

# LAYERPANO3D: Layered 3D Panorama for Hyper-Immersive Scene Generation

Shuai Yang\*, Jing Tan\*, Mengchen Zhang, Tong Wu✉, Yixuan Li, Gordon Wetzstein, Senior Member, IEEE, Ziwei Liu, Member, IEEE, and Dahua Lin✉

**Abstract**—3D immersive scene generation is a challenging yet critical task in computer vision and graphics. A desired virtual 3D scene should 1) exhibit omnidirectional view consistency, and 2) allow for free exploration in complex scene hierarchies. Existing methods either rely on successive scene expansion via inpainting or employ panorama representation to represent large FOV scene environments. However, the generated scene suffers from semantic drift during expansion and is unable to handle occlusion among scene hierarchies. To tackle these challenges, we introduce **LAYERPANO3D**, a novel framework for full-view, explorable panoramic 3D scene generation from a single text prompt. Our key insight is to decompose a reference 2D panorama into multiple layers at different depth levels, where each layer reveals the unseen space from the reference views via diffusion prior. LAYERPANO3D comprises multiple dedicated designs: 1) we introduce a novel text-guided anchor view synthesis pipeline for high-quality, consistent panorama generation. 2) We pioneer the Layered 3D Panorama as underlying representation to manage complex scene hierarchies and lift it into 3D Gaussians to splat detailed 360-degree omnidirectional scenes with unconstrained viewing paths. Extensive experiments demonstrate that our framework generates state-of-the-art 3D panoramic scene in both full view consistency and immersive exploratory experience. We believe that LAYERPANO3D holds promise for advancing 3D panoramic scene creation with numerous applications.

**Index Terms**—3D Scene Generation, Panorama Generation, Diffusion Models, Neural Rendering

## 1 INTRODUCTION

The development of spatial computing, including virtual and mixed reality systems, greatly enhances user engagement across various applications, and drives demand for explorable, high-quality 3D environments. We contend that a desired virtual 3D scene should 1) exhibit high-quality and consistency in appearance and geometry across the full  $360^\circ \times 180^\circ$  view; 2) allow for free exploration among complex scene hierarchies with clear parallax. In recent years, many approaches in 3D scene generation [1], [2], [3] were proposed to address these needs.

One branch of works [4], [5], [6], [7], [8], [9] seeks to create extensive scenes by leveraging a “navigate-and-imagine” strategy, which successively applies novel-view rendering and outpaints unseen areas to expand the scene. However, this type of approaches suffer from the semantic

drift issue: long sequential scene expansion easily produces incoherent results as the out-paint artifacts accumulate through iterations, hampering the global consistency and harmony of the generated scene.

Another branch of methods [10], [11], [12], [13], [14], [15] employs Equirectangular Panorama to represent  $360^\circ$ , large field of view (FOV) environments in 2D. However, the absence of large-scale panoramic datasets hinders the capability of panorama generation systems, resulting in low-resolution images with simple structures and sparse assets. Moreover, 2D panorama [10], [11], [13] does not allow for free scene exploration. Even when lifted to a panoramic scene [16], the simple spherical structure fails to provide complex scene hierarchies with clear parallax, leading to occluded spaces that cause blurry renderings, ambiguity, and gaps in the generated 3D panorama. Some methods [17] typically use inpainting-based disocclusion strategy to fill in the unseen spaces, but they require specific, predefined rendering paths tailored for each scene, limiting the potential for free exploration.

To this end, we present LAYERPANO3D, a novel framework that leverages Multi-Layered 3D Panorama for full-view consistent and free exploratory scene generation from text prompts. The main idea is to create a Layered 3D Panorama by first generating a reference panorama and treating it as a multi-layered composition, where each layer depicts scene content at a specific depth level. In this regard, it allows us to create complex scene hierarchies by placing occluded assets in different depth layers at full appearance.

Our contributions are two-fold. **First**, to generate high-quality and coherent  $360^\circ \times 180^\circ$  panoramas, we propose a novel text-guided anchor view synthesis pipeline. By fine-tuning a T2I model [18] to generate 4 orthogonal perspec-

- S. Yang is with Shanghai Jiao Tong University and Shanghai AI Laboratory, Shanghai 201203, China.  
E-mail: yang\_shuai@sjtu.edu.cn
- J. Tan, T. Wu and Y. Li are with the Multimedia Laboratory, the Chinese University of Hong Kong, Hong Kong SAR.  
E-mail: {tj023, wt020, ly122}@ie.cuhk.edu.hk
- M. Zhang is with Zhejiang University, Zhejiang and Shanghai AI Laboratory, Shanghai, China.  
E-mail: zhangmengchen@zju.edu.cn
- G. Wetzstein is with the Department of Electrical Engineering, Stanford University, Stanford, CA, United States.  
E-mail: gordon.wetzstein@stanford.edu
- Z. Liu is with S-Lab, Nanyang Technological University, Singapore.  
E-mail: ziwei.liu@ntu.edu.sg
- D. Lin is with the Multimedia Laboratory, the Chinese University of Hong Kong, Hong Kong SAR, and Shanghai AI Laboratory, Shanghai, China, and Centre of Perceptual and Interactive Intelligence, Hong Kong SAR.  
E-mail: dhlin@ie.cuhk.edu.hk

\*: equal contribution, ✉: corresponding author.



Fig. 1. **Overview of LAYERPANO3D.** Guided by simple text prompts, LAYERPANO3D leverages multi-layered 3D panorama to create hyper-immersive panoramic scene with  $360^\circ \times 180^\circ$  coverage, enabling free 3D exploration among complex scene hierarchies.

tive views as anchors, we prevent semantic drifts during panorama generation, while ensuring a consistent horizon level across all views. Furthermore, the anchor views enrich the panorama by incorporating complex structures and detailed features derived from large-scale, pre-trained perspective image generators. **Second**, we introduce the Layered 3D Panorama representation as a general solution to handle occlusion for different types of scenes with complex scene hierarchies, and lift it to 3D Gaussians [19] to enable free 3D exploration. By leveraging pre-trained panoptic segmentation prior and K-Means clustering, we streamline an automatic layer construction pipeline to decompose the reference panorama into different depth layers. The unseen space at each layer is synthesized with a finetuned panorama inpainter [11].

Extensive experiments demonstrate the effectiveness of LAYERPANO3D in generating hyper-immersive layered panoramic scene from a single text prompt. LAYERPANO3D surpasses state-of-the-art methods in creating coherent, plausible, text-aligned 2D panorama and full-view consistent, explorable 3D panoramic environments. Furthermore, our framework streamlines an automatic pipeline without any scene-specific navigation paths, providing more user-friendly interface for non-experts. We believe that LAYERPANO3D effectively enhances the accessibility of full-view, explorable AIGC 3D environments for real-world applications.

## 2 RELATED WORKS

In recent years, advancements in AI, particularly deep learning, along with increased computational power and large datasets, have driven rapid development in 2D and

3D generation [20]. These technologies have revolutionized digital content creation in fields like entertainment, design, and virtual reality.

### 2.1 3D Representation

Early neural rendering methods [21], [22], [23], [24] typically incorporate a multi-layer perceptron (MLP) to model 3D scenes as continuous functions. NeRF [25], being the representative MLP-based method, achieves state-of-the-art novel view synthesis performance due to the volume rendering and inductive bias nature of MLPs. However, as the MLP needs to evaluate on large amount of points along every camera ray, the rendering process becomes extremely exhaustive. Subsequent methods [19], [26], [27], [28], [29] improve on faster training and rendering by employing sparse, advanced scene representation. Among these, 3D Gaussian Splatting (3DGS) [19] has recently emerged as a rising star with transformative real-time rendering and high-definition resolutions. It explicitly parameterizes the 3D scene with 3D Gaussians, where each Gaussian is optimized to represent a volume and projected to 2D for rasterization. The discrete nature of 3D Gaussians enables flexible scene editing and composition for 3DGS. Therefore, our approach employs 3DGS as scene representation to facilitate multi-layer scene creation.

### 2.2 3D Scene Generation

Due to the recent success of diffusion models, 3D scene generation has also achieved some development. Scenescape [7] and DiffDreamer [30], for example, explore perpetual view generation through the incremental construction of 3D scenes. One major branch of work employ step-by-step

inpainting from pre-defined trajectories. Text2Room [6] creates room-scale 3D scenes based on text prompt, utilizing textured 3D meshes for scene representation. Similarly, LucidDreamer [4] and WonderJourney [5] can generate domain-free 3D Gaussian splatting scenes from iterative inpainting. However, this line of work often suffer from the semantic drift issue, resulting in unrealistic scene from artifact accumulation and inconsistent semantics. While some other approaches [3], [31], [32] endeavor to integrate objects with environments, they yield relatively low quality of comprehensive scene generation. Recently, our concurrent works, DreamScene360 [16] and HoloDreamer [17] also employ panorama as prior to construct panoramic scenes. However, they only achieve the  $360^\circ \times 180^\circ$  field of view at a fixed viewpoint based on a single panorama of low-quality and simple structure, and do not support free roaming within the scene. In contrast, our framework leverages Multi-Layered 3D Panorama representation to construct high-quality, fully enclosed scenes that enable unconstrained navigation paths in 3D scene.

### 2.3 Panorama Generation

Panorama generation methods are often based on GANs or diffusion models. Early in this field, with the different forms of deep generative neural networks, GAN-based panorama generation methods explore many paths to improve quality and diversity. Among them, Text2Light [15] focuses on HDR panoramic images by employing a text-conditioned global sampler alongside a structure-aware local sampler. However, training GANs is challenging and they encounter the issue of mode collapse. Recently, some studies have utilized diffusion models to generate panoramas. MVDiffusion [10] generates eight perspective views with multi-branch UNet but the resulting closed-loop panorama only captures the  $360^\circ \times 90^\circ$  FOV. The image generated from MultiDiffusion [33] and Syncdiffusion [34] is more like a long-range image with wide horizontal angle as they do not integrate camera projection models. PanoDiff [35] can generate  $360^\circ$  panorama from one or more unregistered Narrow Field-of-View (NFoV) images with pose estimation and controlling partial FOV LDM, while the quality and diversity of results are limited by the scarcity of panoramic image training data like most other methods [14], [36], [37]. In contrast, our model can generate Multi-Layered 3D Panorama for immersive, high-quality, and coherent scene generation from text prompts.

