

A APPENDIX / SUPPLEMENTAL MATERIAL

In this supplementary material, we first clarify the notations used in this paper and then revisit the proposed COMPGS in Algorithms 1. The training details of COMPGS will also be provided. Besides, we provide more numerical and visual evaluations to further validate the effectiveness of our model. We have provided a **demo video** in the attachment to display more visual comparisons between COMPGS and other methods. We will make code public.

A.1 NOTATIONS

We compile a comprehensive list of all the notations utilized in this paper, as shown in Table 4.

Table 4: Notations.

Notation	Description
L	Total number of entities
V	Complex prompt (e.g., 'an owl perches on a branch near a pinecone')
I	Composed image generated by the 2D diffusion model
v_l	Entity-level prompt for entity l , ($l \in L$)
I_l	Segmented image containing entity l , ($l \in L$)
m_l	Rough triangle mesh of the 3D entity l , ($l \in L$)
θ_l	3D Gaussians for the entity l , ($l \in L$)
θ	Composed 3D Gaussians l
N	Number of points indexed from each mesh
μ_i^l	Center positions of each vertex of mesh m_l in \mathbb{R}^3
c_i^l	Texture colors queried from each vertex of mesh m_l in \mathbb{R}^3
bbox_l	3D bounding box for entity l , used for optimization
bbox_{std}	Standardized volumetric space for scaling
μ	Center positions of each vertex in the original 3D space
$\hat{\mu}$	Transformed center positions of entity Gaussian after scaling
β	Shift parameters for the center positions of the bounding box
λ	Scale parameters for standardizing the volumetric space
x	Rendered image from 3D Gaussians
$g(\cdot)$	Gaussian Splatting rendering function
β	the shift parameters for volume-adaptive optimization
λ	the scale parameters for volume-adaptive optimization
$\text{Mean}(\cdot)$	the operator computing the center coordinates of the given bounding box
$\hat{\theta}$	New Gaussians initialized from the edited 2D image

A.2 ALGORITHM

We provide pseudocode in Algorithm 1. Two core designs, including 3D Gaussian initialization with 2D compositionality and dynamic SDS optimization, are detailed.

A.3 ADDITIONAL TRAINING DETAILS

COMPGS is implemented in ThreeStudio Guo et al. (2023). We use DALL-E 3 Betker et al. (2023), LangSAM Medeiros (2024) and TripoSR Tochilkin et al. (2024) to implement the text-to-image, text-guided segmentation, and image-to-mesh, respectively. For entity-level optimization, we adopt MVDream Shi et al. (2023) as the 3D diffusion prior; while for composition-level optimization, we employ *stabilityai/stablediffusion-2-1-base* Rombach et al. (2022b) as the 2D diffusion prior. We set all the diffusion guidance as 50. For all Gaussian parameters, we linearly decreased the learning rate for position μ from 10^{-3} to 10^{-5} , for scale from 10^{-2} to 10^{-3} , and for color c from 10^{-2} to 10^{-3} , respectively. Besides, we fixed the learning rate for opacity a to be 0.05, and for rotation to be 0.001. Additionally, we use a consistent batch size of 4 for both training and test, and a rendered resolution fixed at 1024×1024 . Camera settings during training are set with distances ranging from 0.8 to 1.0 relative units, a field of view between 15 and 60 degrees, and elevation ranging up to 30 degrees. Additionally, there are no perturbations applied to camera position, center, or orientation, maintaining a controlled imaging environment. For test, we set the resolution of

Algorithm 1 COMPGS: 3D Gaussian Initialization and Dynamic SDS Optimization $V, \{v_l\} (l \in L)$:

Input prompt and entity-level prompts.

 $\{m_l\} (l \in L)$: Entity-level meshes. $\theta, \{\theta_l\} (l \in L)$: Composition-level Gaussian parameters and entity-level Gaussian parameters. bbox_{std} : Standardized volumetric space. L : The number of entities. N : The number of Gaussian parameters.

T2I: Text-to-Image models.

TGS: Text-guided segmentation models.

I2M: Image-to-Mesh models.

