop the Wooden Desk, with a Stack of Ancient Spell Books perched atop the Wooden Desk."


A medieval soldier has a sword at his waist, holding A bottle of Coca-Cola and riding a wolf.


An astronaut is sitting on a dark green leather sofa watching TV that is placed on a cabinet. There is a square wooden coffee table between him and the TV, on which is a colorful vase.

Fig. 5. **Interactive editing examples.** The objects in the scene are fully decoupled, allowing for precise and controllable editing.

Table 1. Quantitative results of CLIP Score. The mean and standard deviation (std.) values are reported.

| CLIP Score | DIScene | DreamCraft3D | LucidDreamer | MVDream | SetTheScene | GraphDreamer |
|---|---|---|---|---|---|---|
| Mean ↑ | **0.3688** | 0.3399 | 0.3237 | 0.3099 | 0.2996 | 0.2784 |
| Std. ↓ | **0.0334** | 0.0402 | 0.0419 | 0.0423 | 0.0453 | 0.0678 |

Table 2. Quantitative results of GPTEval3D ELO Scores.

| ELO Scores | DIScene | MVDream | LucidDreamer | DreamCraft3D | GraphDreamer | SetTheScene |
|---|---|---|---|---|---|---|
| TA-A $\times 10^3$ ↑ | **1.571** | 0.5652 | 0.8435 | 0.7236 | 0.4017 | 0.5557 |
| 3D-P $\times 10^3$ ↑ | **1.433** | 1.133 | 1.136 | 0.7925 | 1.061 | 0.5949 |
| TG-A $\times 10^3$ ↑ | **1.754** | 1.440 | 1.038 | 0.9103 | 0.3224 | 0.9447 |
| T-D $\times 10^3$ ↑ | **1.659** | 1.139 | 1.013 | 0.7472 | 0.4240 | 0.5470 |
| G-D $\times 10^3$ ↑ | **1.685** | 1.079 | 1.078 | 1.053 | 0.3483 | 0.3497 |

decouple objects nor accurately model object interaction surfaces, resulting in mixed and entangled generation outcomes, partial loss of some objects and poor geometry. Set-the-Scene [Cohen-Bar et al. 2023], which requires 3D layout as additional input, often generates narrow-view and low-quality 3D content, especially when the 3D layout is not very accurate. MVDream [Shi et al. 2024] fails to decouple objects and understand the semantic relationships in complex text descriptions, resulting in loss of intent and confusion, leading a complete misalignment between generated results and input. LucidDreamer [Liang et al. 2024], suffering from the same defect as MVDream, cannot decouple objects in the scene and understand complex inputs, leading to mixed objects, misalignment with the input, and poor 3D consistency. Without decomposing the complex semantic information of the input, the diffusion model providing 3D prior used in DreamCraft3D [Sun et al. 2024] fails to understand the scene geometry, leading to a severe Janus problem. DIScene models the entire scene as a learnable scene graph, decomposing the object semantics and their relationships in complex inputs. It provides accurate optimization guidance for fully decoupled objects and clearly models the spatial and interaction relationships between multiple objects. Both qualitative and quantitative comparisons demonstrate the significant superiority of DIScene over existing methods.

## 5.3 Interactive Instruction Editing

The objects in the complex scenes generated by DIScene are fully decoupled, allowing users to interact with the scene. The user's input editing instructions are integrated into the current learnable scene graph through LLMs and GRM [Xu et al. 2024]. Further refinement is performed based on the edited Scene Graph with fewer iterations. By focusing optimization on the edited nodes and limiting the learning rate for other objects in the scene, we can achieve highly controllable and personalized scene editing while maintaining the stability of other objects. Notably, in addition to simple additions, deletions, spatial transformations, and style transfers, we can also edit complex interactions within the scene as illustrated in Figs. 1 and 5. For example, changing "Superman holding a flower in his right hand" to "holding a flower in his left hand," changing "riding a unicorn" to "riding a giraffe," or changing "wearing an astronaut helmet" to "wearing a baseball cap",

Table 3. User study results.