## 3 METHOD

The goal of our work is to create a panoramic scene guided by text prompts, that encompasses a complete  $360^\circ \times 180^\circ$  field of view from various viewpoints within an extensive range in the scene, while allowing for unconstrained trajectory for immersive exploration. LAYERPANO3D consists of three stages. In **Stage I** (Sec. 3.1), we propose a text-guided anchor view synthesis pipeline paired with panoramic out-painting to generate high-quality, consistent panorama as reference. In **Stage II** (Sec. 3.2), with the reference panorama, we construct our Layered 3D Panorama representation by iterative layer decomposition, completion and alignment

process. In **Stage III** (Sec. 3.3), the Layered 3D Panorama is lifted to 3D Gaussians in a cascaded manner to enable free 3D exploration.

### 3.1 Reference Panorama Generation

**Text-Guided Anchor View Synthesis.** We start by generating four orthogonal anchor views to establish the fundamental geometric structure and visual appearance. This is achieved by fine-tuning a pre-trained perspective T2I diffusion model.

For data preparation, we construct a high-quality RGB panorama dataset consisting of 911 outdoor images sourced from the web and 9684 images from Matterport3D [38]. For each panorama, we use llava-v1.5-7b [39] to generate captions as text prompts and project four orthogonal views without overlap at a fixed FOV ( $60^\circ$ ), elevation ( $0^\circ$ ) and four azimuths ( $0^\circ, 90^\circ, 180^\circ, 270^\circ$ ). This setting can effectively avoid perspective distortion and fisheye effect, and allow further refinement of consistency in the subsequent synthesis stages. Additionally, we arrange these four views into a  $2 \times 2$  grid image in a fixed order for the ease of training and draw a white cross-line with a certain pixel size between the grids as a split.

Compared with object-level multi-view generation, it is harder for scene-level generation to diffuse more views with high quality and consistency in a single pass. Multi-view cross-attention-based methods [10], [11] learn high-level semantics at the expense of overfitting to monotonous patterns due to the scarcity of panorama data. Other approaches [1] require huge computation and time in both training and inference stages and generate relatively simple scenes with limited assets. Therefore, inspired by [40], we directly fine-tune Stable-Diffusion-XL [18] to efficiently generate high-quality anchor views as image grids for subsequent panorama generation, free of complex regularization techniques. This text-guided anchor view synthesis pipeline can scale positively with the potency of the underlying T2I model and unlock the capability of the base model without additional knowledge through lightweight fine-tuning. Furthermore, unlike the clean background in the object-level generation, scene backgrounds are often diverse and complex. Thus, instead of simply initializing each sub-grid with a 2D Gaussian blob as is commonly done in object-based multi-view approaches, we initialize the image grids with random colored noise and draw a white cross-line between the grids during the inference process, mimicking the training data.

**Equirectangular Panorama Synthesis.** Equirectangular Projection (ERP) is a method that maps a 3D sphere onto a 2D map. In ERP, the lines of longitude are mapped to vertical lines with constant spacing, and the lines of latitude are mapped to horizontal lines with constant spacing. This mapping relationship is neither area-preserving nor angle-preserving and the mapping of latitude leads to strong distortions in the polar regions. To obtain a detailed, consistent and plausible panorama, we first generate a  $360^\circ \times 90^\circ$  FOV panorama to obtain the principle structure and content, then successively extend it to  $360^\circ \times 180^\circ$  FOV panorama.

With the text-guided anchor view synthesis, we have the anchor perspective images  $[I_{anc}^i]_{i=1}^4 \in \mathbb{R}^{h \times w \times 3}$  of a scene

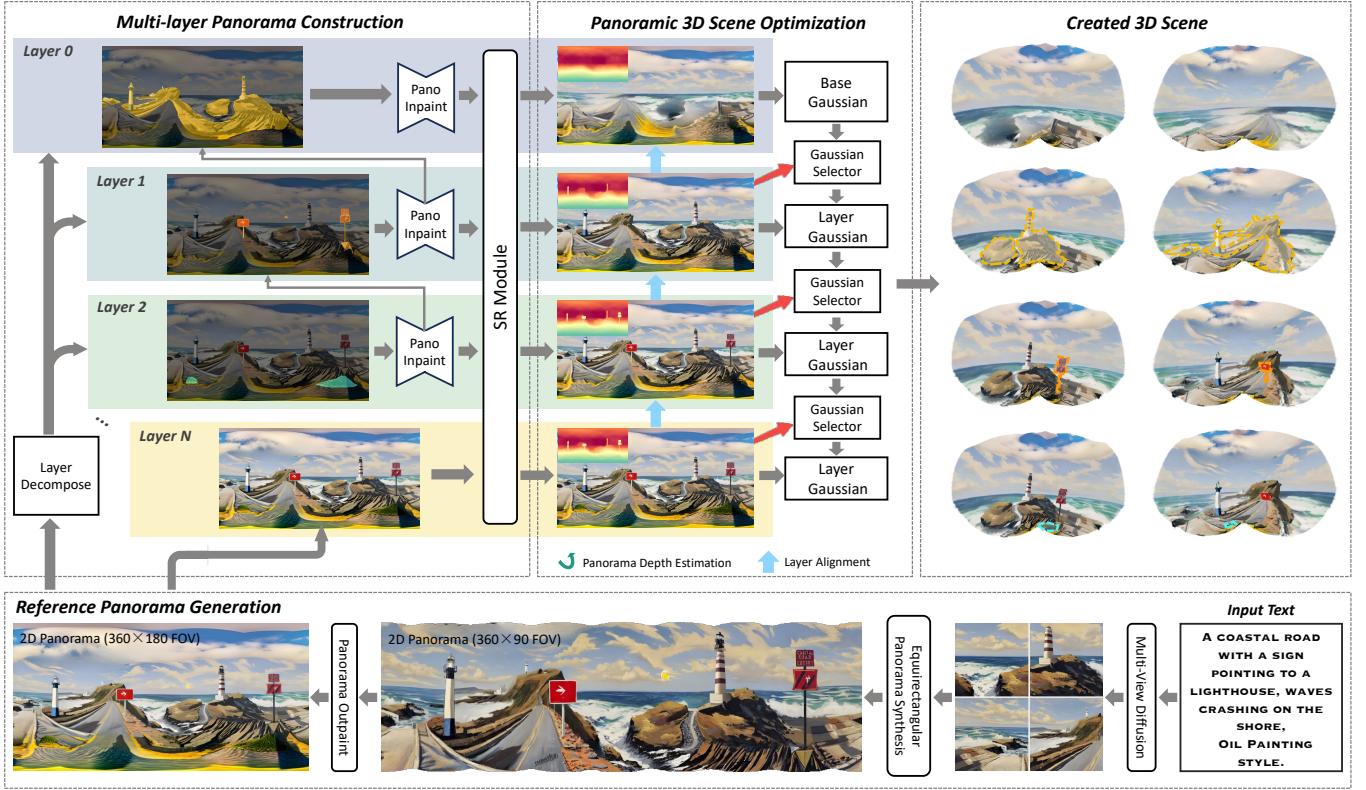


Fig. 2. **Pipeline Overview of LAYERPANO3D.** Our framework consists of three stages, namely reference panorama generation, multi-layer panorama construction and panoramic 3D scene optimization. LAYERPANO3D streamlines an automatic generation pipeline without manual efforts to design scene-specific navigation paths for expansion or completion.

captured by four orthogonal camera in the horizontal plane. Based on the mapping rule and  $60^\circ$  FOV, these anchor views can be directly projected to form the incomplete  $360^\circ \times 60^\circ$  FOV panorama without overlap. Here, we assume that the elevation angle of the anchors are  $\phi_{anc} = 0^\circ$  with azimuth angles  $\{\theta_{anc}^i\} = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ . Then, we set a viewpoint sequence with  $\{(\theta_j, \phi_j)\}_{j=1}^n$  that allows for a 360-degree rotation around the scene in  $90^\circ$  FOV to project the perspective views  $\hat{I}_j$  and masks  $M_j$ :

$$[\hat{I}_j, M_j] = \mathcal{ERP}^{-1}(Pano, [\theta_j, \phi_j, FOV]), \quad (1)$$

where the  $\mathcal{ERP}^{-1}$  is the inverse ERP from panorama to perspective views in 3D. We further apply a text-guided inpainter  $\mathcal{F}_{inpaint}$  [41] to synthesize the missing pixels in the masked regions  $M_j$  of the incomplete views  $\hat{I}_j$ . Note that,  $\hat{I}_j$  is additionally pre-processed with LaMa [42] before inpainting to clean up the masked pixel distribution with contextual knowledge:

$$I_j = \mathcal{F}_{inpaint}(LaMa(\hat{I}_j, M_j)). \quad (2)$$

Through iterative projection and inpainting, the  $Pano_{360 \times 60}$  is extended to  $Pano_{360 \times 90}$ .

The extension from  $Pano_{360 \times 90}$  to  $Pano_{360 \times 180}$  is more difficult, as the polar regions exhibit most distorted patterns that cannot simply be synthesized through perspective image inpainting. Therefore, we fine-tune [43] on panoramic image-caption pairs for polar outpainting. Another challenge is to enforce continuity between the leftmost and rightmost sides of the synthesized image during outpainting, as it is a crucial characteristic [44] of 360° panoramas.

