Towards Text-guided 3D Scene Composition Supplementary Material

Qihang Zhang[1,2*] Chaoyang Wang[2] Aliaksandr Siarohin[2] Peiye Zhuang[2] Yinghao Xu[3]
Ceyuan Yang[1] Dahua Lin[1] Bolei Zhou[4] Sergey Tulyakov[2] Hsin-Ying Lee[2]
[1]CUHK [2]Snap Inc. [3]Stanford [4]UCLA

This appendix is organized as follows. Secs. 1 and 2 present the implementation details of our proposed `SceneWiz3D` and baselines respectively. Sec. 3 lists all the prompts used for evaluation. Sec. 4 provides a more comprehensive comparison by computing metrics using different backbones. Sec. 5 justifies the usage of FID over disparity map to assess the geometry of synthesized scenes. We also show qualitative comparison and a collection of diverse synthesized scenes on the anonymous website: https://zqh0253.github.io/SceneWiz3D/.

1. Implementation Details of `SceneWiz3D`

Details about prompting the generative process. Taking advantage of our disentangled representation, we implement a strategy that, during a specific ratio of iterations (we use 0.3 in all our experiments), we discard all foreground objects and exclusively render the background. For the remaining iterations, we render the entire scene. Consequently, the prompts for these two cases are slightly different. For instance, we utilize the prompt "a single boy visiting a spacious aquarium" for the complete scene, while "a spacious aquarium" for the background. Through this approach, we have observed significant mitigation of the Janus problem.

Perspective guidance. We use VSD as our perspective view guidance. We inherit the same camera sampling strategy and annealed time schedule for score distillation as in ProlificDreamer [9].

Panoramic RGBD guidance. Different from perspective camera that is sampled on a sphere and looks at the middle of the scene, our panoramic camera is placed at the center of the scene with a small random offset. The magnitude of this offset is limited to a maximum of ten percent of the scene radius. Since the panoramic guidance does not consider rendering Object of Interests (OOIs) in the panoramic view, it lacks awareness of the presence of OOIs. To avoid conflicts between the panoramic guidance and OOIs, we exclude the panoramic guidance from the initial 5000 iterations. After the first 5000 iterations, we introduce the panoramic guidance based on the rough scene layout obtained from the perspective guidance. To ensure a smooth transition, we gradually anneal the maximum time step from 0.5 at 5000 iterations to 0.3 at 20000 iterations. We render the panoramic image in 256×512 resolution. Furthermore, we also exclude the panoramic guidance during the last 5000 iterations. We have observed that incorporating it during this stage can slightly compromise the visual quality of the scene.

Particle Swarm Optimization. We observed that optimizing scene configuration with SDS loss tends to result in being trapped in local minima. Therefore, we propose the use of Particle Swarm Optimization (PSO) as an alternative method for updating scene configuration. PSO maintains a swarm of particles and iteratively update them as:

$$\begin{aligned} \mathbf{v}_i(n+1) = \; & k \cdot \mathbf{v}_i(n) \\ & + c_1 \cdot r_1 \cdot (\mathbf{pbest}_i - \mathbf{a}_i(n)) \\ & + c_2 \cdot r_2 \cdot (\mathbf{gbest} - \mathbf{a}_i(n)), \\ \mathbf{a}_i(n+1) = \; & \mathbf{a}_i(n) + \mathbf{v}_i(n+1), \end{aligned} \tag{A1}$$

where $\mathbf{pbest}_i$ is the best position found by the i-th particle, $\mathbf{gbest}$ is the best position found by all the particles in the swarm, k, c_1, c_2 are hyper-parameters, and r_1, r_2 are random numbers controlling the intensity of exploration and exploitation. This process is described in Algorithm 1. In our experiments, a swarm consisting of 30 particles is maintained, and updates are performed over 50 iterations for each PSO phase. We set the hyper parameters as $k = 0.8$ and $c_1 = c_2 = 0.1$.

Depth regularizer $\mathcal{L}_{\text{dep}}$. We use the official *dpt-beit-large-512* version of MiDaS[1] to estimate the target depth for calculating the depth regularizer term: $\min_{s,b \in \mathbb{R}} \|sI_d + b - \hat{I}_d\|_2^2$. As MiDaS's result is up-to-scale, we therefore use a scale s and bias b term to align the rendered disparity map I_d to the

*Work done during internships at Snap Inc.

