CG3D: Compositional Generation for Text-to-3D via Gaussian Splatting

Alexander Vilesov*   Pradyumna Chari*   Achuta Kadambi

University of California, Los Angeles

{vilesov, pradyumnac}@ucla.edu, achuta@ee.ucla.edu

A stool with a lamp, and a table with a potted plant and a banana, next to a living room couch.

(a) DREAMGAUSSIAN [44]   (b) FANTASIA3D [9]   (c) OURS

Figure 1. **We propose a method for scalable, composable 3D generation from text prompts only.** Prior methods are unable to generate scenes consistent with detailed text, while the proposed method leverages explicit representations to enable physically correct compositions, without any additional changes to the guiding diffusion model.

Abstract

With the onset of diffusion-based generative models and their ability to generate text-conditioned images, content generation has received a massive invigoration. Recently, these models have been shown to provide useful guidance for the generation of 3D graphics assets. However, existing work in text-conditioned 3D generation faces fundamental constraints: (i) inability to generate detailed, multi-object scenes, (ii) inability to textually control multi-object configurations, and (iii) physically realistic scene composition. In this work, we propose CG3D, a method for compositionally generating scalable 3D assets that resolves these constraints. We find that explicit Gaussian radiance fields, parameterized to allow for compositions of objects, possess the capability to enable semantically and physically consistent scenes. By utilizing a guidance framework built around this explicit representation, we show state of the art results, capable of even exceeding the guiding diffusion model in terms of object combinations and physics accuracy. Project

webpage: https://asvilesov.github.io/CG3D/

1. Introduction

Current generative text-to-3D methods are incapable of producing scene-level results. While such methods have achieved remarkable performance at the object-level, scene-level prompts result in only parts of the scene being generated or complete failure to semantically adhere to the prompt. In this work we propose compositional scene generation from text by harnessing explicit 3D representations. Explicit representations decouple objects from the scene thus allowing users high flexibility for composition or editing at the object and scene level.

We build upon a large body of existing work to achieve this. With the arrival of image diffusion models trained on large scale datasets, users have been able to create a variety of content. The control over the generation process now encompasses a range of inputs as a form of conditioning which include text [12, 14], images [56], edge outlines [55], and masks [3]. The generative field is rapidly evolving

---

*Indicates equal contribution.

with the aims of expanding to image reconstruction [13], video [7, 48], audio [18], and 3D objects [9, 19, 30]. 3D object synthesis is a complex task which requires the fabricated object to be consistent with a given text prompt from any view. Several pioneering works [16, 25, 37] attempted this through various forms of CLIP guidance. This was advanced by Poole et al. [30] through guiding the formation of Neural Radiance Field (NeRF) with a pre-trained image diffusion model. Subsequent works have focused on improving quality, diversity of objects, and extending to other 3D formats. However, few efforts toward text-to-3D generation have addressed compositionality. We aim to fill this gap by introducing a composition-based method that leverages explicit 3D representations. We summarize our major contributions below:

1. A framework for compositional generation of scalable scenes using explicit radiance fields
2. Physically realistic scene generation while maintaining object separability and fidelity
3. A method to estimate object composition parameters (rotation, translation, scale) through score distillation sampling, without the need for object bounding boxes

2. Related Works

2.1. 3D Scene Representation

A differentiable representation for a 3D scene is key to represent a wide range of scenes and to enable subsequent tasks such as text-to-3D generation. A common method in recent years has been Neural Radiance Fields (NeRFs) [24] which represent 3D scenes with a coordinate-based network that can be easily queried for novel view synthesis. Subsequent NeRF-based works have aimed at improving reconstruction quality [5, 22, 39, 46], speed of training [26], representing large-scale scenes [41], and achieving reconstruction under constrained conditions [1, 45]. Recently, Kerbl et al. [17] proposed an explicit representation for novel-view synthesis, that uses splatting [58] of 3D anisotropic Gaussians to improve rendering speed. As opposed to these works, our focus is on *generating* multi-object compositional scenes from text prompts, rather than representing existing scenes.

2.2. Text-guided generation using diffusion

Recent progress in image generation has been spear-headed by diffusion models [15]. Such models have been adapted to take in complex prompts and generate high-quality images and scenes closely aligning to the given prompt [3, 14, 34, 35]. Song et al. [40] devised a new sampling method to accelerate sampling through denoising diffusion implicit models. The generation of high-resolution images is achieved through a cascade of super-resolution models [4, 36] or performing the diffusion process in a low-resolution latent space where the resulting latents are de-coded to image space [34]. Text guidance was originally achieved through an explicit classifier model [12] and later through classifier free guidance (CFG) [14]. More elaborate guidance schemes have been devised to include additional inputs such as sketches and poses [55] for more control.

With such progress in image generation, interest has surged in text-to-3D generation. Earlier works such as CLIP-forge [37], DreamField [16], and CLIP-mesh [25] optimized their 3D models by maximizing text-image alignments scores. However, diffusion guidance was adapted to generate more detailed 3D primitives through the use of Score Distillation Sampling (SDS) [30]. Ensuing work like Magic3D [19] has improved qualitative results by adopting two-stage training where a course NeRF is first learned and then converted to a DMTET representation for further optimization, Fantasia3D [9] decouples geometry and material using physically-based rendering to create high-quality meshes, and ProlificDreamer [47] introduced variational score distillation to improve upon the low-diversity of samples generated with SDS. We would like to highlight concurrent work (on ArXiv) focused on 3D asset generation using the 3D Gaussian Splatting framework that aim to achieve fast generation of 3D assets [44] and high quality 3D assets by incorporating 3D diffusion priors [10, 54]. Compared to all these prior methods, our focus is on generating physically realistic compositional 3D scenes from text input only, while allowing for individual object control.

2.3. Compositional 3D generation

Compositional 3D generation entails generating a cohesive scene from several lower-level primitives. For the purposes of this paper, lower-level primitives will be objects. One of the first steps to scene synthesis is defining the spatial relationships between objects which is typically done with scene graphs [8, 33, 57]. Traditionally, one of the main applications of scene synthesis has been room layouts, with early approaches constraining the problem using interior design rules and object frequency distributions [23, 53]. Newer methods adopt the generative machine learning paradigm using VAEs [31, 51], GANs [52], and Diffusion models [43], where scene priors are learned instead of handcrafted such that scene generation models are automatic and end-to-end. Other works have focused on composition through decomposition Niemeyer and Geiger [28], Yang et al. [50]. Several works have begun to explore the usage of pre-trained image diffusion models to expand the range of compositions [11, 21, 29] by constraining geometry for each object with a user-defined bounding box. Our focus is on generating scalable scenes in a realistic manner, from text input only without bounding-boxes or training a diffusion model.

```
class Object:
    def __init__(gaussians):
        self.N = gaussians
    def transform(P):
        self.N = P(self.N)

class Composition:
    def __init__(objects, poses):
        self.N = []
        for i in range(len(objects)):
            objects[i].transform(poses[i])
            self.N.append(objects[i].N)
```
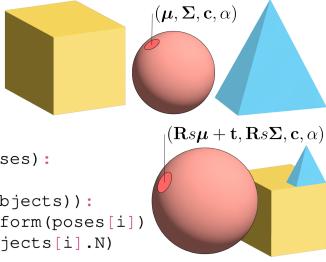
Figure 2. **We realize multi-object scenes through a Gaussian radiance field.** Pseudocode to enable compositionality in Gaussian radiance fields incorporating rotation, translation, and scale to convert 3D Gaussians from object to composition coordinates.

3. Compositional generation

Our goal is to generate a compositional 3D scene $\mathcal{S}$ given a text prompt $\mathbf{y}$, and access to a 2D image diffusion model. We consider a general scene $\mathcal{S}$ consisting of a set of K objects, $\mathcal{O} = \{\mathbf{O}_i \mid i \in [1, ..., K]\}$. Each object in $\mathcal{O}$ is represented by a set of Gaussians, θ. An image, $\mathbf{X}$, can then be generated by rendering the set of 3D Gaussians, $\mathbf{X} = \mathcal{R}\{\theta\}$, using Gaussian splatting [17]. The value of a pixel at location $\mathbf{p}$ is found by projecting each Gaussian onto the image plane and alpha-blending their contributions [58]:

$$\mathcal{R}\{\mathbf{p}, \mathbf{M}, \theta\} = \sum_{i \in \theta} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1}(1 - \alpha_j),$$

$$\text{with } \alpha_i = o_i e^{-\frac{1}{2}(\mathbf{p}-\underline{\mu}_i)^T \underline{\Sigma}^{-1}(\mathbf{p}-\underline{\mu}_i)},$$

$$(1)$$

where $\mathbf{c}_i$, o_i, $\boldsymbol{\mu}_i$, and $\boldsymbol{\Sigma}_i$ represent the color, opacity, position, and covariance, respectively, of the ith 3D Gaussian. Additionally, $\underline{\boldsymbol{\mu}}_i$ and $\underline{\boldsymbol{\Sigma}}_i$ represent the 2D projected counterparts of the mean and covariance, in screen-space coordinates.

To render a compositional scene, we require a transformation from object to composition coordinates, consisting of a rotation $\mathbf{R}$, translation $\mathbf{t}$, and scale s. These transformations apply to the individual Gaussian means and variances. We therefore define the scene to also contain the interactions between objects, $\mathcal{P} = \{\mathbf{P}_{i,j} \mid \forall i,j \in [1, ..., K], \ i \neq j\}$, such that $\mathcal{S} = \{\mathcal{O}, \mathcal{P}\}$. Here, $\mathbf{P}_{i,j} = (\mathbf{R}_{i,j}, \mathbf{t}_{i,j}, s_{i,j})$. Fig. 2 illustrates this radiance field structure and shows how this representation combines multiple objects into one data structure.