Following [14], [33], we adopt a circular blending strategy in both denoising and VAE decoding stage, as illustrated in Figure 3. A circular closed loop is created by extending the rightmost side of the panorama with the pixels in the leftmost parts. Per-pixel adaptive weights  $W_k$  are utilized to blend the denoised patches mapped from extensions and from the leftmost parts to generate the leftmost panorama. Similarly, in the VAE tiled decoding stage, to avoid tiling artifacts, we ensure that each tile overlaps and blends together to form a smooth output due to each tile using a different decoder. Therefore, each pixel of the panorama is computed as a weighted average of all its mapped patch updates  $Z_{k,t}$  at each denoising step.

$$\Psi(\mathcal{P}_t) = \sum_k \frac{\mathcal{F}_{map,k}^{-1}(W_k)}{\sum_m \mathcal{F}_{map,m}^{-1}(W_m)} \odot \mathcal{F}_{map,k}^{-1}(\Phi(Z_{k,t})), \quad (3)$$

where  $\mathcal{F}_{map,k}^{-1}$  is image space mapping from patch  $Z_k$  to its corresponding latent in  $\mathcal{P}$ ;  $\Psi$  and  $\Phi$  denote the diffusion process at panorama-level and patch-level. Finally, we can get a high-quality result as reference for the subsequent layered panorama construction.

### 3.2 Multi-Layer Panorama Construction

We introduce the Layered 3D Panorama representation based on the following assumption: “an enclosed 3D scene contains a background and various assets positioned in front of it”. In this regard, using Layered 3D Panorama for 3D scene generation is a general approach to handle occlusion for various types of scene. To create a complete scene from the

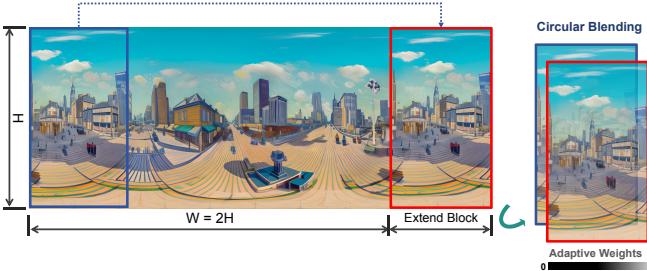


Fig. 3. **The circular blending Strategy.** During the inference process, to synthesize a 360° seamless panorama, adaptive weights are utilized to blend the leftmost part of the latent with the rightmost part in denoising and VAE decoding stage.

reference panorama, we propose to decompose it into  $N + 1$  layers along the depth dimension. As shown in Figure 2, these layers, arranged from farthest to nearest, represent both the scene background (layer 0) and the layouts situated behind the observation point.

**Layer Decomposition.** As shown in Figure 2, the reference panorama is decomposed by first identifying the scene assets and then cluster these assets in different layers according to depth. First, we employ an off-the-shelf panoptic segmentation model [45] pretrained on ADE20K [46] to automatically find all scene assets visible in the reference panorama. A good layer decomposition requires that the layer assets share a similar depth level within layers and are distant from assets in other layers. In this sense, we assign each asset a depth value and apply K-Means to cluster these masks into different groups. Given the reference panorama depth map, the depth value for each asset mask is determined by calculating the 75th percentile of the depth values within the masked region. According to the depth values, the assets are clustered into  $N$  groups from layer 0 to  $N - 1$  and are merged into layer masks to guide the subsequent layer completion.

**Layer Completion.** With the layer mask, we focus on completing the unseen content caused by asset occlusion. In order to synthesize background pixels instead of creating new elements, we finetune PanFusion [11] with the ERP-perspective aware cross-attention as the panoramic background inpainter. Specifically, at each layer, our model takes the layer mask  $M_l$ , the reference panorama, and the “empty scene, nothing” [47] prompt as input, and output coherent content at the masked area. The inpainted panorama at layer  $l$  is denoted  $P_l$  and is used as supervision to the subsequent panoramic 3D Gaussian scene optimization. Note that, we additionally apply SAM [48] to extend the layer mask, based on the inpainted panorama from the previous layer, to eliminate unwanted new generations from inpainting.

Moreover, to enable free rendering in 3D, where observers can examine scenes from varying distances, the unprocessed textures of distant assets may appear blurred as the observer approaches. Therefore, distant layers require higher resolution to preserve texture details at different viewpoints. To address this, Super Resolution (SR) module [49] is employed to enhance the resolution of the layered panorama from layer 0 (background layer) to layer  $N$  (reference panorama), achieving a  $2 \times$  upscale in resolution. SR processing significantly improves the texture quality of

distant objects, maintaining their visual clarity and texture details even when observed from a closer perspective.

**Layer Alignment.** Given the Layered 3D RGB Panorama  $[P_l]_{l=0}^N$ , we perform the depth prediction and alignment to ensure consistency in a shared space. To begin with, we apply the 360MonoDepth [50], to first estimate the layer  $N$  (reference panorama) as the reference depth  $P_{depth}^N$ . The predicted depth  $P_{depth}$  inherently contains extremely small values from normalization, causing peak-like artifacts under the reference camera position. Consequently, we post-process the directly output disparity map  $\hat{P}_{disp}$  with the following transformation to mitigate the geometry distortion:

$$\begin{aligned} D_{bias} &= (\max(\hat{P}_{disp}) - \min(\hat{P}_{disp})) / 4, \\ \hat{P}_{depth} &= \max(\hat{P}_{disp}) - \hat{P}_{disp} + D_{bias}, \\ P_{depth} &= \hat{P}_{depth} / \max(\hat{P}_{depth}). \end{aligned} \quad (4)$$

To align the layer depth in 3D space, we find it infeasible to simply compute a global shift and scale as in [4], [6] due to the nonlinear nature of ERP. Therefore, we leverage depth inpainting model  $\mathcal{F}_{depth}$  from [51] to directly restore depth values based on the inpainted RGB pixels.  $\mathcal{F}_{depth}$  harnesses strong generalizability from large-scale diffusion prior and synthesizes inpainted depth values at an aligned scale with the base depth. We start from reference panoramic depth  $P_{depth}^N$  to implement step-by-step restoration from layer  $N - 1$  to layer 0:

$$P_{depth}^l = \mathcal{F}_{depth}(P_l, M_l \odot P_{depth}^{l+1}), \quad (5)$$

where, in layer  $l$ , the inpainted panorama  $P_l$  and masked depth map  $M_l \odot P_{depth}^{l+1}$  are provided as inputs to  $\mathcal{F}_{depth}$  for restoration.

### 3.3 Panoramic 3D Gaussian Scene Optimization

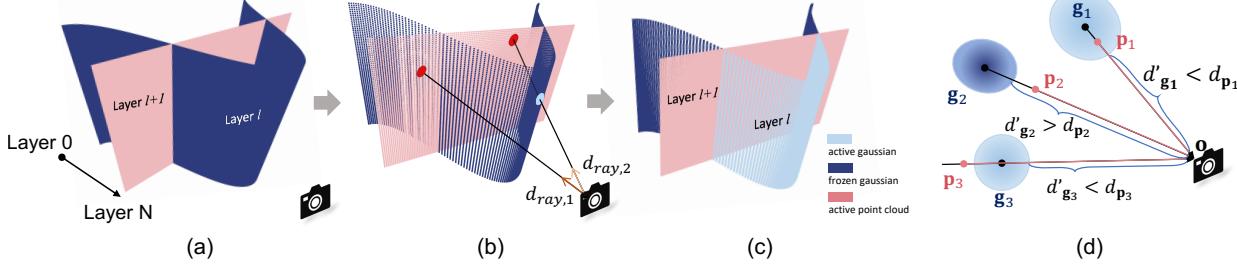
**3D Scene Initialization.** To enable free 3D exploration, we lift the Layered 3D Panorama to 3D Gaussians [19], where the Gaussians are initialized from the layered 3D panoramic point clouds. Considering the intrinsic spherical structure of panorama, we can easily transform an equirectangular image  $P \in \mathbb{R}^{H \times W \times 3}$  into 3D point cloud  $S(\theta, \phi, P_{depth})$ . Each pixel  $(u, v)$  is represented as a 3D point and the angles  $\theta, \phi$  are computed as  $\theta = (2u/W - 1)\pi, \phi = (2v/H - 1)\pi/2$ .

Then, the corresponding 3D coordinates  $(X, Y, Z)$  from the depth value  $P_{depth}(\theta_u, \phi_v)$  are derived as follows:

$$\begin{aligned} X &= P_{depth}(\theta_u, \phi_v) \cos \phi_v \cos \theta_u, \\ Y &= P_{depth}(\theta_u, \phi_v) \sin \phi_v, \\ Z &= P_{depth}(\theta_u, \phi_v) \cos \phi_v \sin \theta_u. \end{aligned} \quad (6)$$

Based on this transformation, we can extract the point cloud for each layer panorama to initialize 3D Gaussians.

Drastic depth changes at layout edges introduce noisy stretched outliers that would turn into artifacts during scene refinement. Therefore, we propose an outlier removal module that specifically targets stretched point removal using heuristic point cloud filtering strategies. As stretched points are usually sparsely distributed in space, we design the point filtering strategy based on its distance from the neighbors. First, we filter out all points with the minimum distance to neighbors over threshold  $\beta_1$ . Then, we eliminate



**Fig. 4. Illustration of the Gaussian Selector.** Given the new asset point cloud, the Gaussian Selector identifies the active Gaussians for next layer’s optimization. (a) Optimized Gaussians at layer  $l$  partially blocks new asset point cloud in layer  $l + 1$ . (b) By computing the viewing directions and the distance to camera, Gaussian Selector identifies the blocking Gaussians. (c) The marked Gaussians are labeled as active, and merged with the new points for optimization. (d) Showed a special case where the optimized Gaussian center is behind the new points, but also identifies as an active Gaussian because its volume scale overlaps with the points (case 1).

points with very few neighbors. The idea is to calculate the number of neighbors of each point within a given radius and drop the points where their number of neighbors is below threshold  $\beta_2$ . To speed up the calculation, we map the points into 3D grids, then remove all points within grids that have less than  $\beta_2$  number of neighbors.