[1]https://github.com/isl-org/MiDaS

Algorithm 1 PSO for scene config update

Require: I ▷ Number of particles
Require: N ▷ Number of time steps
Require: $f(\cdot)$ ▷ Scoring (CLIP similarity) function
Require: k, c_1, c_2 ▷ Hyper parameters
for $i = 0$ **to** $I - 1$ **do**
  $\mathbf{a}_i[0] \leftarrow \textsc{Random}$ ▷ Random initial solution
  $\mathbf{v}_i[0] \leftarrow \textsc{Random}$ ▷ Random initial velocity
  $\mathbf{pbest}[i] \leftarrow \mathbf{a}_i[0]$ ▷ Initial local best
end for
for $i = 0$ **to** $I - 1$ **do**
  if $f(\mathbf{pbest}[i]) > f(\mathbf{gbest})$ **then**
    $\mathbf{gbest} \leftarrow \mathbf{pbest}_i$ ▷ Initial global best
  end if
end for

for $n = 0$ **to** $N - 1$ **do**
  for $i = 0$ **to** $I - 1$ **do**
    $r_1 \leftarrow \textsc{Rand}(0, 1)$ ▷ Exploration intensity
    $r_2 \leftarrow \textsc{Rand}(0, 1)$ ▷ Exploitation intensity
    $\mathbf{v}_i[n+1] \leftarrow k \times \mathbf{v}_i[n]$ ▷ Update velocity
      $+c_1 \times r_1 \times (\mathbf{pbest}[i] - \mathbf{a}_i[n])$
      $+c_2 \times r_2 \times (\mathbf{gbest} - \mathbf{a}_i[n])$
    $\mathbf{a}_i[n+1] \leftarrow \mathbf{a}_i[n] + \mathbf{v}_i[n+1]$ ▷ Update solution
    if $f(\mathbf{a}_i[n+1]) > f(\mathbf{pbest}[i])$ **then**
      $\mathbf{pbest}[i] \leftarrow \mathbf{a}_i[n+1]$ ▷ Update local best
    end if
  end for
  for $i = 0$ **to** $I - 1$ **do**
    if $f(\mathbf{pbest}[i]) > f(\mathbf{gbest})$ **then**
      $\mathbf{gbest} \leftarrow \mathbf{pbest}_i$ ▷ Update global best
    end if
  end for
end for

predicted disparity map $\hat{I}_d$. The optimal scale s and bias b term has closed-form solution:

$$s = \frac{(\mathbf{1}^\top \mathbf{1})(I_d^\top \hat{I}_d) - (I_d^\top \mathbf{1})(\hat{I}_d^\top \mathbf{1})}{(\mathbf{1}^\top \mathbf{1})(I_d^\top I_d) - (I_d^\top \mathbf{1})^2},$$
$$b = \frac{\hat{I}_d^\top \mathbf{1} - s I_d^\top \mathbf{1}}{(\mathbf{1}^\top \mathbf{1})}. \tag{A2}$$

After computing the optimal scale and bias, we then calculate the Mean Square Error (MSE) between aligned rendered disparity map $sI_d + b$ and the predicted disparity map $\hat{I}_d$ as the loss term.

Coefficients. λ_{pers}, λ_{pano}, and λ_{dep} in **??** are set to $1, 10^{-1}, 10^4$ for all experiments.

2. Implementation Details of Baselines

ProlificDreamer [9]. We adopt the implementation from threestudio[2] which achieves a similar visual quality to the results in the original paper. We inherit the camera sampling scheme and density initialization as proposed in the original paper. We only render 64×64 resolution images for the first 5000 iterations, and then render in 512×512 images for another 20000 iterations.

DreamFusion [6]. We implement DreamFusion based on the ProlificDreamer's implementation specified above. We preserve all the configs, except for the modification of the guidance term from VSD to SDS.

Text2room [3]. We train text2room using our text prompts, following its official guidance. We generated panoramic images and depth maps using Blender. As the mesh color is assigned to each vertex, we rendered the panoramic images without additional lighting.