Such an explicit compositional representation is key to enable scalable scene generation. The input to our method is the text prompt $\mathbf{y}$ manually deconstructed by the user into a scene graph. The scene graph's set of object nodes and interaction nodes are textual descriptions of the objects and interactions between them, respectively, thus forming a one-to-one correspondence to $\mathcal{O}$ and $\mathcal{P}$ in $\mathcal{S}$. That is, $\mathbf{y}$ is represented by,

$$\mathbf{y} = \{\mathbf{y}_k | k \in [1, \ldots, K]\} \cup \{\mathbf{y}_{i,j} | i,j \text{ s.t. } \mathbf{P}_{i,j} \in \mathcal{P}\}, \quad (2)$$

where $\mathbf{y}_k$ is the textual description for object $\mathbf{O}_k$, and $\mathbf{y}_{i,j}$ is the textual description of the interaction between objects $\mathbf{O}_i$ and $\mathbf{O}_j$.

We wish to maximize the probability of a compositional scene given a text prompt,

$$p(\mathcal{R}\{\mathcal{S}\}|\mathbf{y}) = p(\mathcal{R}\{\mathcal{O}, \mathcal{P}\}|\mathbf{y}). \quad (3)$$

$\mathcal{R}\{\cdot\}$ denotes the rendering operator, which we will drop subsequently for notational convenience. This joint probability can be decomposed as follows:

$$p(\mathcal{O}, \mathcal{P}|\mathbf{y}) = p(\mathcal{O}|\mathbf{y}) \cdot p(\mathcal{P}|\mathcal{O}, \mathbf{y}). \quad (4)$$

The textual scene graph can then be interpreted as a probabilistic graphical model (PGM) where the directed edges are transformed such that the tails are at the object nodes and the heads are at the interaction nodes as illustrated in Fig. 3. With this formulation we make two assumptions: (a) the objects in $\mathcal{O}$ are independent of each other, and (b) the interaction between two objects $\mathbf{P}_{i,j}$ are conditionally dependent only on their connected objects $\mathbf{O}_i$ and $\mathbf{O}_j$. These assumptions arise out of the inability of image diffusion models (which we use for guidance) to generate and represent scenes with a large number of objects. Therefore our main formulation is as follows:

$$p(\mathcal{S}|\mathbf{y}) = \prod_{k=1}^{K} \underbrace{p(\mathbf{O}_k|\mathbf{y}_k)}_{\text{Sec. 3.2}} \cdot$$

$$\prod_{i,j \text{ s.t. } \mathbf{P}_{i,j} \in \mathcal{P}} \underbrace{p(\mathbf{R}_{i,j}, \mathbf{t}_{i,j}, s_{i,j}|\mathbf{y}_{i,j}, \mathbf{O}_i, \mathbf{O}_j)}_{\text{Sec. 3.1}}. \quad (5)$$
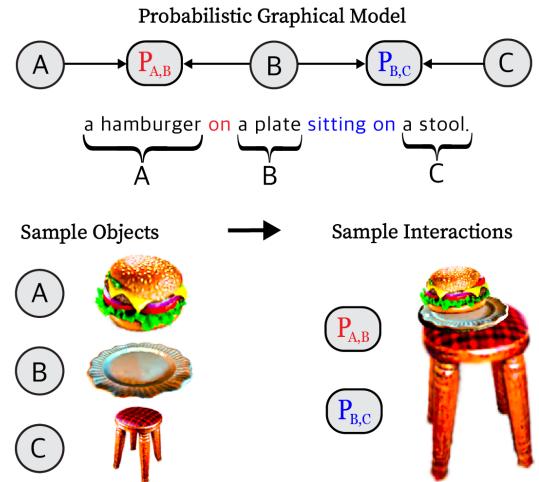

Figure 3. **Our method achieves compositional generation through ancestral sampling of a PGM of the scene.** We first sample objects followed by their pairwise interactions.
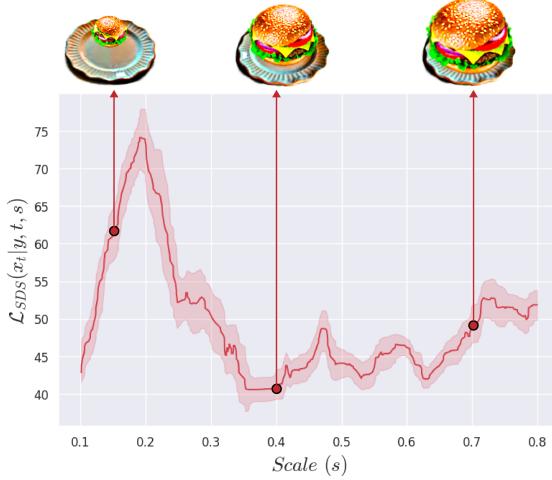
Figure 4. **Gradient descent optimization is poorly conditioned for estimating optimal $\mathbf{R}_{2,1}$, $s_{2,1}$ and $\mathbf{t}_{2,1}$.** Here, we show an anomaly in the SDS loss for unnaturally small $s_{2,1}$. Similar anomalies exist in the estimation of $\mathbf{t}_{2,1}$.

Through this formulation, we can generate a scene through ancestral sampling, by first generating the objects (Sec. 3.2) followed by their interactions (Sec. 3.1). We note that such a formulation creates limitations that constrain generation abilities: (i) inability to represent interactions that require object geometry changes, (ii) explicit knowledge of all objects and interactions, and (iii) intersection events that may occur between two objects on separate branches of the graph. However, as our results show, the expressivity of this representation remains high, and able to cover a range of scene configurations.

3.1. Interaction Parameter Estimation

Without loss of generality, let us consider a two-object scene with objects $\mathbf{O}_1$ and $\mathbf{O}_2$. We set $\mathbf{O}_1$ to be the anchor object, with respect to which the interaction parameters for $\mathbf{O}_2$, $(\mathbf{R}_{2,1},\mathbf{t}_{2,1}, s_{2,1})$, are defined. The anchor object, by definition, does not move. Our goal is to sample the interaction parameters such that,

$$(\mathbf{R}_{2,1}^*, \mathbf{t}_{2,1}^*, s_{2,1}^*) =$$
$$\underset{\mathbf{R}_{2,1},\mathbf{t}_{2,1},s_{2,1}}{\arg\max}\ p(\mathbf{R}_{2,1},\mathbf{t}_{2,1}, s_{2,1}|\mathbf{y}_{2,1}, \mathbf{O}_1, \mathbf{O}_2). \quad (6)$$

3.1.1 Score Distillation Sampling for Interaction Parameters

We wish to infer configuration parameters consistent with text guidance $\mathbf{y}_{2,1}$. We begin by using Score Distillation Sampling (SDS) to estimate these parameters. SDS provides a gradient update to guide generation through:

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{SDS}(\boldsymbol{\theta}) = \mathbb{E}_{t,\epsilon}\left[w(t)(\epsilon_\psi(\mathbf{X}_t|\mathbf{y},t) - \epsilon)\frac{\partial \mathbf{X}}{\partial \boldsymbol{\theta}}\right], \quad (7)$$

where ϵ_ψ is the denoising function of the diffusion model ψ that predicts sampled noise ϵ applied to an image X_t at a particular time-step t and text-prompt $\mathbf{y}$. Deriving intuition from Poole et al. [30], we view $\mathcal{L}_{SDS}$ as an estimate of the likelihood of an image scene. However, we interpret it as a function of $\mathbf{R}_{2,1}, \mathbf{t}_{2,1}, s_{2,1}$ and denote this function as $\mathcal{F}$:

$$(\mathbf{R}_{2,1}^*, \mathbf{t}_{2,1}^*, s_{2,1}^*) = \underset{\mathbf{R}_{2,1},\mathbf{t}_{2,1},s_{2,1}}{\arg\min}\ \mathcal{F}. \quad (8)$$

We refer to $\mathcal{F}$, as a function of interaction parameters, to be the configurational liklihood function (CLF), since it indicates the viability of a particular scene configuration.

Optimizing Eq. (8) through stochastic gradient descent is a natural first attempt. However this is non-trivial: the CLF is found to be extremely noisy, and with specific behavioral quirks that make the configuration generation task especially difficult. Figure 4 shows the CLF for the prompt "A hamburger on a plate", varied across various scales of the hamburger. The following details may be inferred:
1. CLF provides an extremely noisy loss landscape leading to potentially getting stuck in local optima
2. There are multiple solutions (local minima) in terms of semantic viability
3. CLF provides inaccurate guidance at lower scales

A particularly unusual aspect is the inaccurate guidance at lower scales $s_{2,1}$. Specifically, we note that below a certain value of $s_{2,1}$, γ, the loss function value progressively decreases, leading to false optima at lower scales. This behavior is consistent across multiple compositions.

We observe a thematically similar behavior affecting the estimation of the translation $\mathbf{t}_{2,1}$. Configurations that involve $\mathbf{O}_2$ being occluded by $\mathbf{O}_1$ from the perspective of the camera viewpoints (such as in a plate under a table) lead to lower CLF values than configurations without such occlusions (even though these configurations may violate the provided text conditioning). While we defer a more detailed discussion of this behavior of the SDS to the supplement, we note that these anomalies pose a significant challenge to estimating configuration parameters. Without appropriately accounting for these effects, SDS-based sampling of $(\mathbf{R}_{2,1}, \mathbf{t}_{2,1}, s_{2,1})$ would unnaturally prefer small objects and in occluded configurations.

It is therefore impractical and inaccurate to apply gradient descent to solve Eq. (8). We propose a different approach: in the $s_{2,1}, \mathbf{t}_{2,1}$ space, we perform random Monte Carlo (MC) sampling to cover a greater representative span of the CLF landscape, and provide ourselves a better chance of arriving at global optima. We use an alternating optimization approach, as follows:
1. First, a joint Monte Carlo search in the $s_{2,1}, \mathbf{t}_{2,1}$ space
2. Freeze $s_{2,1}$, sample and search for $\mathbf{t}_{2,1}$
3. Freeze $\mathbf{t}_{2,1}$, sample and search for $s_{2,1}$
4. Repeat steps 2 and 3 $L = 3$ times while keeping $\mathbf{R}_{2,1}$ fixed

4

Figure 5. **Diffusion models, such as Stable Diffusion v2.1 [34] are unable to always adhere to physical laws such as gravity, even for image generation.** Additional physical guidance is required for realistic-looking scene compositions.