 $\text{Zoom}^\uparrow, \text{Zoom}^\downarrow$: Zoom-in and Zoom-back operators in Eq. 4. η : Learning rate. T : Total training iterations.*Stage 1: Initializing 3D Gaussians with 2D Compositionality.*

$I = \text{T2I}(V)$ \triangleright Generate well-composed Image from the given prompt
 $\{v_l\} = \text{LLM}(V)$ \triangleright Obtain entity-level prompts via LLM
 $\{m_l\} = \text{I2M}(\text{TGS}(\{v_l\}, I))$ \triangleright Obtain entity-level meshes
 $\mu_i (i \in N), c_i (i \in N) \leftarrow m_l (l \in L)$ \triangleright Positions and colors of the 3D Gaussians.
 $D \leftarrow \mu_i (i \in N)$ \triangleright Distance between the nearest two positions.
 $\Sigma_i (i \in N), \alpha_i (i \in N) \leftarrow D, 0.1$ \triangleright Covariance and opacity of the 3D Gaussians.
 $\text{bbox}_l (l \in L) \leftarrow \mu_i (i \in N)$ \triangleright Boundary of bounding box

*Stage 2: Dynamic SDS Optimization.***for** $t = 1$ to T **do** $l \leftarrow \text{randint}(1, L)$ \triangleright Randomly select an integer l from the range 1 to L **if** $i = 0$ **then**

$\nabla_\theta \mathcal{L}_{\text{SDS}}^{2d}(\phi, \mathbf{x} = g(\theta)) \triangleq \mathbb{E}_{t, \epsilon} [w(t) (\hat{\epsilon}_\phi(\mathbf{z}_t, V, t) - \epsilon) \frac{\partial \mathbf{x}}{\partial \theta}]$
 \triangleright Obtain the gradients via SDS loss with 2D priors
 $\nabla_\theta \mathcal{L}_{\text{SDS}}^{3d}(\phi, \mathbf{x} = g(\theta)) \triangleq \mathbb{E}_{t, \epsilon} [w(t) (\hat{\epsilon}_\phi(\mathbf{z}_t, v, t) - \epsilon) \frac{\partial \mathbf{x}}{\partial \theta}]$
 \triangleright Obtain the gradients via SDS loss with 3D priors
 $\theta \leftarrow \theta - \eta(\nabla_\theta \mathcal{L}_{\text{SDS}}^{2d} + \nabla_\theta \mathcal{L}_{\text{SDS}}^{3d})$
 \triangleright Update the compositional Gaussian parameters via back-propagation

else

$\hat{\theta}_l \leftarrow \text{Zoom}^\uparrow(\theta_l, \text{bbox}_l, \text{bbox}_{std})$
 \triangleright Dynamically zoom-in Gaussian parameters from bbox_l to a standardized space bbox_{std}
 $\nabla_{\hat{\theta}_l} \mathcal{L}_{\text{SDS}}^{3d}(\phi, \mathbf{x} = g(\hat{\theta}_l)) \triangleq \mathbb{E}_{t, \epsilon} [w(t) (\hat{\epsilon}_\phi(\mathbf{z}_t, v_l, t) - \epsilon) \frac{\partial \mathbf{x}}{\partial \hat{\theta}_l}]$
 \triangleright Obtain the gradients via SDS loss with 3D priors
 $\hat{\theta}_l \leftarrow \hat{\theta}_l - \eta \nabla_{\hat{\theta}_l} \mathcal{L}_{\text{SDS}}^{3d}$
 \triangleright Update the compositional Gaussian parameters via back-propagation
 $\theta_l \leftarrow \text{Zoom}^\downarrow(\hat{\theta}_l, \text{bbox}_l, \text{bbox}_{std})$
 \triangleright Dynamically zoom-back Gaussian parameters from the standardized space bbox_{std} to bbox_l

end for

A.4 EXTENDED EXPERIMENTS ON QUALITATIVE COMPARISONS

19

Table 5: **Quantitative comparisons with baselines on T³Bench** [He et al. \(2023\)](#) (all three tracks). COMPGS is compared with feed-forward models, optimization-based models, and models specifically designed for compositional generation.

Method	Single Object			Single Object with Surroundings			Multiple Objects		
	Quality	Alignment	Average	Quality	Alignment	Average	Quality	Alignment	Average
LRM Hong et al. (2023)	29.4	38.2	33.8	20.3	35.1	27.7	15.2	25.5	20.4
TripoSR Tochilkin et al. (2024)	34.3	38.9	36.6	21.8	37.2	29.5	16.7	28.6	22.7
DreamFusion Poole et al. (2022)	24.9	24.0	24.4	19.3	29.8	24.6	17.3	14.8	16.1
SJC Wang et al. (2023a)	26.3	23.0	24.7	17.3	22.3	19.8	17.7	5.8	11.7
LatentNeRF Metzner et al. (2023)	34.2	32.0	33.1	23.7	37.5	30.6	21.7	19.5	20.6
Fantasia3D Chen et al. (2023b)	29.2	23.5	26.4	21.9	32.0	27.0	22.7	14.3	18.5
ProlificDreamer Wang et al. (2024)	51.1	47.8	49.4	42.5	47.0	44.8	45.7	23.8	35.8
Magic3D Lin et al. (2023)	38.7	35.3	37.0	29.8	41.0	35.4	26.6	24.8	25.7
Set-the-Scene Cohen-Bar et al. (2023)	32.9	31.9	32.4	30.2	45.8	35.5	20.8	29.9	25.4
VP3D Chen et al. (2024c)	54.8	52.2	53.5	45.4	50.8	48.1	49.1	31.5	40.3
COMPGS	55.1	52.5	53.8	43.2	46.8	45.0	54.2	37.9	46.1