LDM3D [7]. We follow the official implementation of LDM3D. We first synthesize a panoramic RGBD image by LDM3D-pano. Then we use TouchDesigner to convert it into a 3D mesh for rendering novel-view images. Fig. A1 illustrates the process in details. The rendering pipeline processes the input depth map as a height map to deform a 3D sphere with a radius of 1, using the input image as the texture map for the sphere. The degree of deformation is controlled by the 'displacement scale' parameter in the Phong shader, which we empirically set to 1 to minimize distortion in perspective view rendering. We position the camera on a circle with a radius of 0.4 and ensure it always points towards the scene's center. Rendering RGB and depth images for the perspective view is straightforward using the renderer TOP. For panoramic views, we configure the renderer TOP to produce dual paraboloid images and then use a projection TOP to convert them into equirectangular panorama images.

3. Prompt List Used for Evaluation

During evaluation, each method generates scenes based on 10 indoor scene prompts. Here we list the prompts:

- a bedroom, realistic photo style, 4k
- a dining room, realistic detailed photo, 4k
- a living room, realistic detailed photo, 4k
- a museum exhibition hall displaying sculptures, realistic detailed photo, 4k
- a car exhibition center, realistic photo style, 4k
- a study room, realistic detailed photo, 4k

[2] https://github.com/threestudio-project/threestudio#prolificdreamer

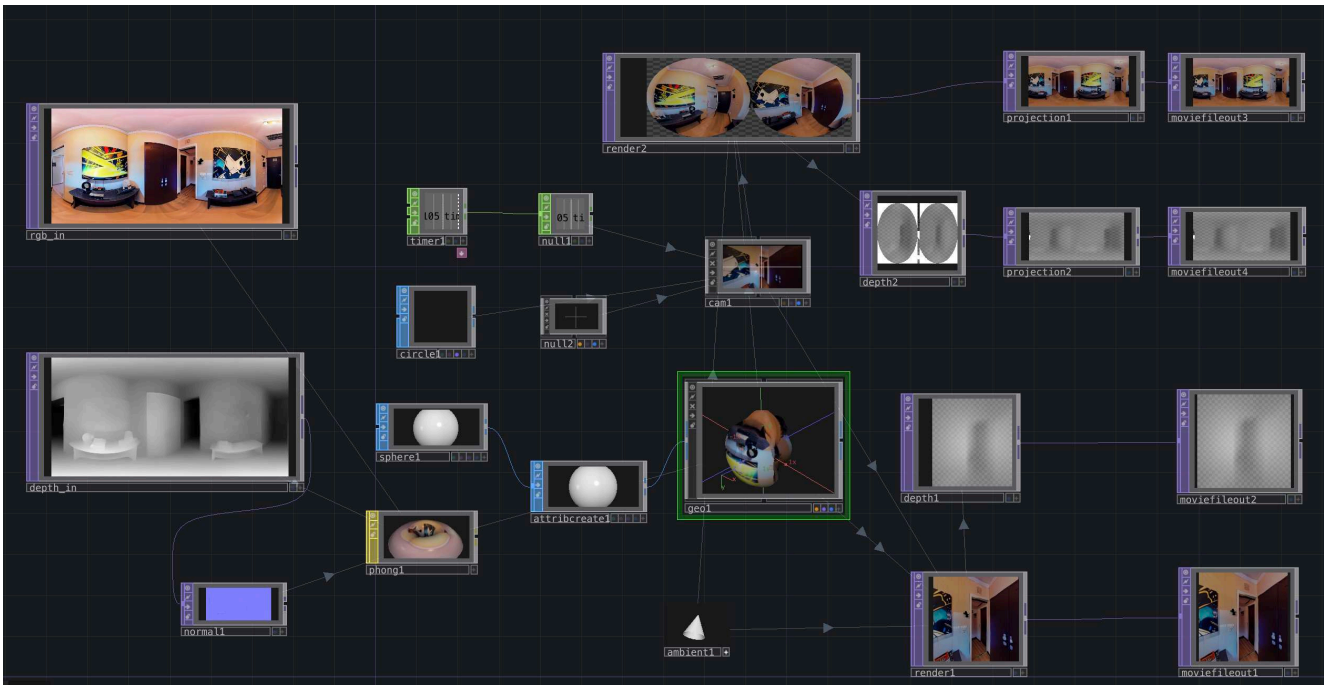

Figure A1. Rendering pipeline of TouchDesigner for LDM3D.