**3D Scene Refinement.** During scene refinement, we devise two types of Gaussian training schemes for varying scene content: the *base Gaussian* for reconstructing the scene background and the *layer Gaussian* for optimizing scene layouts. Additionally, a Gaussian selector module is introduced between layer Gaussians to facilitate scene composition.

In scene refinement, the base Gaussian model is initialized on a whole of the background point cloud, and the layer Gaussian model initiates on and optimizes the foreground assets. In practice, we project the layer mask  $M_l$  onto point clouds and use the masked points to initiate Gaussians. The optimized Gaussians from previous layers are frozen to avoid unwanted modification. In this way, the scene background is optimized once in the base Gaussian to reduce unnecessary computation and conflicts of Gaussians in subsequent layers.

We observe that the quality of the optimized scene is easily hampered by unaligned layers, and sometimes  $\mathcal{F}_{depth}$  does fail to produce perfectly aligned layer depths. Gaussians at layer  $l$  could span into unwanted depth levels and block assets in the subsequent layer, as illustrated in Figure 4(a). To handle this issue, we introduce the Gaussian selector module to detect these conflicted Gaussians, reactivate them from frozen, and optimize them away from the blockage. First, the selector computes the distance vector from the camera center  $\mathbf{o} = (0, 0, 0)$  to each new point  $\mathbf{p}$ , as in Figure 4(b). The absolute distance from asset points  $\mathbf{p}$  and scene Gaussians  $\mathbf{g}$  to the camera is denoted as  $d_{\mathbf{p}}$  and  $d_{\mathbf{g}}$  respectively:

$$d_{\mathbf{p}} = \|\mathbf{p} - \mathbf{o}\|_2, \quad d_{\mathbf{g}} = \|\mathbf{g} - \mathbf{o}\|_2. \quad (7)$$

By examining all Gaussians that on the same ray with asset points but at a closer distance:  $\mathbf{p} / d_{\mathbf{p}} = \mathbf{g} / d_{\mathbf{g}}$ ,  $d_{\mathbf{g}} < d_{\mathbf{p}}$ , we mark them as active (Figure 4(c)). We also observe that despite the Gaussian center being behind the asset points, if the scaling of the Gaussian covers the point, then the Gaussian is still affecting the overall rendering quality. Hence, we update the gaussian distance as

$$d'_{\mathbf{g}} = d_{\mathbf{g}} - \max\{\mathbf{s}(\mathbf{g})\}, \quad (8)$$

where  $\mathbf{s}$  is the scaling of the Gaussian  $\mathbf{g}$ . If  $d_{\mathbf{g}_1} > d_{\mathbf{p}_1}$  and  $d'_{\mathbf{g}_1} < d_{\mathbf{p}_1}$  as in Figure 4(d), the referred Gaussian is also marked active. For efficient memory storage and fast look-up, we hash the distance vectors into a 3D grid. The mapping function from vector coordinates to grid indices writes:  $f(\mathbf{p}) = \text{ceil}(\beta_3 \log(\mathbf{p} + 1))$ .

## 4 EXPERIMENTS

### 4.1 Implementation Details

In the anchor view synthesis stage, we train the model starting from SDXL-base-1.0 [18] with a batch size of 16 and learning rate of  $10^{-5}$  for only 10K iterations on the 11k data pairs. The training is done using 4 NVIDIA A100 GPUs for 7 hours. For inference, we initialize four  $512 \times 512$  grids with random colored noise, subsequently superimposing a white cross-line with a pixel width of 10 to ensure that the final inference result can be accurately segmented into four grids of size  $507 \times 507$ . In the reference panorama synthesis stage, we leverage the Stable Diffusion Inpainting pipeline [41] from huggingface [52] as the inpainter and repurposed Diffusion360 [43] as the outpainter to synthesize  $360^\circ \times 180^\circ$  FOV panorama.

In the layered panorama construction stage, we employ OneFormer [45] to obtain the panoptic segmentation map for the reference panorama. Background categories are manually determined (i.e. sky, floor, ceiling, etc.) to filter out background components in asset masks. Generally, we cluster all asset masks into  $N = 3$  layers via KNN and merge all masks within each layer to form a unified layer mask. With the obtained layer mask, we combine our fine-tuned panorama inpainting model with LaMa [42] to achieve multi-layer completion and apply 360MonoDepth [50] to predict the reference panorama depth.

In the 3D panoramic scene optimization stage, we lift the panorama RGBD into 3D point clouds. For scene initialization, we set  $\beta_1$  to 0.0001 and  $\beta_2$  to 4 based on empirical practice. These point clouds are used to initialize the base Gaussian model and the layer Gaussian model. During the scene refinement stage, we optimize the base Gaussian model for 3,000 iterations, then the layer Gaussians each for 2,000 iterations. The training objective for base and layer Gaussian is the L1 loss and D-SSIM term between the ground-truth views and the rendered views. We use a single 80G A100 GPU for reconstruction and the reconstruction

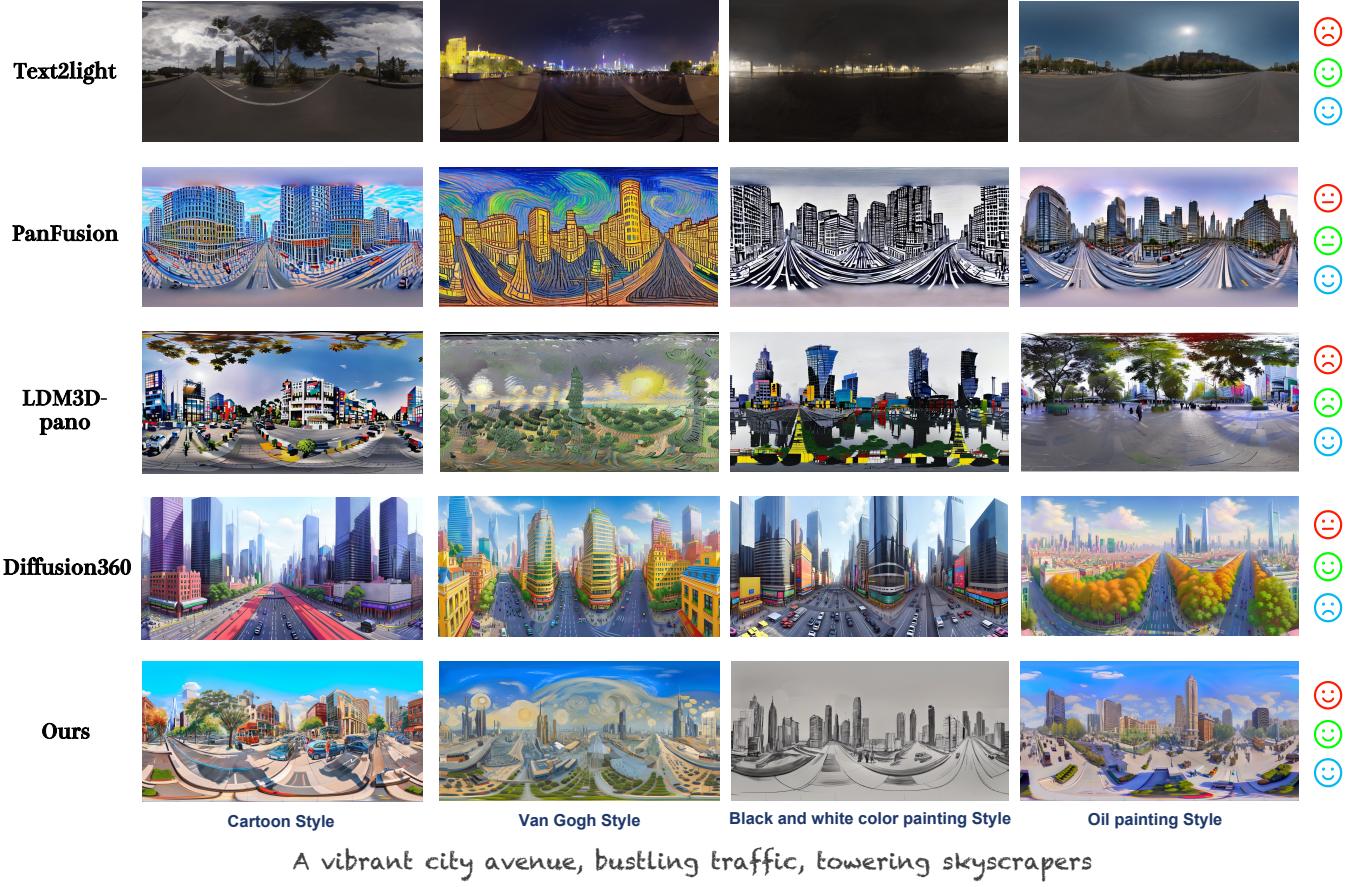


Fig. 5. **Qualitative comparisons in Panorama Generation.** We compare 2D panorama generation methods across three dimensions: *Creativity* (red), *Resolution* (green), and *Spherical Structure* (blue). LAYERPANO3D produces high-resolution, creative, and consistent panoramas while accommodating multi-style generation requirements.

time for each layer is 1.5 minutes on average for  $1024 \times 1024$  resolution inputs.