The proposed MC sampling accounts for the noisy loss landscape of the CLF, but is still susceptible to the scale and translation anomalies. To **correct for the scale anomaly**, we sample scale above the anomaly threshold γ, which we estimate. To ensure this does not limit our ability to handle scenes with smaller scales, we also enable camera radius reduction across steps of the CLF MC sampling, if the current $s_{2,1}$ estimate is close to the threshold.

To **accommodate the occlusion CLF anomaly**, we scale down the CLF $\mathcal{F}$ with a visibility function $v(s_{2,1}, \mathbf{t}_{2,1})$, when estimating $\mathbf{t}_{2,1}$. The function $v(\cdot)$ is designed to be very low in the case of occlusions and high otherwise, and is realized using the viewspace gradients of our compositional Gaussian field. A detailed description of our initialization technique may be found in the supplement.

3.1.2 Enabling Physically Accurate Composition

Optimizing for Eq. (8) provides an estimate for $\mathbf{R}_{2,1}^{*}, \mathbf{t}_{2,1}^{*}, s_{2,1}^{*}$. However, this estimate is still limited by the prior of the underlying base diffusion model. Most significant among these is the lack of explicit physical constraints. This manifests in the form of unrealistic image generation that ignores physics laws such as gravity and contact forces (Figure 5). For a 3D generation and configuration task such as ours, this leads to the estimated $\mathbf{P}_{2,1}$, being physically unrealistic.

Leveraging the explicit nature of our Gaussian radiance field, we propose a two-stage estimation: the first being guided by SDS (Sec. 3.1.1), and the second being a fine-tuning based on physics-enforcing constraints $\mathcal{L}_{\text{Phys}}$, along with SDS. Note that gradient descent on SDS can provide meaningful guidance at this stage, post the initialization. That is,

$$
\begin{aligned}
\mathbf{P}_{2,1}' = (\mathbf{R}_{2,1}', \mathbf{t}_{2,1}', s_{2,1}') &= \underbrace{\underset{\mathbf{R}_{2,1}, \mathbf{t}_{2,1}, s_{2,1}}{\arg\min} \mathcal{F}}_{\text{SDS init. (Sec. 3.1.1)}}, \\
(\mathbf{R}_{2,1}^{*}, \mathbf{t}_{2,1}^{*}, s_{2,1}^{*}) &= \underbrace{\underset{\mathbf{R}_{2,1}, \mathbf{t}_{2,1}, s_{2,1}}{\arg\min} \mathcal{F} + \mathcal{L}_{\text{Phys}}\big|_{\text{init. at } \mathbf{P}_{2,1}'}}_{\text{Phys. finetune}}.
\end{aligned}
\tag{9}
$$

$\mathcal{L}_{\text{Phys}}$ models two phenomena: gravity and normal contact
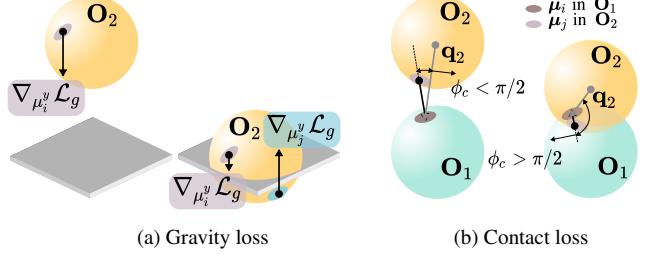

(a) Gravity loss (b) Contact loss

Figure 6. **Explicit representations enable physically realistic scene composition**. Consider spherical objects, made up of several Gaussians (represented by colored ellipses). (a) The **gravity loss** provides a gradient to move the object to the virtual floor without considerably penetrating the floor. (b) The **contact loss** prevent objects from unrealistically intersecting with each other, by minimizing the angle θ_c for intersecting points.

forces. Practically, at this step, we freeze the scale parameter $s_{2,1}$, and only optimize for the rotation and translation, for greater training stability.

Consider object $\mathbf{O}_1$. It consists of a set of 3D Gaussians, $\theta_{\mathbf{O}_1}$. We are concerned with the set of 3D Gaussian means, $\mathcal{K}_{\mathbf{O}_1} = \{\boldsymbol{\mu}_i | i \in \theta_{\mathbf{O}_1}\}$, where $\boldsymbol{\mu}_i = [\mu_i^x, \mu_i^y, \mu_i^z]$.

Gravity constraint $\mathcal{L}_g$: We define a floor for the scene as the lowermost point of anchor object $\mathbf{O}_1$. Intuitively, the gravity constraint applied on $\mathbf{O}_2$ enforces all Gaussians in $\mathbf{O}_2$ to move towards the floor without passing through it. The constraint has two distinct regimes of operation. When $\mathbf{O}_2$ is entirely above the floor, all Gaussians are guided to move towards the floor, via an absolute distance penalty with the floor height. When some Gaussians are below the floor, their absolute distance to the floor level is considered with a larger relative weighting, so as to pull $\mathbf{O}_1$ back above the floor. This is shown in Fig. 6(a).

Contact constraint $\mathcal{L}_c$: $\mathbf{O}_1$ and $\mathbf{O_2}$ are non-rigid Gaussian-represented objects. Without explicit constraints, they can intersect with each other, leading to unrealistic-looking scenes. Our key observation to enable contact constraints pertains to what we refer to as **the contact angle** $\phi_c^{\boldsymbol{\mu}_j}$ for a Gaussian $\boldsymbol{\mu}_j \in \mathcal{K}_{\mathbf{O}_2}$. For a Gaussian with mean $\boldsymbol{\mu}_j$ such that $j \in \mathbf{O}_2$, let $\boldsymbol{\mu}_i$ be the mean of the closest Gaussian in $\mathbf{O}_1$. Then, $\phi_c^{\boldsymbol{\mu}_j}$ is the angle between the vectors $\boldsymbol{\mu}_i - \mathbf{q}_2$ and $\boldsymbol{\mu}_i - \boldsymbol{\mu}_j$, where $\mathbf{q}_2$ is the center of $\mathbf{O}_2$. When $\boldsymbol{\mu}_j$ is not intersecting $\mathbf{O}_1$, $\phi_c^{\boldsymbol{\mu}_j} < \pi/2$ (Fig. 6(b), left side), and when $\boldsymbol{\mu}_j$ is intersecting $\mathbf{O}_1$, $\phi_c^{\boldsymbol{\mu}_j} > \pi/2$ (Fig. 6(b), right side). To avoid intersection, we enforce that the contact angle is acute for intersecting Gaussians by penalizing the negative cosine.

The overall physics constraint is therefore given by,

$$
\mathcal{L}_{\text{Phys}} = \lambda_g \mathcal{L}_g + \lambda_c \mathcal{L}_c, \tag{10}
$$

Prompt	Dream-Gaussian [43]	GS-Gen [9]	Fantasia-3D [8]	DreamFusion [29]*	Ours

A hamburger on a plate on a stool.

A bunny on a living room sofa.

A chair next to table.
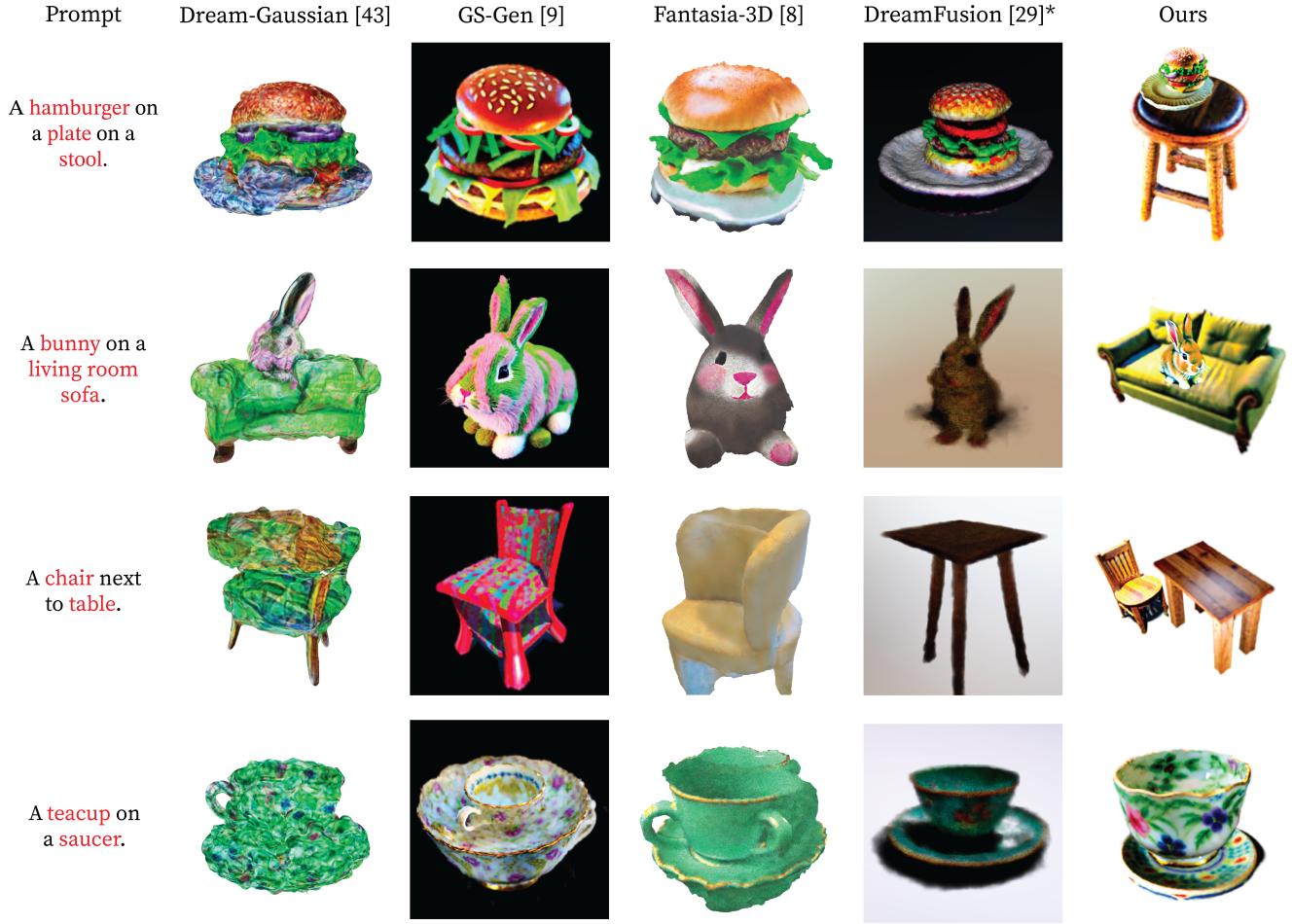
A teacup on a saucer.