Table A1. **Metrics computed with diverse versions of base models.**

	CLIP-AP$_\uparrow$		Depth-align$_\downarrow$	
	EVA-CLIP	BLIP	ZoeDepth	Depth-Anything
DreamFusion	93.2	89.3	0.30	0.21
ProlificDreamer	91.0	84.5	0.36	0.27
Text2room	73.3	71.9	0.41	0.26
LDM3D	81.8	79.9	0.32	0.22
Ours	**94.2**	**92.2**	**0.18**	**0.19**

- a table tennis room, realistic detailed photo, 4k
- a washing room, realistic detailed photo, 4k
- a classroom, realistic detailed photo, 4k
- a computer laboratory, realistic detailed photo, 4k

4. Metrics computed with different backbones

CLIP similarity and depth estimation are both used during the optimization and evaluation process. To ensure a fair comparison, we incorporate different base models to compute the metrics. Concretely, we use EVA-CLIP [8] and BLIP [4] (instead of CLIP)'s image encoders for measuring image-text similarity, and two state-of-the-art monocular depth prediction models, ZoeDepth [1] and Depth-Anything [10] (instead of MiDaS) for depth alignment estimation. Tab. A1 shows that our method consistently outperforms all baselines, when evaluated using base models that were not utilized during optimization.

5. Fréchet Inception Distance over Disparity Map

As we observe severe visual artifacts exist in rendered disparity map of baseline methods (floating, distortion, blurriness, and discontinuity), we would like to use Fréchet Inception Distance (FID) to assess the image quality. FID is commonly used to evaluate generators trained on real-world images. It utilizes a backbone network that is pretrained on general vision tasks to extract features from each image. By comparing the feature distributions of real dataset images and synthesized fake images, FID quantifies the divergence between these two distributions. Naturally, a question arises regarding the robustness of the backbone network, specifically Inception-v3 in our case, to provide meaningful features that can effectively differentiate between real and fake disparity maps.

To answer this question, we conduct an experiment to verify whether FID exhibits a positive correlation with changes in disparity image quality. To approximate variations in image quality, we test different types of degradatations of the real data, proposed in [2]: **Gaussian noise** is used to approximate the floating artifacts, **Gaussian blur** is used to approximate blurriness, **Swirl** is used to approximate global distortion, and **Implanted black rectangles**

is used to approximate discontinuity and hollows. We use NYU-dep-v2 [5] as the ground truth dataset. This dataset and the images generated from 10 prompts have significant domain shift globally (e.g different number of objects and different types of scenes). On the other hand local pattern such as edges and surfaces should be pretty similar in these two datasets. To this end we should select the metric that is sensitive to local degradation types, such as **Gaussian blur** and **Gaussian noise**, but largely ignore global transformations such as **Swirl**. We first test the features from different layers of Inception-v3, including 64, 192, 2048 channels.

As shown in Fig. A3, initial feature maps with 64 channels is not sensitive to **Gaussian blur**. On the other hand, global features with 2048 channels exhibit an overly intense response to **Swirl**. To this end we opt to utilize features with 192 channels for calculating the FID in all of our experiments, which adequately captures both local transformation, **Gaussian blur** and **Gaussian noise**, but is almost indifferent for global **Swirl** transformation.

Finally, we shown in Fig. A2, that FID with 192 channels adequately captures different disturbance levels. This justifies our choice of 192 features FID as an evaluation metric.

References

[1] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017.

[3] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *ICCV*, 2023.

[4] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[5] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[6] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *Int. Conf. Learn. Represent.*, 2023.

[7] Gabriela Ben Melech Stan, Diana Wofk, Scottie Fox, Alex Redden, Will Saxton, Jean Yu, Estelle Aflalo, Shao-Yen Tseng, Fabio Nonato, Matthias Muller, et al. Ldm3d: Latent diffusion model for 3d. *arXiv preprint arXiv:2305.10853*, 2023.

[8] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

[9] 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.

[10] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024.