successfully captures both the key entities described in the prompt and generates reasonable spatial relationships and interactions between the two objects. This phenomenon can also be observed in other cases, such as the key and lock in the third row, and the fisherman in the sea in the fifth row, and so on. Besides the issue of 3D consistency, we found that COMPGS performs better in texture alignment. For example, in the second-to-last row, other methods failed to display the combination of chessboard, king, and queen. Specifically, VP3D did not recognize the king and queen as chess pieces. In contrast, COMPGS generates these entity details more accurately. Overall, the comparisons in both visual quality and textural alignment with previous methods demonstrate the effectiveness of the proposed COMPGS.

Qualitative Model Comparisons on Single-object Generation Though COMPGS is specifically designed for compositional generation, it can naturally handle single-object generation as well. We present the qualitative comparisons between COMPGS and previous works in Fig. 8. It is observed that COMPGS performs better in maintaining multi-view consistency and generating fine-grained details of the object. For example, in the last row of Fig. 8, COMPGS is capable of generating a 3D consistent candle holder, including detailed copper textures. In contrast, other methods either fail to produce the corresponding shape [Chen et al. \(2023b\)](#), only generate rough outlines without detailed textures [Poole et al. \(2022\)](#); [Lin et al. \(2023\)](#); [Metzner et al. \(2023\)](#); [Wang et al. \(2023a\)](#), or produce 3D patterns with discontinuities [Wang et al. \(2024\)](#); [Chen et al. \(2024c\)](#).

Qualitative Model Comparisons with Scene-generation Methods We also compare COMPGS with closed-source models [Zhou et al. \(2024\)](#); [Cohen-Bar et al. \(2023\)](#) that generate 3D scenes. Figures were selected from [Zhou et al. \(2024\)](#) and are presented in Fig. 9. The results indicate that COMPGS excels in generating high-fidelity texture details and complex interactions. In the second row of Fig. 9, COMPGS produces more detailed textures for table legs and rabbit fur. Regarding interaction generation, Set-the-Scene [Cohen-Bar et al. \(2023\)](#) fails to create complex spatial relationships, as shown with the dog and the Great Pyramid in the first row. Although GALA3D can generate reasonable spatial relationships, it fails to incorporate mutual interactions between objects. This is because it performs compositional generation by optimizing the layout of each object individually, neglecting other inter-interactions such as the rabbit’s mouth on the cake and the dog’s paw on the plate. In contrast, COMPGS generates higher-fidelity textures (e.g., the table body, rabbit fur) and more realistic interactions among objects (e.g., the dog’s paw hanging off the plate rather than just resting on top).

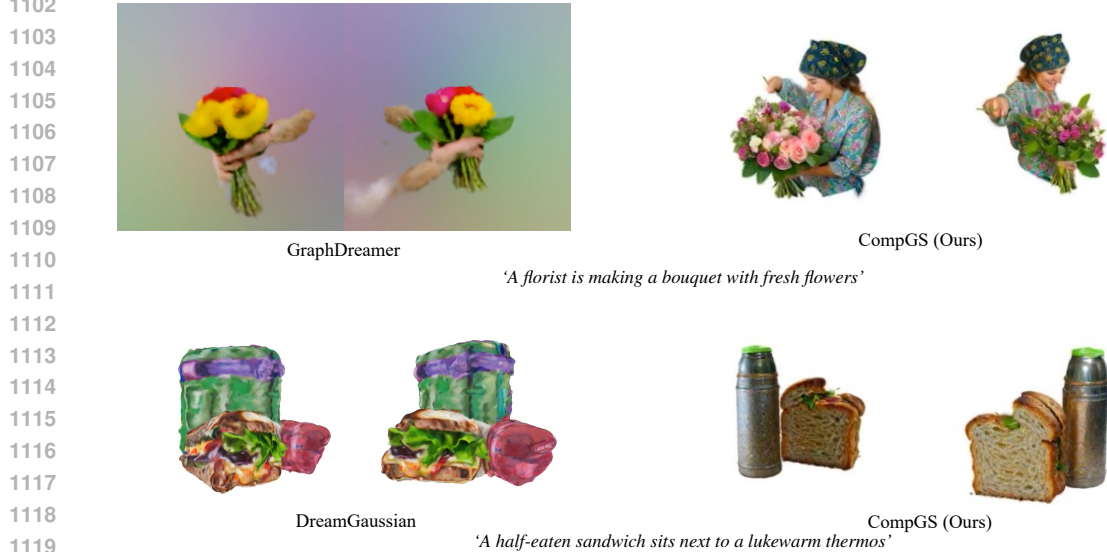
Qualitative Model Comparisons with Other Compositional Generation Methods In the main paper, we have compared COMPGS with both open-sourced compositional 3D generation baselines (Set-the-scene and VP3D) in Table 1, and close-sourced baselines (GALA3D) in Figure 9. Results show that the 3D assets generated by COMPGS are not only high-quality in appearance, but also align with the given prompts more strictly. We have included qualitative comparisons with both GraphDreamer [Gao et al. \(2024\)](#) and DreamGaussian [Tang et al. \(2023\)](#) in Fig. 10. Results show that COMPGS demonstrates superior performance on both generation quality and text-3d alignment.