Figure 7. **Compositional generation results across diverse scenes.** For each prompt, we highlight the individual objects taken as input to our method. For DreamFusion [30]*, we use a Stable Diffusion implementation [42] due to the original paper's proprietary model.

where λ_g and λ_c are appropriate regularization parameters.

In addition to avoiding overlap, we provide a second contact-based constraint. Namely, if the top-view cross section areas of $\mathbf{O}_1$ and $\mathbf{O}_2$ overlap in area above a threshold, and if $\mathbf{O}_1$ and $\mathbf{O}_2$ have come into contact, $\mathbf{O}_2$ receives a small impulse toward the central axis of $\mathbf{O}_1$. This constraint compensates for limitations in our configuration initialization through SDS sampling, where objects might end up not being perfectly aligned (arising out of improper priors or 3D to 2D projective ambiguities). Mathematical descriptions and implementation details for all the constraints may be found in the supplement.

3.2. Text-guided object generation with Gaussian Splatting

We sample the required objects in the composition through optimization of 3D Gaussians that can be rendered through splatting [17]. The primary objective is the SDS loss [30] that we rescale according to [20] to reduce over-exposures. In addition, we use efficient geometrical initialization of the

Gaussians [10], and auxillarly loss functions that distribute the Gaussians in an object in a more uniform manner. Before beginning optimization, we initialize a set of Gaussians similarly to Chen et al. [10], where a 3D diffusion model, Point-E [27], generates a sparse set of points, $\mathcal{E}$, that aligns with the given prompt. The points can then be replaced by a set of isotropic Gaussians before optimization begins. However, direct optimization with the SDS objective function can produce concavities (holes) that causes blotchy object appearance. We employ K nearest neighbors (KNN) losses to distribute Gaussians more uniformly across the surface:

$$\mathcal{L}_{\text{KNN}}(\theta_1, \theta_2) = \sum_{i \in \theta_1} \sum_{j \in \text{KNN}(i, \theta_2)} ||\mu_i - \mu_j - \min(\mathbf{\Sigma_i})||^2,$$

(11)

where θ_1 and θ_2 can be any arbitrary sets of Gaussians. To bring Gaussians to the surface we find the set of points that make up the Alpha hull of the object. A pair of points is considered part of the Alpha hull, if a line can be drawn between the pair such that a sphere of radius $1/\alpha$ contains no points in the set except for those two points on its bound-

6

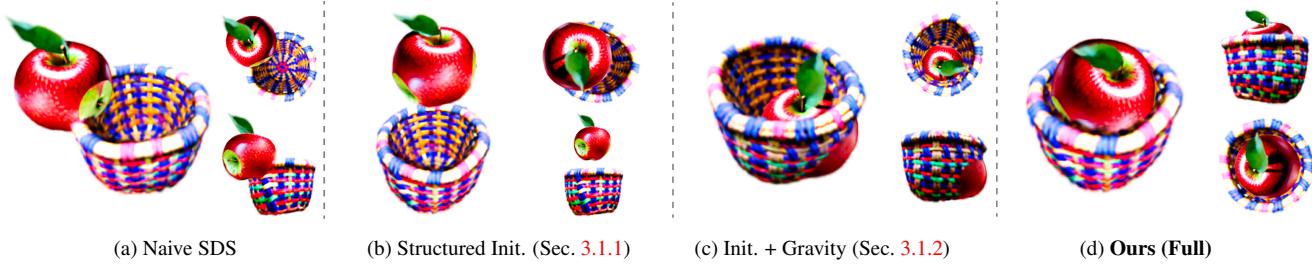|			
(a) Naive SDS	(b) Structured Init. (Sec. 3.1.1)	(c) Init. + Gravity (Sec. 3.1.2)	(d) **Ours (Full)**

Figure 8. **Ablation study for the composition method.** The prompt is "a photo of a red apple in a basket". **(a)** Naive SDS with random initialization, while **(b)** shows our initialization. **(c-d)** show the effect of the additional gravity and contact constraints.

ary. For efficiency, we determine the set of points belonging to the Alpha hull, $\mathcal{A}$, from a subset of all Gaussians using furthest point sampling [32]. The final objective function during the generation process is then:

$$\mathcal{L}_{3D} = \mathcal{L}_{SDS} + \beta \mathcal{L}_{\text{KNN}}(\theta, \mathcal{A}), \tag{12}$$

where β acts as a weight for the strength of the 3D regularizer. In addition, we add the option of regularizing the locations of Gaussians to be close to the Point-E initialization, $\mathcal{L}_{\text{KNN}}(\mathcal{N}, \mathcal{E})$, to preserve concavity or flatness of surfaces.

4. Experiments

In this section, we present the results of our method to evaluate its effectiveness. We begin with Sec. 4.1 where we show the ability of our method to generate compositional scenes while correctly predicting the pose and scale of objects with respect to each other in a semantically meaningful way. Next, in Sec. 4.2, we show that objects in a composition can be distilled for scene memory efficiency. In Sec. 4.3 we demonstrate the flexibility our model provides in editing and recomposing scenes. We conclude with an ablation study, Sec. 4.4, that justifies our design choices. Additional results are included in the supplement.

Implementation Details. Our method uses 3D Gaussian Splatting [17] and Stable Diffusion v2.1 [34]. Our method can be parallelized for G GPUs such that total generation time is $30\lceil K/G \rceil + 10P$ minutes for K objects and P interactions, where we use 4 Nvidia RTX3090 GPUs. The full set of hyperparameters for object generation and composition can be found in the supplement.

4.1. Zero-shot compositional generation

We compare our method's ability to generate 3D assets from language descriptions in Fig. 7 by comparing it to two state-of-the-art methods, Fantasia3D [9] and DreamFusion [30], that use NeRF and/or mesh optimization schemes and two recently proposed methods, GS-Gen [10] and DreamGaussian [44], which use Gaussian splatting. We note that the input between our method and the compared methods is different: while compared methods take in a plain text prompt,

ours method takes in the text prompt decomposed into a graph, as illustrated in Fig. 3. That being said, baseline methods often fail at generating the expected scene. The behaviour is to either ignore one of the items, or to fuse objects into one. For example, in row one, none of the baseline methods are successfully able to generate the stool. In the last row, we show a simpler prompt where other methods are able to generate both the saucer and teacup due to stronger diffusion guidance for the composition. Our method, generates all objects and synthesizes plausible poses and scales with respect to each other. It is important to note the detail on each object that arises out of our method, and which is notably absent from all existing baselines.
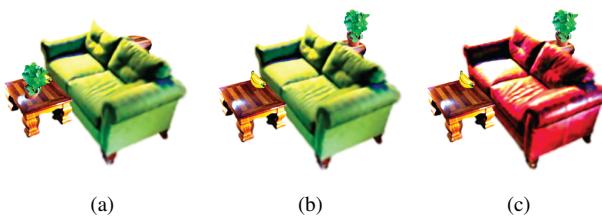
4.2. Radiance Field Distillation.

Due to the stochastic nature of optimizing Gaussian Radiance fields, duplication of Gaussians (splitting/cloning) is done regularly to enable expressability of the model at the cost of memory. Consequently, scenes generated with our method contain redundant Gaussians. We can distil our dense representation for each object by optimizing a new Gaussian radiance field trained on views from the original representation. We show quantitative results in Tab. 1.

4.3. Scene editing

Our method offers further control over composition through scene editing. Fig. 9 shows several examples where our model is able to delete an object from a scene, rearrange objects in the scene and edit an individual object in the scene. Object editing is simple since each object has its own distinct parameters which we can start optimizing from for a new prompt. Recomposition requires rerunning interactions in the compositional text graph that can potentially be affected by the replacement. Thus, a user is free to recompose a full scene to create new compositions or change just one of the interactions between objects.

4.4. Ablation studies

To evaluate the efficacy of our compositional method, we show the utility of each component evaluated on the composition of "a red apple in a basket", illustrated in Fig. 8.

(a)　　　　　(b)　　　　　(c)

Figure 9. **Our compositional, explicit representation allows for post-generation scene editing with high fidelity.** We use the scene from Fig. 1 to **(a):** delete the lamp from the stool, **(b) compositional editing:** move the plant from the table to the stool. **(c) object editing:** change the couch to "a red leather couch".
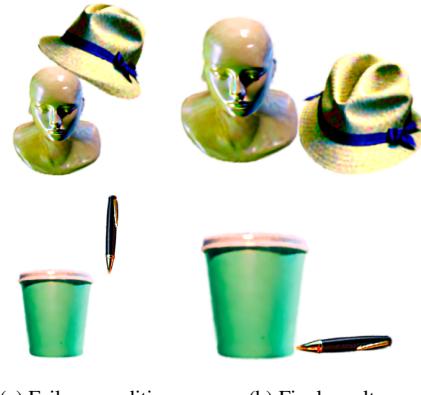
The right-most column shows the full model that includes the initialization strategy discussed in Sec. 3 and the physics constraints: gravity and contact. By using only SDS for optimization with a random initialization, interactions between objects appear far from the prompt due to the poor loss landscape mentioned in Sec. 3.1.1. Our initialization, on the other hand, has higher agreement with the given prompt. However, without the gravity constraint, we find that objects may often appear to be suspended in mid-air since the diffusion model may find this to be a plausible composition as illustrated in Fig. 5. The contact loss is important for preserving each object's individuality and to eliminate cases where objects are within each other. All components are vital to create physically realistic scenes and overcome biases in text-to-image diffusion models.