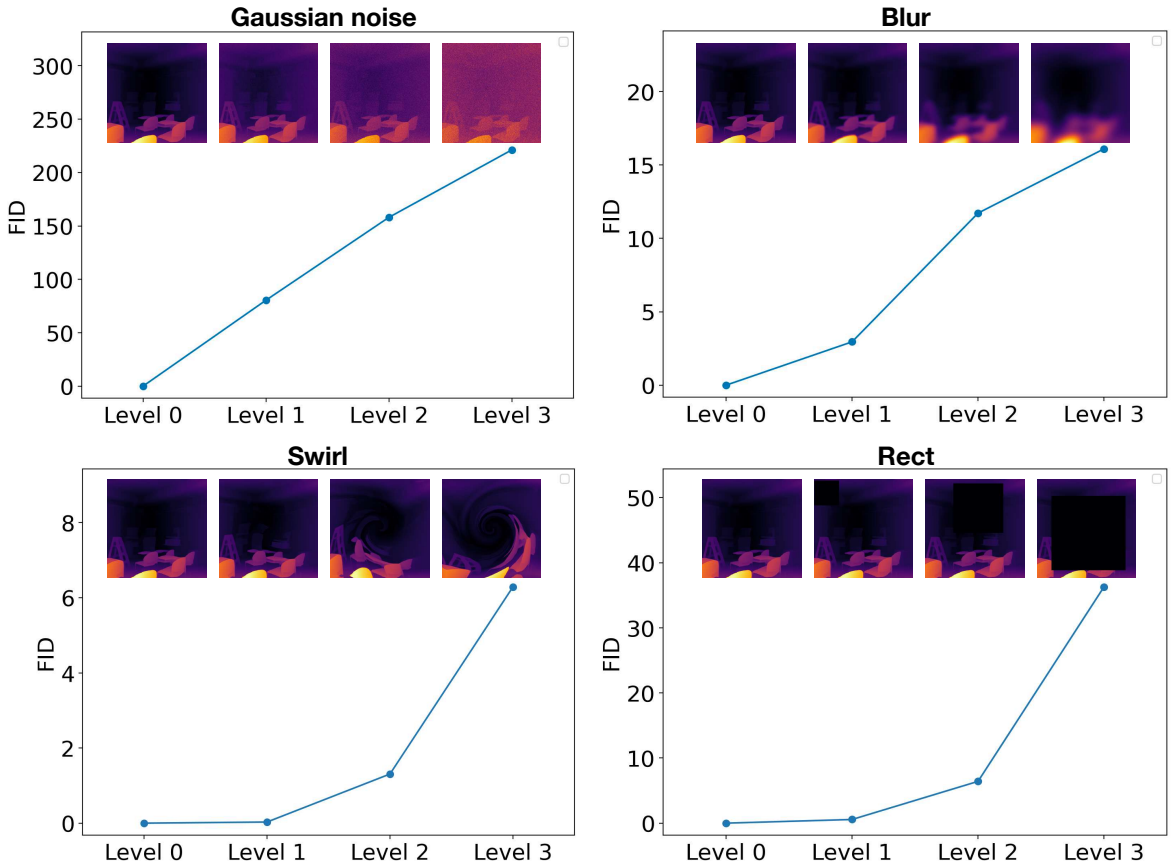

Figure A2. FID is evaluated for **upper left**: Gaussian noise, **upper right**: Gaussian blur, **bottom left**: swirled images, **bottom right**: implanted black rectangles. The disturbance level rises from zero and increases to the highest level. The FID captures the disturbance level very well by monotonically increasing.

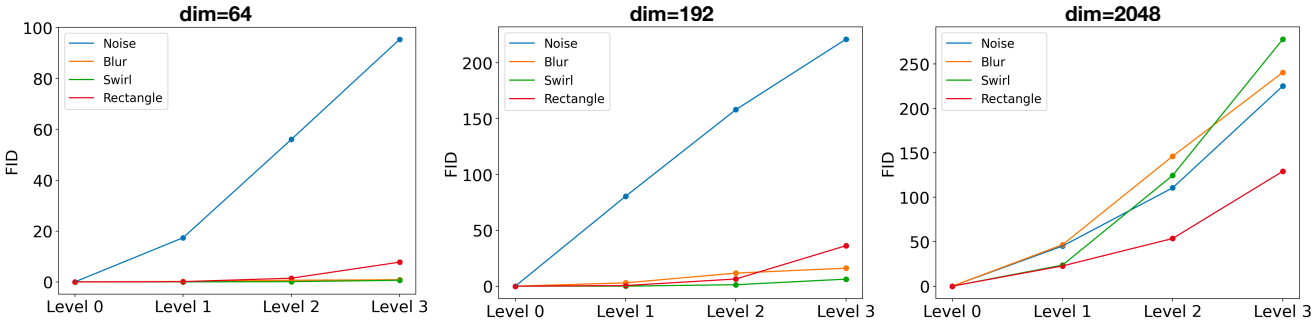

Figure A3. The FID score is calculated for various feature levels, including 64, 192, and 2048 channels. Among these, the mid-level feature with 192 channels exhibits a favorable balance between different types of noise.