A.5 QUANTITATIVE MODEL COMPARISONS

Tab. 5 presents the complete quantitative comparisons on all three tracks of T³Bench. The results indicate that COMPGS achieved state-of-the-art performance in compositional generation and slightly outperformed competitors in the single object track. For instance, in the multiple object track, our model surpassed the second-best work [Chen et al. \(2024c\)](#) by 5.1 in quality and 6.4 in texture



1096 **Figure 9: Qualitative Comparisons Between COMPGS and 3D Scene Generation Methods.** We
 1097 selected the figures from [Zhou et al. \(2024\)](#) for these comparisons due to the unavailability of the
 1098 code. COMPGS performs better in generating object textures and complex interactions.



A.6 EXAMPLES IN USER STUDY

We provide examples of images and scenes used in our user study. In particular, we present concatenated rendering videos and ask participants to rank the eight methods shown in the video based on the overall quality of the 3D objects and the alignment between the text and the 3D models. We average the rank number as its ranking score for comparisons in Tab. 1.



Figure 11: Examples used in our user study.

A.7 ROBUSTNESS

We empirically found that COMPGS demonstrates the ability to address certain deficits caused by off-the-shelf model priors (e.g., T2I and segmentation priors). Here are some illustrative examples: (1) If certain parts of the target objects are not correctly segmented, COMPGS can complete the unsegmented part with correct 3D information. This is demonstrated in Fig. ??12(left), where the swing has not been segmented but has been generated by COMPGS correctly. This is facilitated through the Entity-level Optimization procedure proposed in the DO strategy. (2) If the T2I models fail to generate proper intra-object interactions, COMPGS can correct the multi-object interactions. This is shown in Fig. ??12(right), where the spatial relationships in the given image are incorrect and then corrected in the text-to-3D process. This is achieved by the Composition-level Optimization in the proposed DO strategy.

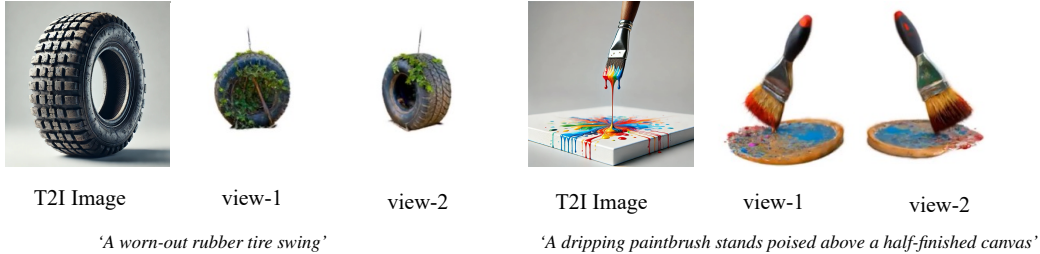


Figure 12: CompGS demonstrates the ability to address certain deficits caused by off-the-shelf priors.

A.8 FAILURE CASES

As discussed in Sec. 5, COMPGS exhibits limitations in generating backgrounds, such as ground and sky. This is likely due to the current text-guided segmentation model’s inability to effectively segment these abstract concepts. When the background is not well-segmented, we lose the corresponding 2D compositionality needed for initializing 3D Gaussians. This leads to two failure cases: (1) the absence of background in the compositional 3D scenes, as seen with the missing grass in the second column of Fig. 13, or (2) background generation of poor visual quality, such as the vague and unclear depiction of grass in the first column of Fig. 13. It’s crucial to note that such limitations, whilst exist, are not the focus of this work. These shortcomings can be overcome by enhancing the capabilities of off-the-shelf models, effectively mitigating the manifested issues.