5. Limitations and Open Problems

3D scene generation from text is a difficult problem. While we propose a new framework for compositional generation, several limitations still exist. Our method assumes rigid-body interactions during composition, hence we do not support interactions where objects need to go into each other or require deformations. Here, we discuss two specific examples. We show in Fig. 10, a failure of an attempt at "a hat on a mannequin" where the hat slips off due to not perfectly fitting the head, and the lack of frictional forces in our current physical constraints. Additionally, our method is sensitive to initialization. For example, as shown in the "pen in a cup" scene, the pen had a poor initialization that led it to

(a) Failure conditions　　　(b) Final result

Figure 10. **Our physical constraints and initialization can fail under certain conditions.** The top and bottom rows show runs for "a hat on a mannequin" and "a black pen in a office cup".

not being in the cup. In this case, the failure arises as a result of our scheme to address the translation anomaly in the SDS loss Sec. 3.1.1. Our configuration estimation is also susceptible to SDS guidance: any anomalies in the diffusion model guidance will therefore reflect as failure cases. Our method also inherits common object generation problems such as the Janus problem [2] and geometry/texture ambiguities that may potentially confuse composition generation. Our method does not solve these issues, but several recent works can alleviate them [2, 38, 49].

6. Conclusion

We propose CG3D, a method for generating compositional 3D scenes using only text. We aim to fill the gap where prior methods often fail to generate coherent multi-object scenes. By utilizing explicit radiance fields, our method can create physically correct scenes, while maintaining object individuality and fidelity. Generated compositions can be edited and rearranged in under 15 minutes through text prompts alone, thus giving users flexibility and freedom to create more diverse scenes in less time. In future research, we aim to enable more elaborate interactions between objects and support large scale compositions with orders of magnitude higher number of objects.

Ethics Statement: Generative models have the potential to create harmful content. We condemn any such use for malicious purposes.

References

[1] Byeongjoo Ahn, Michael De Zeeuw, Ioannis Gkioulekas, and Aswin C Sankaranarayanan. Neural kaleidoscopic space sculpting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4349–4358, 2023. 2

Object	Original (#Gaussians)	Distilled (#Gaussians)	PSNR (dB)
Table	41,928	6,872	44.16
Teacup	48,370	15,856	39.22
Hamburger	51,937	21,955	36.57

Table 1. **Distillation of Gaussian radiance field.** For several objects from the Fig. 7 compositions, we report the number of Gaussians and PSNR between the original and distilled representations.

[2] Mohammadreza Armandpour, Huangjie Zheng, Ali Sadeghian, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. *arXiv preprint arXiv:2304.04968*, 2023. 8

[3] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 42 (4):1–11, 2023. 1, 2

[4] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2

[5] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[6] Ken Bellockk. Alphashape: Toolbox for constructing alpha shapes. 17

[7] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 2

[8] Angel Chang, Manolis Savva, and Christopher D Manning. Learning spatial knowledge for text to 3d scene generation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 2028–2038, 2014. 2

[9] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023. 1, 2, 7

[10] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *arXiv preprint arXiv:2309.16585*, 2023. 2, 6, 7, 17

[11] Xinhua Cheng, Tianyu Yang, Jianan Wang, Yu Li, Lei Zhang, Jian Zhang, and Li Yuan. Progressive3d: Progressively local editing for text-to-3d content creation with complex semantic prompts. *arXiv preprint arXiv:2310.11784*, 2023. 2

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 2

[13] Berthy T Feng, Jamie Smith, Michael Rubinstein, Huiwen Chang, Katherine L Bouman, and William T Freeman. Score-based diffusion models as principled priors for inverse imaging. *arXiv preprint arXiv:2304.11751*, 2023. 2

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1, 2

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[16] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022. 2

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 2, 3, 6, 7, 14

[18] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 2

[19] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 2

[20] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. *arXiv preprint arXiv:2305.08891*, 2023. 6, 17

[21] Yiqi Lin, Haotian Bai, Sijia Li, Haonan Lu, Xiaodong Lin, Hui Xiong, and Lin Wang. Componerf: Text-guided multi-object compositional nerf with editable 3d scene layout. *arXiv preprint arXiv:2303.13843*, 2023. 2

[22] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16252–16262, 2022. 2

[23] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. Interactive furniture layout using interior design guidelines. *ACM transactions on graphics (TOG)*, 30(4):1–10, 2011. 2

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[25] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 conference papers*, pages 1–8, 2022. 2

[26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2

[27] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 6, 17

[28] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 2

[29] Ryan Po and Gordon Wetzstein. Compositional 3d scene generation using locally conditioned diffusion. *ArXiv*, abs/2303.12218, 2023. 2

[30] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 4, 6, 7, 17

[31] Pulak Purkait, Christopher Zach, and Ian Reid. Sg-vae: Scene grammar variational autoencoder to generate new indoor scenes. In *European Conference on Computer Vision*, pages 155–171. Springer, 2020. 2

[32] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 7

[33] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5899–5908, 2018. 2

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 5, 7, 13, 15

[35] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 2

[36] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2

[37] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18603–18613, 2022. 2

[38] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*, 2023. 8

[39] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 2

[40] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[41] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 2

[42] Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. https://github.com/ashawkey/stable-dreamfusion. 6

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

[44] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 1, 2, 7

[45] Kushagra Tiwary, Akshat Dave, Nikhil Behari, Tzofi Klinghoffer, Ashok Veeraraghavan, and Ramesh Raskar. Orca: Glossy objects as radiance-field cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20773–20782, 2023. 2

[46] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 2

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

[48] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023. 2

[49] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20918, 2023. 8

[50] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021. 2

[51] Haitao Yang, Zaiwei Zhang, Siming Yan, Haibin Huang, Chongyang Ma, Yi Zheng, Chandrajit Bajaj, and Qixing Huang. Scene synthesis via uncertainty-driven attribute synchronization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5630–5640, 2021. 2

[52] Ming-Jia Yang, Yu-Xiao Guo, Bin Zhou, and Xin Tong. Indoor scene generation from a collection of semantic-segmented depth images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15203–15212, 2021. 2

[53] Yi-Ting Yeh, Lingfeng Yang, Matthew Watson, Noah D Goodman, and Pat Hanrahan. Synthesizing open worlds with

constraints using locally annealed reversible jump mcmc. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 2

[54] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussian-dreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arXiv preprint arXiv:2310.08529*, 2023. 2

[55] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1, 2

[56] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6027–6037, 2023. 1

[57] Ji-Zhao Zhu, Yan-Tao Jia, Jun Xu, Jian-Zhong Qiao, and Xue-Qi Cheng. Modeling the correlations of relations for knowledge graph embedding. *Journal of Computer Science and Technology*, 33:323–334, 2018. 2

[58] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378, 2001. 2, 3

CG3D: Compositional Generation for Text-to-3D via Gaussian Splatting

Supplementary Material

A. Supplemental Contents

This supplement is organized as follows:
- Score Distillation Sampling for Composition Guidance
- Incorporating Physical Losses for Composition
- Object Generation Implementation Details
- User Input Details
- Additional Results

B. Score Distillation Sampling for Composition Guidance

B.1. The scale anomaly

Secition 3.1.1 covers an initial description of the scale anomaly for configuration using SDS. Here, we will cover it in additional detail and explore additional aspects of the anomaly.

For the purpose of continuity, we will begin by redefining the anomaly. We refer to this anomaly as the monotonic decrease in the configurational function $\mathcal{F}$ with decreasing scale $s_{2,1}$. (**Note:** In the main paper, we refer to the function $\mathcal{F}$ as the configuration likelihood function, however the desired objective for this function is minimization rather than maximization (which is the case for likelihood functions usually).)
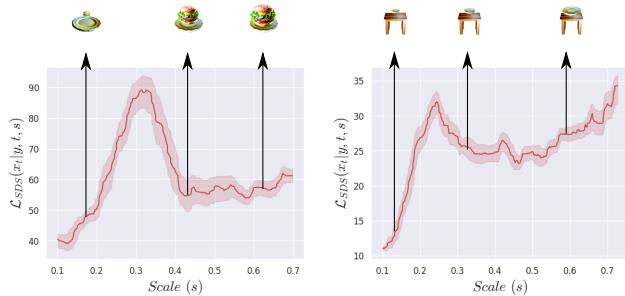

Figure A. **The scale anomaly exists across different object combinations.** For a fixed camera radius, the threshold below which the anomaly occurs, and nature of the anomaly are broadly consistent.

Effect of object variation: Figure A shows the anomaly for two different object configurations. We find the scale anomaly to be broadly consistent across objects. We also note that the threshold below which the scale anomaly occurs is broadly independent of the object, and depends on the relative size of the object in the rendered image. Since our object generation step generates objects of broadly the same size, the threshold remains constant regardless of the object.
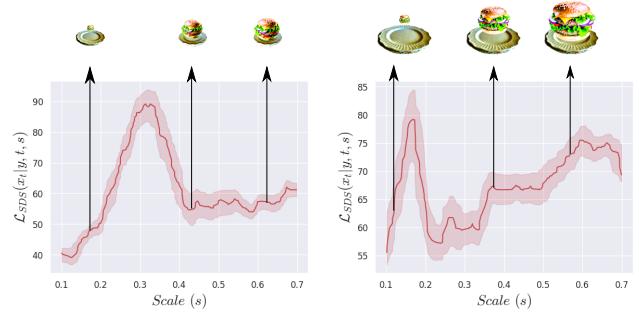

Figure B. **Effect of camera radius on the scale anomaly.** We find that the threshold for the scale anomaly moves to a lower scale value as the camera radius becomes smaller.

Effect of camera radius: Figure B shows the anomaly for two different camera radii. We find that the scale anomaly is affected by the radius. The smaller the camera radius from the center of the object, the lower is the scale below which the anomaly is seen. This observation is consistent with the hypothesis that the anomaly occurs when the relative object size is small in the rendered image.