| | DIScene | DreamCraft3D | MVDream | LucidDreamer | GraphDreamer | SetTheScene |
|---|---|---|---|---|---|---|
| Scene Quality ↑ | **9.3** | 7.4 | 5.9 | 6.1 | 3.9 | 4.4 |
| Input alignment ↑ | **9.1** | 5.3 | 4.3 | 5.1 | 4.7 | 4.2 |
| Geometric fidelity↑ | **8.9** | 6.9 | 5.5 | 6.6 | 3.4 | 3.5 |
| Scene consistency ↑ | **9.7** | 5.3 | 7.9 | 6.1 | 4.6 | 4.8 |

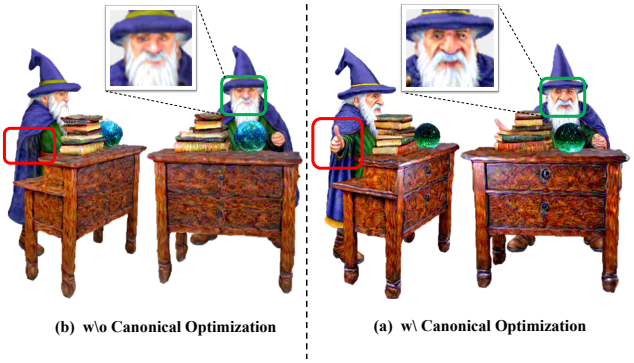



Fig. 6. **Ablation study on Canonical Optimization.** Our canonical Optimization helps to generate high-quality subject (see the hand and face).

## 5.4 User Study

To further validate that DIScene can generate high-quality 3D complex scenes, we conduct a user study. We provide 8 scenes, each containing the generation results from text description by DIScene and 5 other baselines, for users to evaluate. A total of 74 participants, 40% of whom were professionals in the fields of art design and 3D modeling, answered the following 4 questions for each case: Scene quality,input alignment,Geometric fidelity,Scene consistency, by rate on a scale from 1 to 10. User preferences for the generated 3D assets are reflected through the average scores from the trial, as shown in the Table 3, which demonstrates the effectiveness of DIScene.

## 5.5 Ablation Studies

*Canonical Space Optimization.* We transform individual objects to the global space for rendering to demonstrate the importance of the canonical space optimization. As Fig. 6 shows, without canonical optimization, it can be observed that the "wizard's hand" is missing, which is similar to the partial missing issue in GraphDreamer, as it does not incorporate a canonical space design. And the quality of various details on the objects also significantly deteriorates.

*Object-aware rendering.* Fig. 7 illustrates the effectiveness of our object-aware rendering. Without object-aware rendering, the geometry and appearance of sword is obviously affected by the wolf: the sword is unnaturally stretched, as well as silhouette and color in the tip of the sword are similar to wolf. Besides, the contact-like interaction (See in Fig. 7 2(a-b)) usually involves a mixture of Gaussians from different objects. The original rendering strategy tends to produce blurry results in these areas, making $\mathcal{L}_{\text{SDS}}$ hard to provide correct guidance on interaction areas. However, with our object-aware rendering, the objects do not influence each other for better object generation quality, allowing for a clear modeling of interaction surfaces.

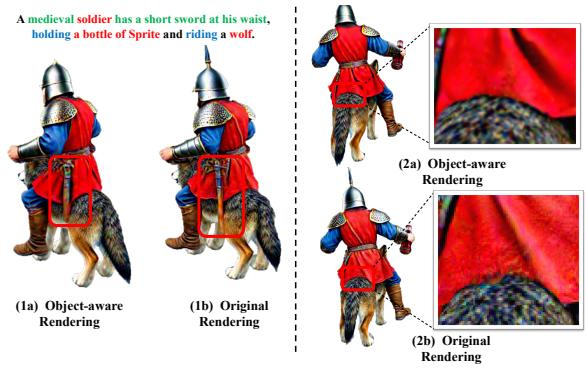



Fig. 7. **Ablation study on Object-aware Rendering.** Our object-aware rendering reduces the mutual influence of adjacent objects ((1a) and (1b)) and provides a clearer interaction surface definition ((2a) and (2b)).