## 4.2 Comparison Methods.

To evaluate the performance of our approach in the context of text-driven 3D panoramic scene generation, We compare with previous methods in two phases: **2D Panorama Generation** and **3D Panoramic Scene Reconstruction**. For 2D Panorama Generation, we choose four approaches to check the quality and creativity of 2D panorama: **Text2light** [15] adopts the VQGAN [53] structure to synthesize an HDR panorama image from text in a two-stage auto-regressive manner. **Panfusion** [11] designs a novel dual-branch diffusion model to generate a 360-degree image from a text prompt. **Diffusion360** [43] directly uses the DreamBooth [54] training on the panorama dataset [55]. **LDM3D-pano** [12] extends LDM3D [56] to generate a realistic RGB panorama and its corresponding depth. For 3D Panoramic Scene Reconstruction, we select three methods to compare the 3D performance of scenes: **LucidDreamer** [4] starts from a single image and a text prompt to create a 3D-GS scene. **Text2Room** [6] generates a textured 3D mesh scene with preset camera trajectory. **DreamScene360** [16] creates panoramic 3DGS scene with full  $360^\circ$  coverage from text prompts. Note that, since DreamScene360 does not open-

source its official code, we reproduce its model for comparison.

## 4.3 Qualitative Comparison

**2D Panorama Generation.** We show some qualitative comparisons with several state-of-the-art panorama generation works in Figure 5. Text2light ignores style prompts due to being trained on a realistic HDRI dataset based on the VQGAN structure, and the components in the generated panorama are relatively simple. The results by PanFusion are ambiguous and low in quality, while it can adapt to style changes due to its operation in panorama and perspective domains. LDM3D-pano is insensitive to style prompts and the results are low-resolution. The instances generated by Diffusion360 appear to be of higher quality compared to the above, but do not have the inherent spherical structural property of panoramas, and the style seems to be the same. This is because the model is merely trained on the panorama dataset based on Dreambooth [54] without injecting any prior knowledge. Our method achieves the highest quality both in consistency and diversity, presenting creative and reasonable generations.

**3D Panoramic Scene Reconstruction.** To more precisely evaluate the quality of 3D panoramic scene reconstruction, we present qualitative comparisons with Text2Room [6], LucidDreamer [4], and DreamScene360 [16] across two di-

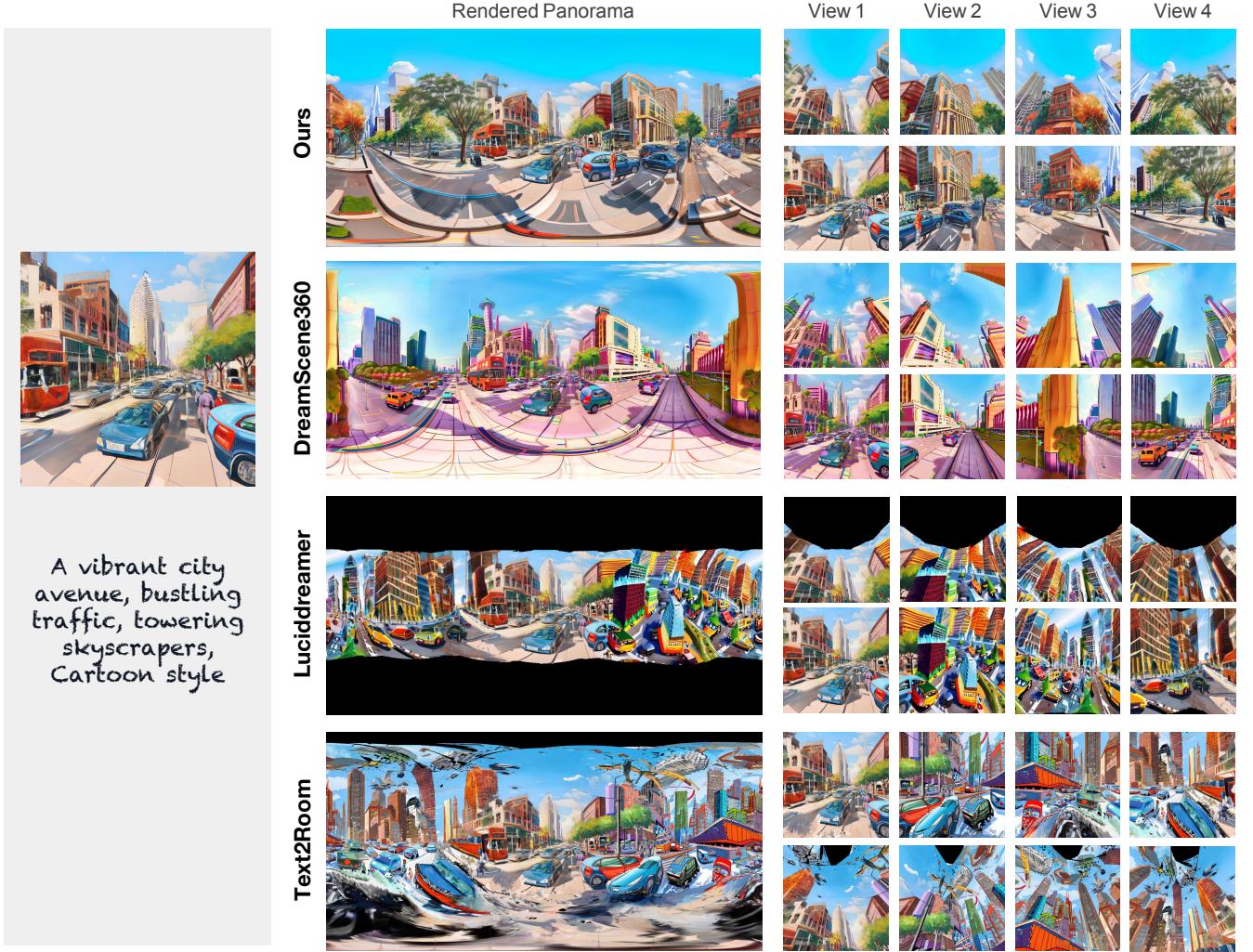
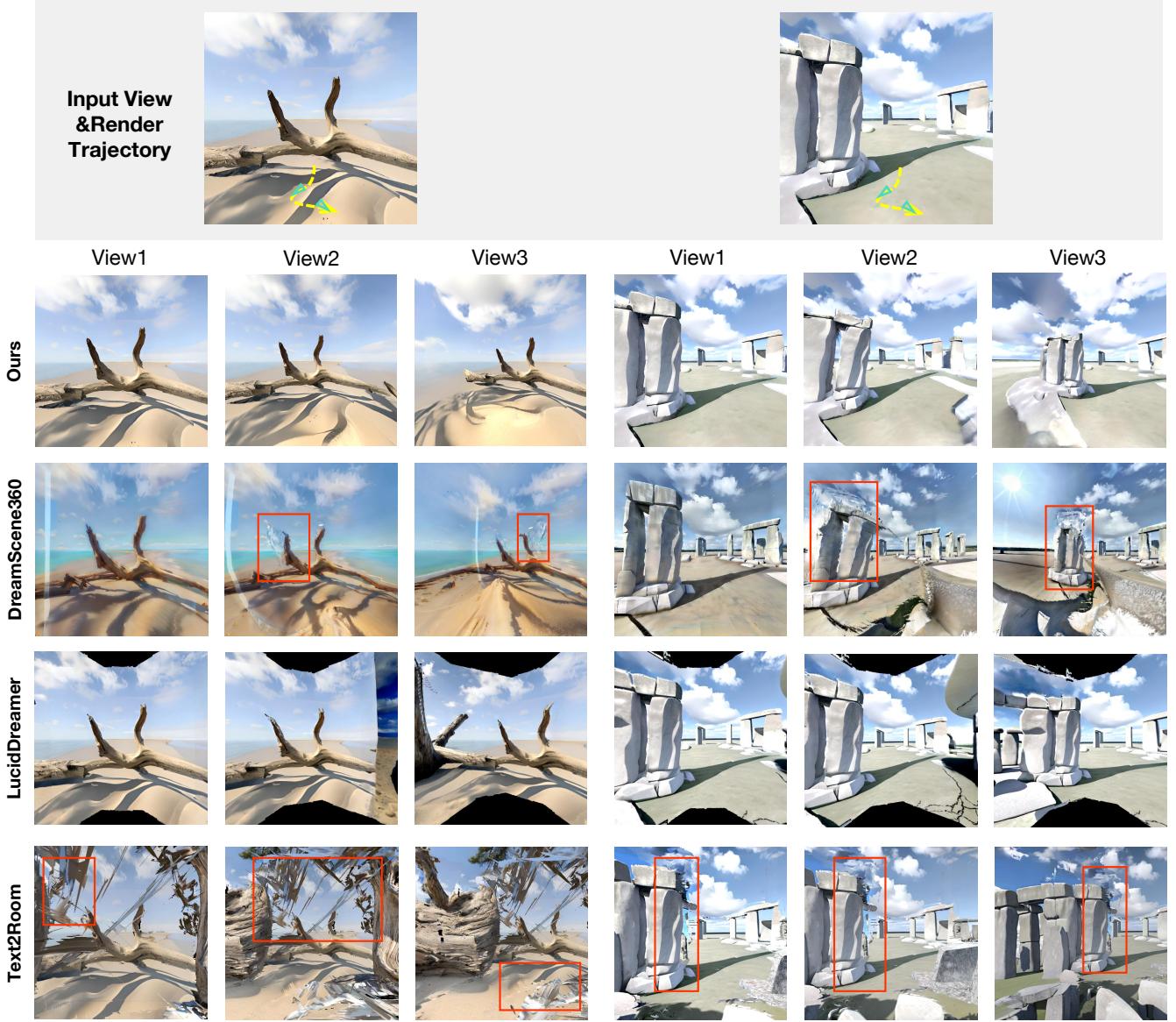


Fig. 6. **Qualitative comparisons in full 360°×180° Scene.** We compare the panorama and multiple views of the scene generated by four methods. LAYERPANO3D exhibits more *consistent* content with full 360° × 180° coverage.

mensions. For full 360° × 180° view consistency, we input an image with a text prompt to guide scene creation and render multiple views from the center of the scene. This approach synthesizes the texture of the panoramic environment, allowing for a more intuitive assessment after generation. As shown in Figure 6, LucidDreamer [4] and Text2room [6] fail to cover the full 360° × 180° view, resulting in semantic incoherence and artifacts due to their successive inpainting-based strategy. DreamScene360 [16] supports a 360° × 180° view at a single fixed viewpoint, but it deviates from the input conditions, and the quality of the generated results is relatively low. In contrast, our model excels in maintaining full 360° × 180° view consistency while demonstrating superior content creativity. To evaluate the capability for free trajectory rendering, we design a zigzag trajectory to guide the camera’s movement through the scene, with novel view renderings sampled along the trajectory for comparison. As illustrated in Figure 7, we showed 3 random samples from this fixed flythrough trajectory. Compared with the other three methods, our model achieves a more complete 3D scene with consistent textures and a reasonable geometric structure.