Practical limitations for composition: The scale anomaly enforces limitations on SDS-based configuration estimation. Specifically, (a) if the camera radius is large enough, below a certain scale the $\mathcal{F}$ function will not provide accurate guidance (as a result of the scale anomaly), and (b) if the camera radius is small enough, above a certain scale the $\mathcal{F}$ function will not provide accurate guidance (since $\mathbf{O}_2$ will not be entirely visible in the rendered image).

B.2. The translation anomaly

Section 3.1.1 covers an initial description of the translation anomaly for configuration using SDS. Here, we will cover it in additional detail and explore additional aspects of the anomaly.

The translation anomaly is the reduction in the configurational function $\mathcal{F}$ for regions with object occlusion. That is, the value of $\mathcal{F}$ is anomalously lower if certain parts of the object are not visible to the camera.
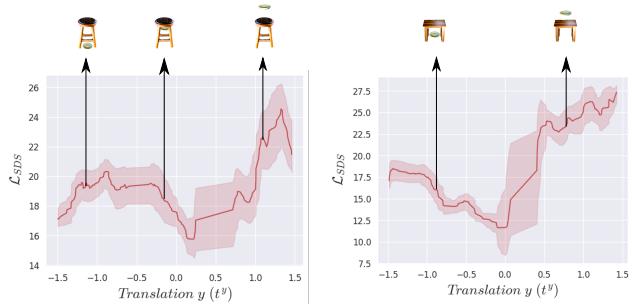
Figure C. **The translation anomaly for composition guidance.** The SDS loss observes lower values for regions with occlusion, such as below a table or a stool. This can lead to false optima in these occluded regions.

Figure D. **Our base diffusion model, Stable Diffusion v2.1 [34], prefers positive camera elevations (looking down from up).** (Left) For the prompt "a plate on a table", the generative prior considerably prefers top-down views. Therefore, for our composition, we do not use camera views with negative elevation (looking up from down) since these views do not have significant support in the domain of the generative prior (as shown on the right, for the scene "a plate on a table").

Effect of occlusion: Figure C shows the translation anomaly for three different object configurations. We find that the translation anomaly qualitatively exists across various object configurations. However, the nature of the anomaly varies across configurations in terms of the extent of the anomaly.

Practical limitations for composition: The following limitations to composition estimation are enforced by the translation anomaly: (a) occluded scenes (eg. a pen in a cup, a corgi under a table etc.) do not have accurate guidance from the $\mathcal{F}$ function, (b) the scale of the object affects the translation anomaly. Therefore, the $\mathcal{F}$ function would favor smaller, occluded objects as a result of the optimization.

B.3. Camera positioning

Another aspect that can affect the behavior of SDS-based configuration optimization is the camera positioning and view perspectives used. To understand this, it useful to understand canonical views that a diffusion-based image generation model is inclined towards generating, from the perspective of combinations of multiple objects. Figure D shows an image generated by the Stable Diffusion v2.1 model. Note how camera perspective has a positive elevation (looking down from up) in most compositional scenes. This is consistent with our observations that positively elevated camera angles lead to more stable composition.

Negative camera elevations: From the perspective of loss function design, it is intuitively beneficial to have symmetric camera positions. Therefore, along with positive camera elevations, we would ideally also prefer to have negative camera elevations. This might also help with addressing the translation anomaly. However, there is a problem. Figure D shows an example view from a negative elevation:

such views are not distinctive of the scene prior in the diffusion model, and therefore do not contain signal for compositional guidance. Therefore, we decide to limit our camera views to contain only positive camera elevation angles. Specifically, for our structured initialization, we use 8 fixed images, split across two elevations (30 and 60 degrees) and four azimuths (0, 90, 180 and 270 degrees).

B.4. Our structured initialization, in detail

As mentioned in the main paper, to address the aforementioned anomalies, simple Monte Carlo sampling is insufficient to identify an initial compositional configuration that is faithful to the given text prompt $\mathbf{y}$. At a high level we perform an alternating sampling of translation and scale, while keeping rotation fixed (the rotation gets optimized subsequently through the physics and SDS loss updates post initialization). That is,

1. First, joint initialization of scale and translation
2. Keeping scale fixed, choose a suitable translation
3. Keep translation fixed, choose a suitable scale
4. Repeat the above two steps $L = 3$ times, while keeping the rotation fixed

Algorithm 1 describes the structured initialization. We now explore both scale and translation initialization in detail.

B.4.1 Scale initialization

The scale anomaly is broadly consistent across object-pairs and depends on the camera radius for rendering (Figs. A and B). We find that the most stable approach therefore is to limit the candidate range of scales to exclude the anomalous region, for a given camera radius. In practice, we begin scale sampling at a camera radius of 4.5 units, by identify 50 random scales. The anchor object $\mathbf{O}_1$ has a fixed

---

Algorithm 1 Our structure initialization

Input: $\mathbf{O}_1, \mathbf{O}_2, \mathbf{y}_{1,2}$, M.C. sampler **MonteCaro**$\{\cdot\}$ ▷
   Objects, interaction text description, sampling operators
Output: $\mathbf{R}_{2,1}^*, \mathbf{t}_{2,1}^*, s_{2,1}^*$ ▷ Interaction parameters
   $N \leftarrow 3$ ▷ # of alternating optimization steps
   $n \leftarrow 1$ ▷ Current step
   $\mathbf{R}_{2,1}^* = [1, 0, 0, 0]$ ▷ Fix rotation to identity quaternion
   $\mathbf{t}_{2,1}^*, s_{2,1}^* = \underset{\mathbf{t}_{2,1}, s_{2,1}}{\textbf{MonteCarlo}}\{\mathcal{F}_{\text{Trans}}(\mathbf{R}_{2,1}^*, \mathbf{t}_{2,1}, s_{2,1})\}$ ▷
   Joint
   while $n \leq N$ **do**
      $\mathbf{t}_{2,1}^* = \underset{\mathbf{t}_{2,1}}{\textbf{MonteCarlo}}\{\mathcal{F}_{\text{Trans}}(\mathbf{R}_{2,1}^*, \mathbf{t}_{2,1}, s_{2,1}^*)\}$ ▷
   Transl.
      $s_{2,1}^* = \underset{s_{2,1}}{\textbf{MonteCarlo}}\{\mathcal{F}(\mathbf{R}_{2,1}^*, \mathbf{t}_{2,1}^*, s_{2,1})\}$
                                  ▷ Scale
      $n \leftarrow n + 1$
   end while

---

scale of 0.8, while the set of candidate scales for the section object $\mathbf{O}_2$ are chosen such that $s_{2,1} \in [0.3, 0.7]$. The desired scale is chosen as the average of the 5 scales with least values of the CLF $\mathcal{F}$. If the chosen scale is found to be within a threshold of 0.05 of the range lower limit (that is, if $s_{2,1} < 0.35$), we reduce the camera radius to 2.5 units and change the candidate scale range to $[0.2, 0.6]$. The lower range limits are chosen and fixed based on our scale analysis across several objects, and we find this approach to be more stable that one involving and adaptive estimate of the range lower limit (as a result of the noisy nature of the CLF $\mathcal{F}$). Therefore, our overall effective range of candidate scales is $[0.2, 0.7]$ with the anchor object $\mathbf{O}_1$ having a scale of 0.8. We find this range to be sufficient to accommodate a variety of composition requirements as evidenced through our results.

B.4.2 Translation initialization

Visibility function: For effective Monte Carlo search across translation, we need to account for the translation anomaly. To achieve this, we design a representative function for object visibility. Specifically, we use the viewspace gradients available within the 3D Gaussian Splatting framework [17]. Given an objective function, the viewspace gradients essentially supply the dependence of the objective on individual Gaussians. By defining said objective to be the rendered image (averaged across pixels and noised, for conditioning), we can get the contributions of each Gaussian towards the image. Averaging for all Gaussians in $\mathbf{O}_2$ is then our measure of object visibility $v(\cdot)$. There remains a problem with this visibility function. Since it is a pure measure of object contribution to an image, apparent visibility will reduce as the object is farther away from the cameras.

While this can be solved with symmetric cameras (in terms of elevation), we are unable to do that due to the nature of the diffusion guidance Sec. B.3. We find that defining the visibility function with an exponent $\gamma < 1$ reduces this effect favorably, in practice. Additionally, to nullify the dependence of the effective visibility function on the scale of the object (larger objects occupy more pixels in the image), we normalize with the square of the scale $s_{2,1}^2$. Therefore, our effective CLF for translation becomes,

$$\mathcal{F}_{\text{Trans}} = \mathcal{F} \cdot / \left(\frac{v}{s_{2,1}^2} \right)^{-\gamma}, \tag{13}$$

where v is the visibility function defined by us, in terms of he viewspace gradients, $s_{2,1}$ is the scale of $\mathbf{O}_2$, $\mathbf{F}$ is our originally defined CLF and γ is the exponent.

The initialization regime: At each step, we sample 50 points on a sphere around the object $\mathbf{O}_1$, to get candidate translations $\mathbf{t}_{2,1}$ (we choose this as $\mathbf{O}_1$ as generated with our object generation is smaller than this radius). Candidate translations are pruned to ignore samples where the two objects intersect, and when $\mathbf{O}_2$ is below the floor (lowermost point of $\mathbf{O}_1$). Post this, the best translation is chosen such that it minimizes the updated translation CLF $\mathcal{F}_{\text{Trans}}$.

B.4.3 Joint initialization

To accommodate joint initialization, we combine aspects of both scale and translation initialization. We sample 150 $\mathbf{t}_{2,1} - s_{2,1}$ combinations, being consistent with the smapling constraints of translation (pruning floor violations and intersections) and scale (limiting scale sampling to within the radius-specific range). We use the translation CLF $\mathcal{F}_{\text{Trans}}$ to identify the combination with highest likelihood. We do not average across the 5 best samples, as we do for scale initialization.

B.5. Success Rate of Composition

We find different object configurations to have different rates of success with regards to composition. We summarize our observations below.

B.5.1 Dependence on Diffusion Guidance

We find strong diffusion guidance to be the most important factor in determining the success of our composition step. We are therefore limited to object configurations and pairs for which the base diffusion model has a strong prior, and hence strong guidance.