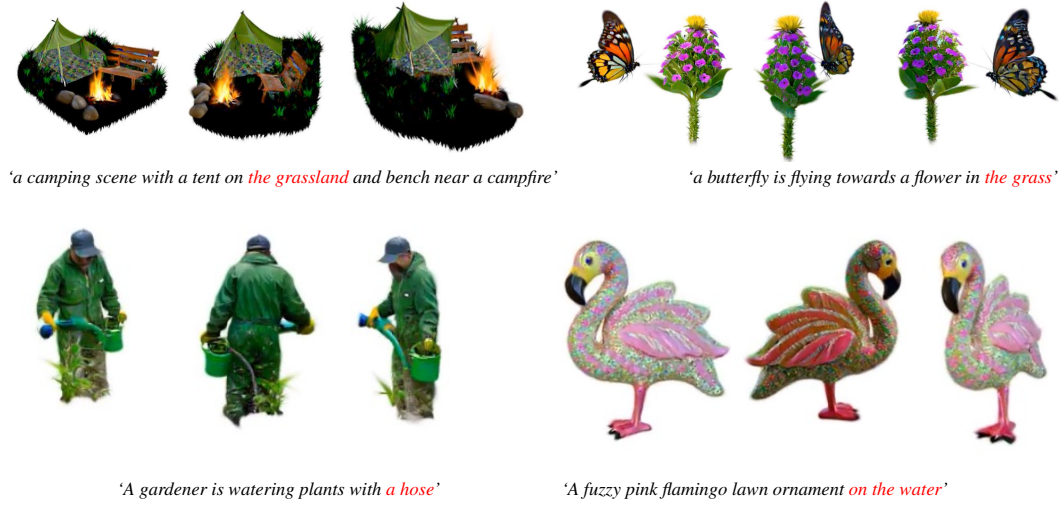


Figure 13: **Failure Cases of COMPGS in background generation** When text-guided segmentation mode fails to segment the backgrounds, COMPGS may generate background with poor visual quality or fails to generate background.

A.9 3D EDITING EXAMPLES

COMPGS offers a user-friendly approach to progressively conduct 3D editing for compositional 3D generation. More visual examples are presented in Fig. 14. For instance, given a compositional prompt such as 'A puppy lying on the iron plate on the top of the Great Pyramid, with a pharaoh nearby', we divide the generation process into four stages. Initially, we generate 'the Great Pyramid' on the left, then progressively add 'the plate', 'the puppy', and 'the pharaoh' to complete the 3D scene. Notably, both the interactions and texture details can be well-produced during the editing pipeline of COMPGS.

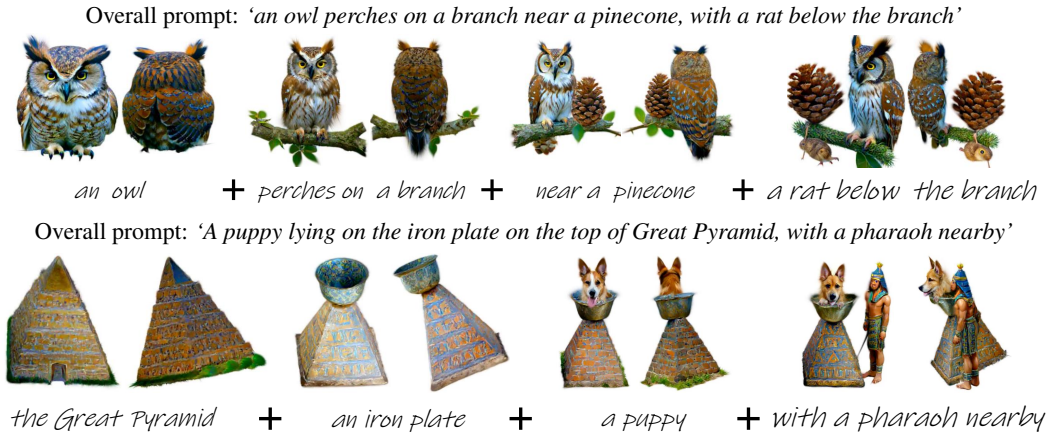


Figure 14: **More examples of 3D Editing.** COMPGS provides a user-friendly way to progressively edit on 3D scenes for compositional generation.