#### 4.4 Quantitative Comparison

**2D Panorama Generation.** We adopt three metrics for quantitative comparisons: 1) **Intra-Style** [34], [57] evaluates the coherence of the generated panoramas; 2) **FID** [58] evaluates both fidelity and diversity; 3) **CLIP** [59] measures the compatibility of results and input prompts. We selected 73 prompts to generate panoramas for each method and averaged the results for all metrics. Moreover, a user study is also conducted to further evaluate the quality of panoramas, where we project 4 views at a fixed FOV (90°) to the user for sorting. As shown in Table 1, our method achieves the best scores in both FID and CLIP metrics, indicating its high fidelity and strong compatibility with the input prompts. The Intra-Style results demonstrate that our model achieves global coherence across the image, consistently maintaining the overall style. While Text2light has a smaller Intra-Style score, this is primarily due to its tendency to generate monotonous results with extensive uniform color block backgrounds. We also attained outstanding results in aesthetic metrics, marginally trailing Diffusion360. However, as discussed in Section 4.3, panoramas generated by Diffusion360 lack the inherent spherical structural property, resulting in an unreasonable final perspective projection.



**Fig. 7. Qualitative comparisons in Scene Free Exploration.** We show the novel view renderings along a zigzag trajectory to compare the capability of free trajectory rendering. Our method is able to maintain high-quality content rendering and does not show distortion or gaps in unseen space, which shows the ability of LAYERPANO3D to create hyper-immersive panoramic scenes.

Moreover, we conducted a user study comparing generated panoramas, evaluating them across three dimensions: **Coherence** (internal consistency), **Plausibility** (plausibility of perspective views), and **Compatibility** (align with prompts). Following [60], we use the **Average User Ranking (AUR)** as a preference metric where users rank each result on a scale of 1 to 5 (lower is worse). We invite 29 users including graduate students that expertise in 3D and average users to rank the 60 results from 5 methods individually. The results in Table 1 demonstrate that our method outperforms others by a significant margin across all dimensions.

**3D Panoramic Scene Reconstruction.** Following [2], we adopt two widely-used non-reference image quality assessment metrics, **NIQE** [61] and **BRISQUE** [62], to evaluate the capability of free trajectory rendering, i.e. the novel view quality along an immersive navigation path in the 3D scene. Furthermore, we also measure the 3D reconstruction quality on training views via classic reconstruction metrics: **PSNR**,

**SSIM** and **LPIPS** [63]. In addition, we render four orthogonal views, each distributed at a fixed FOV ( $90^\circ$ ), elevation ( $0^\circ$ ) and four equidistant azimuths ( $0^\circ, 90^\circ, 180^\circ, 270^\circ$ ) respectively, to predict perspective field. Building upon [64], we independently predict camera parameters and evaluate the horizon tilt problem by calculating the **Roll-Mean** and **Roll-Var**, i.e. the mean and variance of the elevation angles across these views. As shown in Table 2, our method surpasses the existing methods in both novel view quality metrics (NIQE and BRISQUE) and 3D reconstruction metrics (PSNR, SSIM, and LPIPS). Furthermore, we conducted another user study to further evaluate the quality of the generated 3D panoramic scene from two aspects: 1)  $360^\circ \times 180^\circ$  view consistency and 2) free path rendering quality. For the first aspect, we render 60 frames in a 360-degree view at the 0-degree and 45-degree elevation respectively for evaluation. For the second aspect, we select the same trajectory as in Figure 7 to render navigation

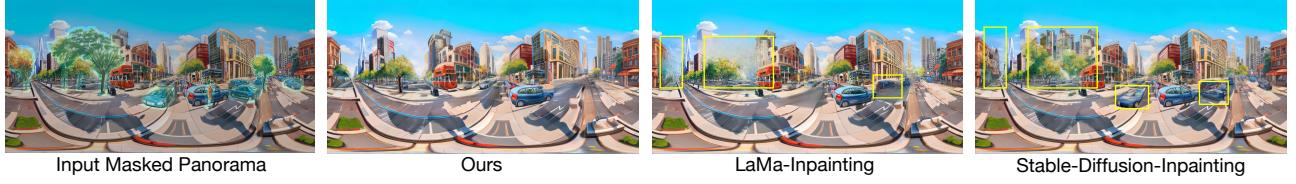


Fig. 8. **Analysis on the Layer Completion Inpainting.** We present the panorama inpainting results for three methods guided by the same text prompt: “empty scene, nothing” [47]. Our model effectively handles complex scenarios, delivering clear results with consistent and coherent structures.

TABLE 1

**Quantitative comparison with SoTA methods on 2D Panorama Generation.** **Bold** indicates the best result, and underline indicates the second-best result.

Method	FID ↓	Intra-Style ↓	CLIP ↑	User Study (AUR)		
				Coherence ↑	Plausibility ↑	Compatibility ↑
Text2light [15]	286.90	<b>0.31</b>	18.69	2.95	2.47	1.89
Panfusion [11]	283.80	18.66	21.22	3.06	3.03	<u>3.26</u>
Diffusion360 [43]	<u>274.03</u>	3.70	<u>21.65</u>	3.08	<u>3.22</u>	<u>3.24</u>
LDM3D-pano [12]	298.64	8.05	20.58	2.56	<u>2.84</u>	2.87
Ours	255.60	<u>2.24</u>	22.08	3.35	<u>3.44</u>	<u>3.73</u>

TABLE 2

**Qualitative comparison with SoTA methods on 3D Panoramic Scene.** **Bold** indicates the best result.

Method	Appearance					Geometry		User Study (AUR)	
	NIQE ↓	BRISQUE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Roll-Mean ↓	Roll-Var ↓	360° × 180° ↑	Free-path ↑
Text2room [6]	5.117	44.598	30.437	0.890	0.033	1.877	1.566	1.76	2.28
LucidDreamer [4]	5.877	54.230	31.020	0.973	0.029	2.775	2.167	1.85	1.29
DreamScene360 [16]	5.083	39.403	28.925	0.941	0.050	1.696	2.524	2.94	2.90
Ours	<b>4.374</b>	<b>39.006</b>	<b>40.797</b>	<b>0.980</b>	<b>0.012</b>	<b>0.540</b>	<b>0.028</b>	<b>3.45</b>	<b>3.53</b>

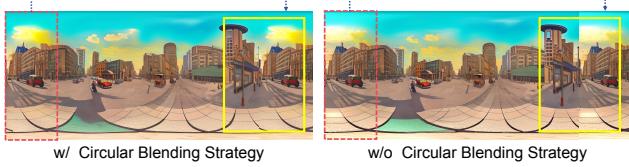


Fig. 9. **Ablation on the Circular Blending Strategy.** The circular blending strategy ensures the continuity between the leftmost and rightmost sides of the panorama, which preserves the crucial characteristic of 360-degree panoramas.

videos for evaluation. We invite 43 users including graduate students that expertise in 3D and average users to rank the 40 results from 4 methods individually. The average ranking is shown in Table 2. Our LAYERPANO3D achieves the best performance in both 360° × 180° view consistency and free path rendering quality among all four approaches.

#### 4.5 Analysis and Ablative Study

**Ablation on Circular Blending.** Figure 9 demonstrates that the circular blending strategy effectively maintains geometric continuity, contributing to the generation of a seamless 360-degree panorama. Without this design, the rightmost and leftmost sides of the panorama could form inconsistent patterns, resulting in a discontinuous panorama.

**Analysis on Layer Completion Inpainting.** We discuss the effectiveness of our fine-tuned panorama inpainter in layer completion. We compare the inpainting results among three approaches: LaMa [42], Stable Diffusion inpainting model and our proposed inpainter. As illustrated in Figure 8, LaMa produces inconsistent texture and blurry artifacts at