Figure E. **The specific object instances can affect the composition parameters.** The apparently larger appearance of the table on the left (as a result of slender legs, etc.) leads to a smaller overall plate relative to it. The smaller appearance of the table on the right leads to a relatively larger plate.

B.5.2 Variance across object instance

Given a fixed prompt, score-based composition guidance can show considerable variation as a function of the individual objects. Here, we show two examples of composition for the prompt "a photo of a plate on a table". Specifically, we apply or composition technique across two different pairs of plates and tables. Across three trials, we consistently observe that our guidance infers a considerably larger size for one of the plates on the table, as opposed to the other. In other words, depending on the specific instance of a plate and a table, the SDS loss has a variable interpretation of their relative scales that it enforces. Figure E highlights this observation.

B.5.3 Close-up Versus Long-shot Compositons

Figure F. **Our base diffusion model, Stable Diffusion v2.1 [34], prefers object-centric, close-up views of scenes, which may hinder guidance for larger-scale scene composition.** (Left) "Wide shot, a night stand next to a bed". (Right) "wide shot, a plate on a table". In both cases, the second object (bed, table) does not have good visibility.

We find that the Stable Diffusion v2.1 guidance favors close-up camera views centered on the primary object, even for wider-scale scenes. Figure F shows two room-scale

scenes, for which the image remains relatively closer-up. In terms of composition, we find this to manifest in the form of more reliable guidance for close-up scenes ("a hamburger on a plate") than for wide-angle scenes ("a nightstand next to a bed").

B.5.4 Dependence on object-level Janus Effect

Figure G. **For the above scene "a nightstand next to a bed", object-level Janus effects can lead to multiple possible solutions for composition.** (Left) A viable composition for the scene, given the prompt. (Right) However, as a result of the Janus effect, rotating the azimuthal angle by $180°$ shows the possibility for a different viable configuration (with the nightstand further up along the bed from this perspective).

Object-level Janus effects can lead to multiple likely configurations. We show an example of this in Figure G shows the scene "a nightstand next to a bed". Janus effects on the bed can lead to multiple likely configurations, out of which one is arrived at through our composition.

B.5.5 Objects with Occlusions

As discussed in Section B.2, quirks in the SDS guidance lead to false optima at locations with occlusions (like "a pen in a cup"). Therefore, we are unable to support such configurations with occlusions through our guidance.

C. Physical Losses for Composition

We now describe the physics-based optimization step to enable realistic composition (Sec. 3.1.2, main paper). As described there, post the structured initialization, we aim to optimize the following loss function:

$$(\mathbf{R}_{2,1}^*, \mathbf{t}_{2,1}^*, s_{2,1}^*) = \underbrace{\arg\min_{\mathbf{R}_{2,1}, \mathbf{t}_{2,1}, s_{2,1}} \mathcal{F} + \mathcal{L}_{\text{Phys}}\big|_{\text{init. at } \mathbf{P}_{2,1}'}}_{\text{Phys. finetune}} \cdot \tag{14}$$

That is, after a structured initialization, we optimize the rotation, translation and scale of $\mathbf{O}_2$ with respect to $\mathbf{O}_1$ using the CLF regularized by physics losses.
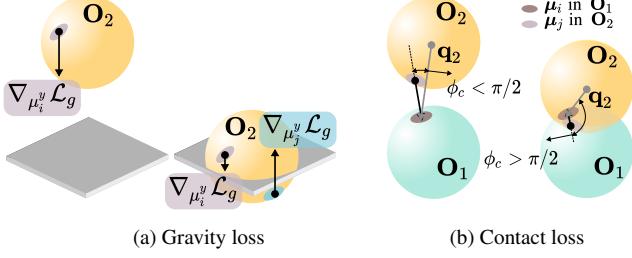
(a) Gravity loss   (b) Contact loss

Figure H. **Explicit representations enable physically realistic scene composition**. Consider spherical objects, made up of several Gaussians (represented by colored ellipses). (a) The **gravity loss** provides a gradient to move the object to the virtual floor without considerably penetrating the floor. (b) The **contact loss** prevent objects from unrealistically intersecting with each other, by minimizing the angle θ_c for intersecting points. Repeated from main paper for clarity.

Consider object $\mathbf{O}_1$. It consists of a set of 3D Gaussians, $\theta_{\mathbf{O}_1}$. We are concerned with the set of 3D Gaussian means, $\mathcal{K}_{\mathbf{O}_1} = \{\boldsymbol{\mu}_i | i \in \theta_{\mathbf{O}_1}\}$, where $\boldsymbol{\mu}_i = [\mu_i^x, \mu_i^y, \mu_i^z]$. Based on this, the geometric center is defined as $\mathbf{q}_1 = \frac{1}{|\theta_{\mathbf{O}_1}|} \sum_{\boldsymbol{\mu}_i \in \mathcal{K}_{\mathbf{O}_1}} \boldsymbol{\mu}_i$, where $|\cdot|$ is the cardinality operator. Similarly, we can define $\mathcal{K}_{\mathbf{O}_2}$ and associated parameters. The 'up' direction is along the y-axis.

Gravity loss $\mathcal{L}_g$: We define a floor for the scene as the lowermost point of $\mathbf{O}_1$. Specifically,

$$c^{\text{floor}} = \min \mu_i^y \, \forall \, \boldsymbol{\mu}_i \in \mathcal{K}_{\mathbf{O}_1}. \tag{15}$$

In practice, we choose the floor as the median of the y-coordinates of the lowest 0.1% of Gaussians in $\mathbf{O}_1$. Intuitively, the gravity loss applied on $\mathbf{O}_2$ must enforce all Gaussians in $\mathbf{O}_2$ to move towards the floor without passing through it. That is, the loss has two distinct regimes of operation. When $\mathbf{O}_2$ is entirely above the floor,

$$\mathcal{L}_{\text{g}} = \frac{1}{|\mathcal{K}_{\mathbf{O}_2}|} \sum_{\boldsymbol{\mu}_i \in \mathcal{K}_{\mathbf{O}_2}} \left(\mu_i^y - c^{\text{floor}}\right) \text{ if } \mu_i^y > c^{\text{floor}} \, \forall \, \boldsymbol{\mu}_i \in \mathcal{K}_{\mathbf{O}_2}. \tag{16}$$

This is shown in Fig. H(a), left side. However, when some Gaussians are below the floor, it should provide a larger relative gradient to pull those Gaussians back above the floor. That is, in such a setting, the loss is given by,

$$\mathcal{L}_{\text{g}} = \frac{1}{K_{\text{comb}}} \left[\frac{1}{|\mathcal{K}_{\mathbf{O}_2}^+|} \sum_{\boldsymbol{\mu}_i \in \mathcal{K}_{\mathbf{O}_2}^+} \left(\mu_i^y - c^{\text{floor}}\right) \right.$$
$$\left. + \frac{1}{|\mathcal{K}_{\mathbf{O}_2}^-|} \sum_{\boldsymbol{\mu}_i \in \mathcal{K}_{\mathbf{O}_2}^-} \left(c^{\text{floor}} - \mu_i^y\right), \tag{17} \right.$$

where $K_{\text{comb}} >> 1$ is a combination factor between the two terms, and $\mathcal{K}_{\mathbf{O}_2}^+$ and $\mathcal{K}_{\mathbf{O}_2}^-$ are defined such that,

$$\mathcal{K}_{\mathbf{O}_2}^+ = \{\boldsymbol{\mu}_i \mid \boldsymbol{\mu}_i \in \mathcal{K}_{\mathbf{O}_2}, \mu_i^y > c^{\text{floor}}\}$$
$$\mathcal{K}_{\mathbf{O}_2}^- = \{\boldsymbol{\mu}_i \mid \boldsymbol{\mu}_i \in \mathcal{K}_{\mathbf{O}_2}, \mu_i^y < c^{\text{floor}}\}. \tag{18}$$

This is shown in Fig. H(a), right side. In practice, we use $K_{\text{comb}} = 2000$.

Contact loss $\mathcal{L}_c$: $\mathbf{O}_1$ and $\mathbf{O}_2$ are non-rigid Gaussian-represented objects. Without explicit constraints, they can intersect with each other, leading to unrealistic-looking scenes. The contact loss addresses this. Enforcing such a loss is nuanced. Gaussian-represented objects do not have a defined surface, along which contact occurs. Our key observation pertains to what we refer to as **the contact angle** $\theta_c^{\boldsymbol{\mu}_j}$ for a Gaussian $\boldsymbol{\mu}_j \in \mathcal{K}_{\mathbf{O}_2}$. For a Gaussian with mean $\boldsymbol{\mu}_j$ such that $j \in \mathbf{O}_2$, let $\boldsymbol{\mu}_i$ be the mean of the closest Gaussian in $\mathbf{O}_1$. Then, $\theta_c^{\boldsymbol{\mu}_j}$ is the angle between the vectors $\boldsymbol{\mu}_i - \mathbf{q}_2$ and $\boldsymbol{\mu}_i - \boldsymbol{\mu}_j$. When $\boldsymbol{\mu}_j$ is not intersecting $\mathbf{O}_1$, $\theta_c^{\boldsymbol{\mu}_j} < \pi/2$ (Fig. H(b), left side), and when $\boldsymbol{\mu}_j$ is intersecting $\mathbf{O}_1$, $\theta_c^{\boldsymbol{\mu}_j} > \pi/2$ (Fig. H(b), right side). To avoid intersection, we aim to enforce that the contact angle is acute for intersecting Gaussians. Then,

$$\mathcal{L}_c = \frac{1}{\left|\mathcal{K}_{\mathbf{O}_2}^{<\frac{\pi}{2}}\right|} \sum_{\boldsymbol{\mu}_j \in \mathcal{K}_{\mathbf{O}_2}^{<\frac{\pi}{2}}} - \cos \theta_c^{\boldsymbol{\mu}_j}, \tag{19}$$

where $\mathcal{K}_{\mathbf{O}_2}^{<\frac{\pi}{2}} = \{\boldsymbol{\mu}_j \mid \boldsymbol{\mu}_j \in \mathcal{K}_{\mathbf{O}_2}, \theta_c^{\boldsymbol{\mu}_j} > \pi/2\}$.
The overall physics loss is therefore given by,

$$\mathcal{L}_{\text{Phys}} = \lambda_g \mathcal{L}_g + \lambda_c \mathcal{L}_c, \tag{20}$$

where λ_g and λ_c are appropriate regularization parameters. In practice we use $\lambda_g = 10,000$. For λ_c, we find that λ_c must balance the current gravity loss, to prevent the objects from remaining intersected. Therefore, in the case of object intersection, we set $\lambda_c = \mathcal{L}_g * 30,000$. We find that 200 steps of the physics-guided optimization is sufficient to arrive at feasible compositions. These hyperparameters were found to work for our tested examples, but can be finetuned depending on specific use cases as well, based on the user.