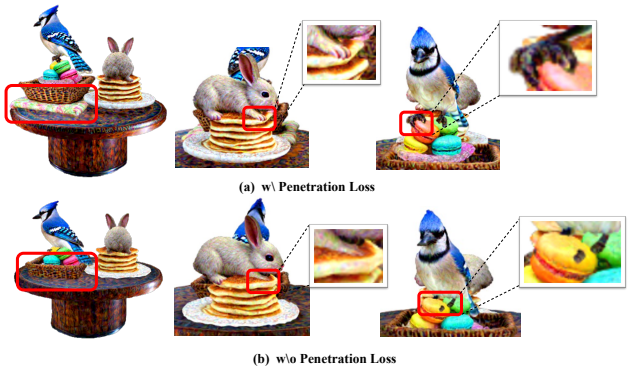



Fig. 8. **Ablation study on Penetration Loss.** The penetration loss enables a more precise modeling of interaction surfaces between objects, facilitating the generation of multi-object interactions with a realistic physical feel.

Table 4. Quantitative results on ablation studies.

| ELO Scores | w/o Canonical space | w/o Penetration Loss | w/o Object-aware rendering | Full Method |
|---|---|---|---|---|
| Mean $\times 10^3 \uparrow$ | 1.006 | 0.982 | 1.249 | **1.620** |

*Penetration Loss.* To verify the effectiveness of the Penetration Loss in modeling object interactions, we removed $\mathcal{L}_{penetr}$ during the optimization process. We could observe a significant decrease in the realism of the generated interactions in Fig. 8, confirming the effectiveness of Penetration Loss, which finely controls the physical interactions between objects and better model the interfaces.

Additionally, we report the quantitative evaluations with mean GPTEval3D ELO Scores on the above three claims in Table 4. The absence of any of them will bring a significant drop in numerical results, which provides more solid evidence on the effectiveness of DIScene.

## 6 Conclusion

In this paper, we first reconsider and define the complex 3D scene generation with distilling knowledge from 2D diffusion models should function in real-world applications: accepting flexible inputs and generating decoupled objects, ensuring stylistic consistency, and capable of modeling interactions. Subsequently, we propose *DIScene*, a novel framework utilizing a learnable scene graph to model the entire scene with complex inputs decomposition, object decoupling and interaction guidance. Additionally, we introduce a novel multi-object rendering and multi scale optimization strategy to effectively generate scenes with significant scale differences and to model the interactions between objects. Our work could directly generate usable complex 3D scenes in mesh for movies, games and animations production, which bridges the complex 3D scene generation task in the research field with the industrial production pipelines, addressing challenges in the industrial 3D content creation workflow.

*Limitations.* Objects with complex structure may experience interpenetration, as seen in the "Superman" case, where Superman's cape intersects with the wolf or the motorcycle. As a future work, we could introduce methods similar to "as rigid as possible" into DIScene to control mesh deformation and avoid such interpenetration. Besides, DIScene is an optimization-based framework, which is relatively time-consuming.

*Failure cases.* DIScene is based on SDS optimization but with initialization, so it may have Janus problem with an very low occurrence rate of around 5% . Besides, initialization is very important for DIScene and in some extreme cases, if the initial posture and position of the object are too far from the reasonable state, it may lead to unrealistic and unreasonable final generation results. This can be greatly avoided by re-initialization and more accurate description. The probability of such extreme situations occurring is about 10%, and it can often be avoided by re-initialization within five times.

## Acknowledgments

We thank all the reviewers for their useful suggestions. This work was supported by the National Science and Technology Major Project (2021ZD0112902), the National Natural Science Foundation of China (62220106003), the Research Grant of Beijing Higher Institution Engineering Research Center, the Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, and the Tsinghua University Initiative Scientific Research Program.

## References

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. 2018. Learning Representations and Generative Models for 3D Point Clouds. In *International Conference on Learning Representations Workshop (ICLR Workshop)*. https://openreview.net/forum?id=r14RP5AUz

S. Alex, K. Misha, B. Daria, S. Christoph, I. Ksenia, and K. Nadiia. 2023. Deepfloyd if: A Modular Cascaded Diffusion Model. https://github.com/deep-floyd/IF/tree/develop.

André Brock, Theodore Lim, James M. Ritchie, and Nick Weston. 2016. Generative and Discriminative Voxel Modeling with Convolutional Neural Networks. *arXiv preprint* (2016). http://arxiv.org/abs/1608.04236

Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. 2023. Generative Novel View Synthesis with 3D-Aware Diffusion Models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. 4194–4206.

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023. Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. 22189–22199.