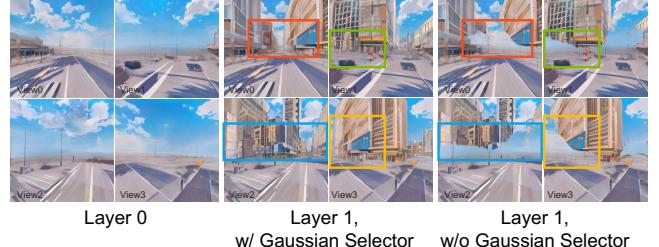
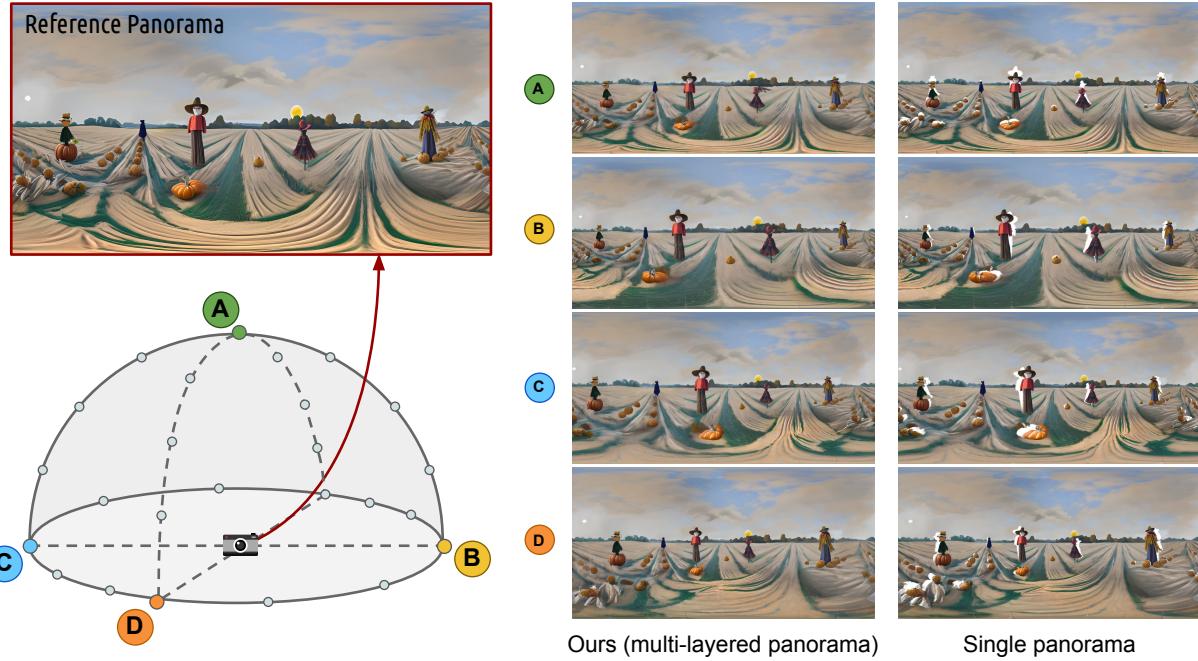


Fig. 10. **Ablation on the Gaussian Selector.** Our Gaussian Selector resolves the optimization conflicts between layers by re-activating the Gaussians from the previous layer that blocks new assets. With the Gaussian Selector, the merged Gaussians are optimized to faithfully reconstruct the ground-truth panorama views.

large-scale inpainting. Stable Diffusion tends to produce distorted new elements due to the domain gap between perspective and panoramic images. In contrast, thanks to the panorama finetuning, our module delivers clean inpainting results with coherent and plausible structures in the masked regions.

**Ablation on Gaussian Selector.** Our Gaussian selector is proposed to select the part of Gaussians that appears in the front of newly added scene assets. By selecting these Gaussians and re-activating them in the optimization, the model achieves accurate appearance and geometry at the current layer. As shown in Figure 10, the leftmost column is the scene Gaussians at layer 0. When adding the building assets at the first layer, the sky Gaussians from the previous layer partially block the building assets (right column). After using the Gaussian selector to select and optimize the sky



**Fig. 11. Analysis on Panorama Rendering at Off-center Viewpoints.** We show that LAYERPANO3D’s robustness in rendering  $360^\circ \times 180^\circ$  consistent panorama at various viewpoints within a large range of space in the generated scene. Compared with our single-layer panorama baseline, LAYERPANO3D renderings exhibit high-quality content without any holes or gaps from occlusion.

Gaussians, these Gaussians learn to either be translucent and pruned for low opacity or move to be a part of the building assets. Therefore in the middle column, we observe a consistent scene with no obvious blockage of the new building assets thanks to the Gaussian Selector.

**Analysis on Panorama Renderings at Off-center Viewpoints.** In Figure 11, we demonstrate that LAYERPANO3D is robust to render consistent panorama images at various locations besides the original camera location in the center. We sample four camera locations on circular trajectories on the hemisphere centered at the origin and render 24 views at  $(-45^\circ, 0^\circ, 45^\circ)$  elevation to compose new panorama images. By evaluating panorama renderings at new viewpoints, we show that our generated panoramic scene is  $360^\circ \times 180^\circ$  consistent and enclosed, robust to various viewpoints at any angle. Compared to the single-layered 3D panorama, our multi-layered 3D panorama exhibits no gaps or holes from the scene occlusion, demonstrating our capability for free 3D exploration in the generated scenes.

## 5 CONCLUSION

In this paper, we propose LAYERPANO3D, a novel framework that generates hyper-immersive panoramic scene from a single text prompt. Our key contributions are two-fold. First, we propose the text-guided anchor view synthesis pipeline to generate detailed and consistent reference panorama. Second, we pioneer the Layered 3D Panorama representation to show complex scene hierarchies at multiple depth layers, and lift it to Gaussians to enable free 3D exploration. Extensive experiments show the effectiveness of LAYERPANO3D in generating  $360^\circ \times 180^\circ$  consistent panorama at various viewpoints and enabling immersive

roaming in 3D space. We believe that LAYERPANO3D holds promise to advance high-quality, explorable 3D scene creation in both academia and industry.

**Limitations and Future Works.** LAYERPANO3D leverages good pre-trained prior to construct panoramic 3D scene, i.e., panoramic depth prior for 3D lifting. Therefore, the created scene might contain artifacts from inaccurate depth estimation. With advancements in more robust panorama depth estimation, we hope to create high-quality panoramic 3D scenes with finer asset geometry.

## REFERENCES

- [1] R. Gao, A. Holynski, P. Henzler, A. Brussee, R. Martin-Brualla, P. Srinivasan, J. T. Barron, and B. Poole, “Cat3d: Create anything in 3d with multi-view diffusion models,” *arXiv preprint arXiv:2405.10314*, 2024.
- [2] H. Li, H. Shi, W. Zhang, W. Wu, Y. Liao, L. Wang, L. hang Lee, and P. Zhou, “Dreamscene: 3d gaussian-based text-to-3d scene generation via formation pattern sampling,” 2024.
- [3] Q. Zhang, C. Wang, A. Siarohin, P. Zhuang, Y. Xu, C. Yang, D. Lin, B. Zhou, S. Tulyakov, and H.-Y. Lee, “Scenewiz3d: Towards text-guided 3d scene composition,” 2023.
- [4] J. Chung, S. Lee, H. Nam, J. Lee, and K. M. Lee, “Luciddreamer: Domain-free generation of 3d gaussian splatting scenes,” *CoRR*, vol. abs/2311.13384, 2023.
- [5] H. Yu, H. Duan, J. Hur, K. Sargent, M. Rubinstein, W. T. Freeman, F. Cole, D. Sun, N. Snavely, J. Wu, and C. Herrmann, “Wonderjourney: Going from anywhere to everywhere,” *CoRR*, vol. abs/2312.03884, 2023.
- [6] L. Höller, A. Cao, A. Owens, J. Johnson, and M. Nießner, “Text2room: Extracting textured 3d meshes from 2d text-to-image models,” *arXiv preprint arXiv:2303.11989*, 2023.
- [7] R. Fridman, A. Abecasis, Y. Kasten, and T. Dekel, “Scenescape: Text-driven consistent scene generation,” 2023.
- [8] J. Zhang, X. Li, Z. Wan, C. Wang, and J. Liao, “Text2nerf: Text-driven 3d scene generation with neural radiance fields,” 2024.

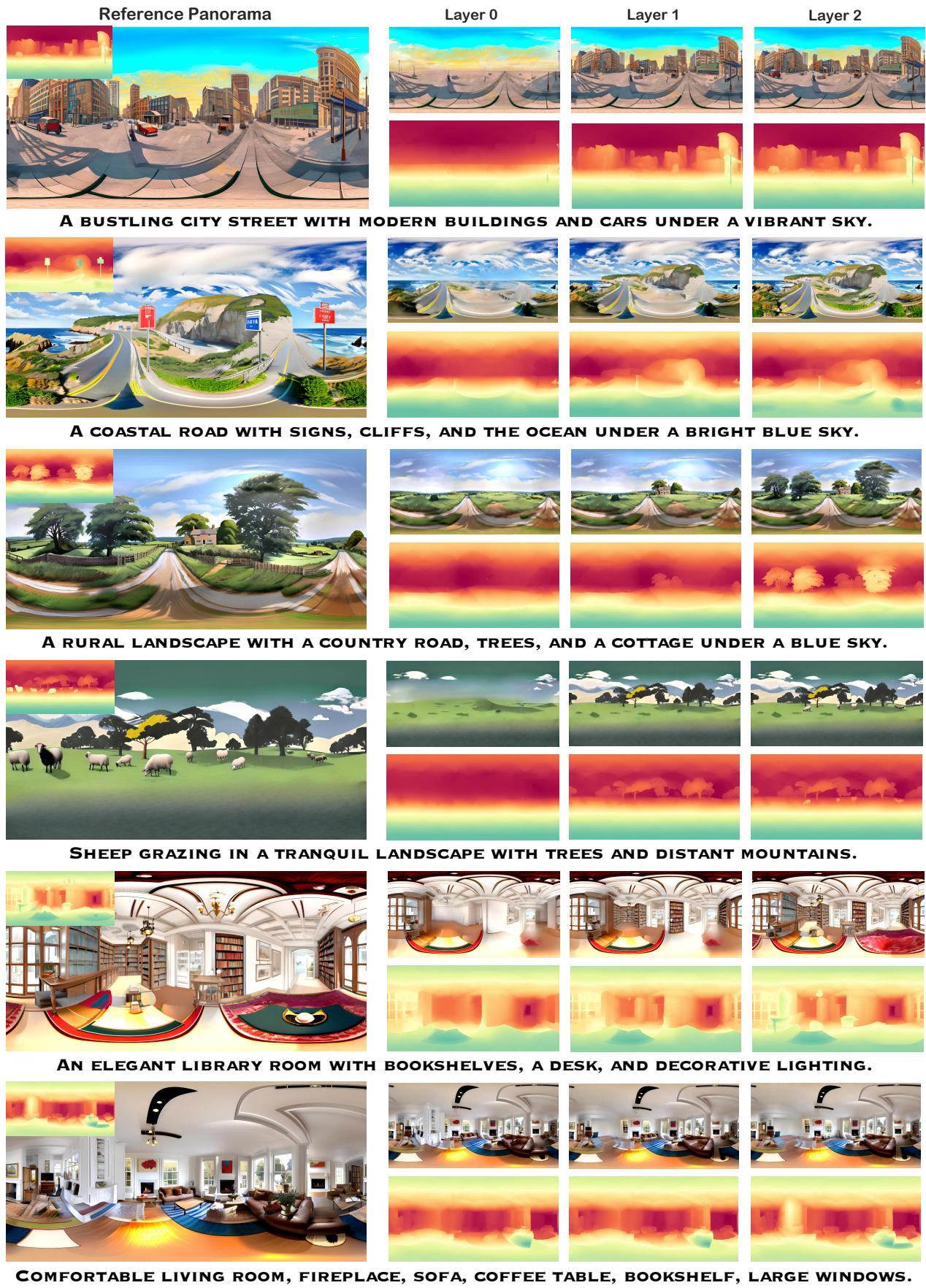


Fig. 12. Additional results of LAYERPANO3D on Diverse Generation. LAYERPANO3D generates various hyper-immersive Layered 3D Panorama that cover cityscape, landscape and indoor environments.