Stabilizing impulse: As mentioned in the main paper, in additiona to the above physics-based losses, we include a "stabilizing impulse" as part of our composition. The necessity for this constraint arises out of limitations of 2D diffusion guidance for 3D composition. Specifically, since we use elevated camera perspectives, some projective ambiguities might creep through. Therefore, for prompts such as "a hamburger on a plate", the hamburger might find itself not perfectly aligned with the plate and hence might overhang. Then, once the physical constraints are activated, this

Parameter	Value
Iterations	3000
Learning rate	
μ (Start, End)[exp. decay]	(2e-3,1e-4)
c	0.01
σ	0.01
Σ	0.002
Densification	
(Start, End)	(0, 2000)
Interval	250
Gradient Threshold	0.5
Max Spherical Harmonic	0
Opacity Reset Interval	$\emptyset$

Table B. **3D Gaussian Splatting hyperparameters for object generation.**

might lead to the hamburger falling off the plate, for example. Therefore, we incorporate the constraint as follows.

If the top-view cross section areas of $\mathbf{O}_1$ and $\mathbf{O}_2$ overlap in area between 40% and 95% of the cross section area of $\mathbf{O}_2$, and if $\mathbf{O}_1$ and $\mathbf{O}_2$ have come into contact, $\mathbf{O}_2$ receives a small impulse toward the central axis of $\mathbf{O}_1$, and upwards. This impulse is set to have a translation distance of 0.3 and an angle with the horizontal of 60 degrees. We limit this impulse to only act 5 times for a composition. This accounts for strong diffusion guidance against the impulse.

D. Object Generation Details

The primary objective function is the SDS loss that we minimize for *iterations* $= 3,000$, *batch size* $= 4$, *prompt* $= y$ and $CFG = 100$. We scale the loss by 0.5, and linearly reduce the sampled timestep, t, interval from $[2, 980]$ to $[2, 500]$ for 2,000 steps for faster convergence. We rescale the SDS loss according to Lin et al. [20] by a factor of 0.7 to reduce overexposure. We initialize the Gaussians using PointE's text-to-3D model [10, 27]. Given a text prompt, it returns 4,096 points that are converted to 3D isotropic Gaussians with random color features, $scale = 0.02$, and $opacity = 0.8$. We sample images of the 3D object from random views: azimuth from $[0°, 360°]$, elevation from $[-30°, 80°]$, and FOV from $[30°, 55°]$. The prompt y is appended with text to describe the direction of the sampled object view according to Poole et al. [30] to reduce Janus affects. The 3D Gaussians are updated according to the parameters defined in Tab. B.

D.1. Alpha Hull

The Alpha hull [6] loss, $\beta \mathcal{L}_{\text{KNN}}(\theta, \mathcal{A})$, brings points to the surface to distribute Gaussians more uniformly and densely. We set $\beta = 5$ and $K = 5$. Fig. I illustrates the set of points, $\mathcal{A}$, that are picked for an object. Fig. K shows a
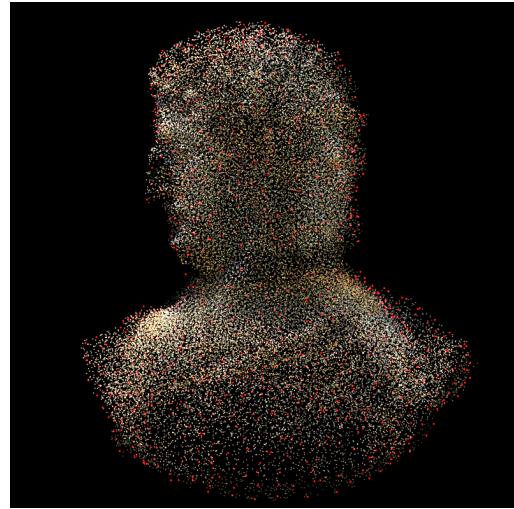

Figure I. **We visualize the set of points belonging to the Alpha hull, $\mathcal{A}$.** For visibility, Gaussians have been scaled down to 1%. The points belonging to $\mathcal{A}$ are marked in red, while the rest of the points keep their original color. The Alpha hull loss brings all Gaussians closer to the set $\mathcal{A}$. The original 3D object is in Fig. K in the right-most column.

comparison between optimization with and without the Alpha hull loss. Without it, the surface of objects can appear blotchy and individual Gaussians are more visible.

Figure J. **Visualization of object distillation for a "pear".**

$$\mathcal{L}_{SDS} \qquad\qquad\qquad \mathcal{L}_{SDS} + \beta\mathcal{L}_{\text{KNN}}(\theta, \mathcal{A})$$

Figure K. **Comparison between object generation without (left) and with (right) the Alpha hull proximity loss.** The object was generated with the prompt "a Greek bust".

$$\mathcal{L}_{SDS} + \beta\mathcal{L}_{\text{KNN}}(\theta, \mathcal{A}) \qquad\qquad \mathcal{L}_{SDS} + \beta\mathcal{L}_{\text{KNN}}(\theta, \mathcal{A}) + \gamma\mathcal{L}_{\text{KNN}}(\theta, \mathcal{E})$$

Figure L. **Comparison between object generation without (left) and with (right) the PointE initialization proximity loss.** The object was generated with the prompt "a wooden table".

D.2. PointE KNN

The optional PointE KNN provides crucial geometrical guidance to enable generation of certain objects with geometries that have 3D ambiguities under image projection. The PointE KNN constrains Gaussians to be near the set of initialization points provided, $\mathcal{E}$. Then, the KNN loss is $\gamma\mathcal{L}_{\text{KNN}}(\theta, \mathcal{E})$ (defined in Eq. (11)), where $\gamma = 20$ and $K = 1$. We find this loss to be instrumental in two cases: preserving flat surfaces and maintaining convexity. While the loss is not strictly required for a single object to appear accurate under 2D projection, its absence can have consequences for 3D composition. For example, we find that objects that have concavities, such as a basket, open box, or cup, will have the concavity filled or partially filled during optimization without the geometrical regularization. This results in failed compositions where another object is supposed to fill the concavity. Similarly, without the PointE KNN, a flat surface will begin to become curved and convex. This is apparent in Fig. L. In column 3, we can see that the top view appears reasonable, however, the side view shows the convexity of the table's surface. Whereas with the PointE KNN, in column 4, such effects are reduced. For composition, flat surfaces are necessary to preserve phys-

ically realistic scenes, especially from side views. On the other hand, PointE KNN can be detrimental if applied to all objects. Strict geometrical regularization to the initialized shape can cause object's to not develop new features. For this reason, the PointE KNN is optional and should be supplied as input by a user.

D.3. Memory Distillation

The geometry of the generated objects is not well-defined during training, unlike in the novel-view synthesis setting with well-defined supervision. Object generation with the 3D Gaussian splatting framework requires frequent duplication of Gaussians to enable high-frequency details and expressability of the model. Therefore, due to overzealous duplication, redundant Gaussians are unavoidable. Moreover, for large-scale scene generation with numerous objects, we need to avoid high memory usage. This problem can be addressed by distilling the dense representation of a generated object. This is done by retraining a new 3D Gaussian model on views from the original dense object. In fact, this amounts to reducing the problem to classic novel-view synthesis. As a bonus, this may be even more stable due to the availability of an infinite number of views of the object from various camera positions and radii. Quantitative results are

shown in Tab. 1 and a qualitative result is shown in Fig. J where the scale of the Gaussians has been reduced to show the effect of distillation.

E. User Input Details

As illustrated in Fig. 3, the composition-level text prompt can be explicitly decomposed into the object-level prompts and interaction prompts. The user then has to provide two forms of input: object-level and interaction text prompts. Formally, for each object $\mathbf{O}_i$, there must be a corresponding text prompt describing the object $\mathbf{y}_i$. Similarly, for all pairs of objects with interactions $\mathbf{P}_{i,j}$, there must also be a corresponding textual description of the interaction $\mathbf{y}_{i,j}$. The interaction text prompt must reference the objects participating in the interaction, $(\mathbf{O}_i, \mathbf{O}_j)$, but does not necessarily need to include the object's original text attributes such as color or style. For example, given the composition-level text prompt, "a roasted chicken on a plain white plate laying on a wooden table", it can be decomposed as follows:

1. Objects:
   (a) $\mathbf{O}_1$ = *photo of a roasted chicken*
   (b) $\mathbf{O}_2$ = *photo of a plain white plate*
   (c) $\mathbf{O}_3$ = *photo of a wooden table*
2. Interactions:
   (a) $\mathbf{P}_{1,2}$ = *roasted chicken on a plate*
   (b) $\mathbf{P}_{2,3}$ = *a plate on a table*

which can then be supplied as input to our model.

F. Additional Results

We include additional results for various compositions of objects in Fig. M and Fig. N on the following pages.

Elevation   45°   0°   90°

A lamp on a
night stand.

A TV on a
table console.

Bacon cooking on
a frying pan.

Christmas tree
on a rug.

Figure M. **Additional Composition Results 1.**

Elevation	45°	0°	90°

Roasted chicken on a plain plate on a wooden table.

Blue cube on top of a red cube.

Camp fire next to a tent.

Frying pan on a stove.

Figure N. **Additional Composition Results 2.**