Yongwei Chen, Tengfei Wang, Tong Wu, Xingang Pan, Kui Jia, and Ziwei Liu. 2024b. ComboVerse: Compositional 3D Assets Creation Using Spatially-Aware Diffusion Guidance. In *European Conference on Computer Vision (ECCV)*.

Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. 2024a. Text-to-3D using Gaussian Splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 21401–21412.

Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. 2023. LucidDreamer: Domain-free Generation of 3D Gaussian Splatting Scenes. *arXiv preprint* (2023). https://doi.org/10.48550/arXiv.2311.13384

Dana Cohen-Bar, Elad Richardson, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. 2023. Set-the-Scene: Global-Local Training for Generating Controllable NeRF Scenes. In *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2912–2921.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2023. Objaverse: A Universe of Annotated 3D Objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 13142–13153.

Dave Epstein, Ben Poole, Ben Mildenhall, Alexei A. Efros, and Aleksander Holynski. 2024. Disentangled 3D Scene Generation with Layout Learning. In *International Conference on Machine Learning (ICML)*, Vol. 235. 12547–12559.

Matheus Gadelha, Subhransu Maji, and Rui Wang. 2017. 3D Shape Induction from 2D Views of Multiple Objects. In *International Conference on 3D Vision (3DV)*. 402–411.

Gege Gao, Weiyang Liu, Anpei Chen, Andreas Geiger, and Bernhard Schölkopf. 2024. GraphDreamer: Compositional 3D Scene Synthesis from Scene Graphs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 21295–21304.

Antoine Guédon and Vincent Lepetit. 2024. SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5354–5363.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. 2024. LRM: Large Reconstruction Model for Single Image to 3D. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=sllU8vvsFF

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *ACM SIGGRAPH 2024 Conference Papers (SIGGRAPH)*. 32:1–32:11.

Michael M. Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing (SGP)*, Vol. 256. 61–70.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (2023), 139:1–139:14.

Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2019. Synthesizing 3D Shapes From Silhouette Image Collections Using Multi-Projection Generative Adversarial Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5535–5544.

Zhiqi Li, Yiming Chen, Lingzhe Zhao, and Peidong Liu. 2024. Controllable Text-to-3D Generation via Surface-Aligned Gaussian Splatting. *arXiv preprint* (2024). https://doi.org/10.48550/arXiv.2403.09981

Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. 2024. LucidDreamer: Towards High-Fidelity Text-to-3D Generation via Interval Score Matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6517–6526.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023b. Magic3D: High-Resolution Text-to-3D Content Creation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 300–309.

Yiqi Lin, Haotian Bai, Sijia Li, Haonan Lu, Xiaodong Lin, Hui Xiong, and Lin Wang. 2023a. CompoNeRF: Text-guided Multi-object Compositional NeRF with Editable 3D Scene Layout. *arXiv preprint* (2023). https://doi.org/10.48550/arXiv.2303.13843

Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12663–12673.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2022. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2022), 99–106.

Tai-Jiang Mu, Hao-Xiang Chen, Junxiong Cai, and Ning Guo. 2023. Neural 3D reconstruction from sparse views using geometric priors. *Computational Visual Media* 9, 4 (2023), 687–697.

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved Denoising Diffusion Probabilistic Models. In *International Conference on Machine Learning (ICML)*, Vol. 139. 8162–8171.

Michael Niemeyer and Andreas Geiger. 2021. GIRAFFE: Representing Scenes As Compositional Generative Neural Feature Fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11453–11464.

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint* (2023). https://doi.org/10.48550/arXiv.2303.08774

Ryan Po and Gordon Wetzstein. 2024. Compositional 3D Scene Generation using Locally Conditioned Diffusion. In *International Conference on 3D Vision (3DV)*. 651–663.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=FjNys5c7VyY

Albert Pumarola, Stefan Popov, Francesc Moreno-Noguer, and Vittorio Ferrari. 2020. C-Flow: Conditional Generative Flow Models for Images and 3D Point Clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7946–7955.

Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. 2024. Magic123: One Image to High-Quality 3D Object Generation Using Both 2D and 3D Diffusion Priors. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=0jHkUDyEO9

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, Vol. 139. 8748–8763.

Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan T. Barron, Yuanzhen Li, and Varun Jampani. 2023. DreamBooth3D: Subject-Driven Text-to-3D Generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2349–2359.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-Shot Text-to-Image Generation. In *International Conference on Machine Learning (ICML)*, Vol. 139. 8821–8831.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022a. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10674–10685.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022b. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10674–10685.

Leonid I. Rudin and Stanley J. Osher. 1994. Total Variation Based Image Restoration with Free Local Constraints. In *International Conference on Image Processing (ICIP)*. 31–35.

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. 2021. Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 6087–6101.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. 2024. MVDream: Multi-view Diffusion for 3D Generation. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=FUgrjq2pbB

Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. 2024. DreamCraft3D: Hierarchical 3D Generation with Bootstrapped Diffusion Prior. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=DDX1u29Gqr

Qinghong Sun, Yangguang Li, ZeXiang Liu, Xiaoshui Huang, Fenggang Liu, Xihui Liu, Wanli Ouyang, and Jing Shao. 2023. UniG3D: A Unified 3D Object Generation Dataset. *arXiv preprint* (2023). https://doi.org/10.48550/arXiv.2306.10730

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. 2024a. LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation. In *European Conference Computer Vision (ECCV)*.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. 2024b. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=UyNXMqnN3c

Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. 2023. MVDiffusion: Enabling Holistic Multi-view Image Generation with Correspondence-Aware Diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36. 51202–51233.

Konstantinos Tertikas, Despoina Paschalidou, Boxiao Pan, Jeong Joon Park, Mikaela Angelina Uy, Ioannis Z. Emiris, Yannis Avrithis, and Leonidas J. Guibas. 2023. Generating Part-Aware Editable 3D Shapes without 3D Supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4466–4478.

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. 2024. SV3D: Novel Multi-view Synthesis and 3D Generation from a Single Image using Latent Video Diffusion. In *European Conference on Computer Vision (ECCV)*.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 27171–27183.

Peng Wang and Yichun Shi. 2023. ImageDream: Image-Prompt Multi-view Diffusion for 3D Generation. *arXiv preprint* (2023). https://doi.org/10.48550/arXiv.2312.02201

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. 2023. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36. 8406–8441.

Karl D. D. Willis, Pradeep Kumar Jayaraman, Hang Chu, Yunsheng Tian, Yifei Li, Daniele Grandi, Aditya Sanghi, Linh Tran, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. 2022. JoinABLe: Learning Bottom-up Assembly of Parametric CAD Joints. In *IEEE/CVF Conference on Computer Vision and Pattern*

*Recognition (CVPR)*. 15828–15839.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. 2016. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 29. 82–90.

Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. 2024a. GPT-4V(ision) is a Human-Aligned Evaluator for Text-to-3D Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 22227–22238.

Tong Wu, Yu-Jie Yuan, Ling-Xiao Zhang, Jie Yang, Yan-Pei Cao, Ling-Qi Yan, and Lin Gao. 2024b. Recent Advances in 3D Gaussian Splatting. *Computational Visual Media* (2024). https://doi.org/10.1007/s41095-024-0436-y

Qun-Ce Xu, Tai-Jiang Mu, and Yong-Liang Yang. 2023. A survey of deep learning-based 3D shape generation. *Computational Visual Media* 9, 3 (2023), 407–442.

Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. 2024. GRM: Large Gaussian Reconstruction Model for Efficient 3D Reconstruction and Generation. *arXiv preprint* (2024). https://doi.org/10.48550/arXiv.2403.14621

Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. 2024. GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6796–6807.

Zehao Yu, Torsten Sattler, and Andreas Geiger. 2024. Gaussian Opacity Fields: Efficient Adaptive Surface Reconstruction in Unbounded Scenes. *ACM Transactions on Graphics* (2024).

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. 3813–3824.

Xiaoyu Zhou, Xingjian Ran, Yajiao Xiong, Jinlin He, Zhiwei Lin, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. 2024. GALA3D: Towards Text-to-3D Complex Scene Generation via Layout-guided Generative Gaussian Splatting. In *International Conference on Machine Learning (ICML)*, Vol. 235. 62108–62118.

Junzhe Zhu, Peiye Zhuang, and Sanmi Koyejo. 2024. HIFA: High-fidelity Text-to-3D Generation with Advanced Diffusion Guidance. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=IZMPWmcS3H