- [9] H. Ouyang, K. Heal, S. Lombardi, and T. Sun, "Text2immersion: Generative immersive scene with 3d gaussians," *arXiv preprint arXiv:2312.09242*, 2023.
- [10] S. Tang, F. Zhang, J. Chen, P. Wang, and Y. Furukawa, "Mvd-diffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion," 2023.
- [11] C. Zhang, Q. Wu, C. C. Gambardella, X. Huang, D. Phung, W. Ouyang, and J. Cai, "Taming stable diffusion for text to 360 panorama image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6347–6357.
- [12] G. B. M. Stan, D. Wofk, E. Aflalo, S.-Y. Tseng, Z. Cai, M. Paulitsch, and V. Lal, "Ldm3d-vr: Latent diffusion model for 3d vr," 2023.
- [13] G. Wang, Y. Yang, C. C. Loy, and Z. Liu, "Stylelight: Hdr panorama generation for lighting estimation and editing," 2022.
- [14] H. Wang, X. Xiang, Y. Fan, and J.-H. Xue, "Customizing 360-degree panoramas through text-to-image diffusion models," 2023.
- [15] Z. Chen, G. Wang, and Z. Liu, "Text2light: Zero-shot text-driven hdr panorama generation," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–16, 2022.
- [16] S. Zhou, Z. Fan, D. Xu, H. Chang, P. Chari, T. Bharadwaj, S. You, Z. Wang, and A. Kadambi, "Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting," *arXiv preprint arXiv:2404.06903*, 2024.
- [17] H. Zhou, X. Cheng, W. Yu, Y. Tian, and L. Yuan, "Holodreamer: Holistic 3d panoramic world generation from text descriptions," *arXiv preprint arXiv:2407.15187*, 2024.
- [18] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "SDXL: improving latent diffusion models for high-resolution image synthesis," *CoRR*, 2023.
- [19] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139:1–139:14, 2023.
- [20] R. Po, W. Yifan, V. Golyanik, K. Aberman, J. T. Barron, A. H. Bermano, E. R. Chan, T. Dekel, A. Holynski, A. Kanazawa *et al.*, "State of the art on diffusion models for visual computing," *arXiv preprint arXiv:2310.07204*, 2023.
- [21] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5589–5599.
- [22] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.
- [23] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, "Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering," in *European conference on computer vision*. Springer, 2022, pp. 106–122.
- [24] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.
- [25] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [26] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision*. Springer, 2022, pp. 333–350.
- [27] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.
- [28] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.
- [29] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5438–5448.
- [30] S. Cai, E. R. Chan, S. Peng, M. Shahbazi, A. Obukhov, L. V. Gool, and G. Wetzstein, "Diffdreamer: Towards consistent unsupervised single-view scene extrapolation with conditional diffusion models," in *ICCV*. IEEE, 2023, pp. 2139–2150.
- [31] D. Cohen-Bar, E. Richardson, G. Metzer, R. Giryes, and D. Cohen-Or, "Set-the-scene: Global-local training for generating controllable nerf scenes," 2023.
- [32] A. Vilesov, P. Chari, and A. Kadambi, "Cg3d: Compositional generation for text-to-3d via gaussian splatting," 2023.
- [33] O. Bar-Tal, L. Yariv, Y. Lipman, and T. Dekel, "Multidiffusion: Fusing diffusion paths for controlled image generation," 2023.
- [34] Y. Lee, K. Kim, H. Kim, and M. Sung, "Syncdiffusion: Coherent montage via synchronized joint diffusions," 2023.
- [35] J. Wang, Z. Chen, J. Ling, R. Xie, and L. Song, "360-degree panorama generation from few unregistered nfov images," in *Proceedings of the 31st ACM International Conference on Multimedia*. ACM, Oct. 2023. [Online]. Available: <http://dx.doi.org/10.1145/3581783.3612508>
- [36] J. Li and M. Bansal, "Panogen: Text-conditioned panoramic environment generation for vision-and-language navigation," 2023.
- [37] T. Wu, C. Zheng, and T.-J. Cham, "Panodiffusion: 360-degree panorama outpainting via diffusion," 2024.
- [38] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017.
- [39] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," 2023.
- [40] J. Li, H. Tan, K. Zhang, Z. Xu, F. Luan, Y. Xu, Y. Hong, K. Sunkavalli, G. Shakhnarovich, and S. Bi, "Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model," *CoRR*, 2023.
- [41] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [42] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lemitsky, "Resolution-robust large mask inpainting with fourier convolutions," 2021.
- [43] M. Feng, J. Liu, M. Cui, and X. Xie, "Diffusion360: Seamless 360 degree panoramic image generation based on diffusion models," 2023.
- [44] T. Hara and T. Harada, "Spherical image generation from a single normal field of view image by considering scene symmetry," 2020.
- [45] J. Jain, J. Li, M. T. Chiu, A. Hassani, N. Orlov, and H. Shi, "One-former: One transformer to rule universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2989–2998.
- [46] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 633–641.
- [47] L. Zhang and M. Agrawala, "Transparent image layer diffusion using latent transparency," 2024. [Online]. Available: <https://arxiv.org/abs/2402.17113>
- [48] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.
- [49] T. Yang, P. Ren, X. Xie, and L. Zhang, "Pixel-aware stable diffusion for realistic image super-resolution and personalized stylization," *arXiv preprint arXiv:2308.14469*, 2023.
- [50] M. Rey-Area, M. Yuan, and C. Richardt, "360monodepth: High-resolution 360deg monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3762–3772.
- [51] Z. Liu, H. Ouyang, Q. Wang, K. L. Cheng, J. Xiao, K. Zhu, N. Xue, Y. Liu, Y. Shen, and Y. Cao, "Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior," *arXiv preprint arXiv:2404.11613*, 2024.
- [52] runwayml, "stable-diffusion-inpainting model card," 2022. [Online]. Available: <https://huggingface.co/runwayml/stable-diffusion-inpainting>
- [53] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," 2021.
- [54] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," 2023.
- [55] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba, "Recognizing scene viewpoint using panoramic place representation," in *2012*

- IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2012, pp. 2695–2702.
- [56] G. B. M. Stan, D. Wofk, S. Fox, A. Redden, W. Saxton, J. Yu, E. Afalo, S.-Y. Tseng, F. Nonato, M. Muller, and V. Lal, “Ldm3d: Latent diffusion model for 3d,” 2023.
- [57] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.
- [58] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by two time-scale update rule converge to a local nash equilibrium,” in *NIPS*, 2017, pp. 6626–6637.
- [59] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi, “Clip-score: A reference-free evaluation metric for image captioning,” in *EMNLP (1)*. Association for Computational Linguistics, 2021, pp. 7514–7528.
- [60] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.
- [61] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a “completely blind” image quality analyzer,” *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209–212, 2012.
- [62] A. Mittal, A. K. Moorthy, and A. C. Bovik, “No-reference image quality assessment in the spatial domain,” *IEEE Transactions on image processing*, vol. 21, no. 12, pp. 4695–4708, 2012.
- [63] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
- [64] L. Jin, J. Zhang, Y. Hold-Geoffroy, O. Wang, K. Matzen, M. Stich, and D. F. Fouhey, “Perspective fields for single image camera calibration,” 2023. [Online]. Available: <https://arxiv.org/abs/2212.03239>



**Tong Wu** received her Ph.D. degree from the Multimedia Laboratory at the Chinese University of Hong Kong in 2024. Before that, she received her Bachelor’s degree from Tsinghua University, Beijing, in 2020. Her research focuses on 3D reconstruction and Generation. She has served as a regular reviewer for CVPR, ICCV, ECCV, NeurIPS, ICLR and AAAI.



**Xixuan Li** received the B.Sc. degree from Nanjing University, Nanjing, China, in 2019, and the M. Sc. degree from Nanjing University, Nanjing, China, in 2022. She is currently pursuing a Ph.D degree in the Chinese University of Hong Kong, Hong Kong. Her research area is 3D vision, especially 3D scene reconstruction and generation, with a recent focus on neural rendering for large-scale scenes. She has served as a reviewer for CVPR, ICCV, ECCV, NeurIPS, IJCV.



**Gordon Wetzstein** (Senior Member, IEEE) is currently an associate professor of EE and, by courtesy, of CS at Stanford University, Stanford, California, leading the Stanford Computational Imaging Lab and a faculty co-director of the Stanford Center for Image Systems Engineering.



**Shuai Yang** received the B.Sc. degree from Tongji University, Shanghai, China, in 2023. He is currently pursuing a Ph.D. degree at School of Electronic Information and Electrical Engineering in Shanghai Jiao Tong University. His research interests lie in the field of 3D Vision, particularly content creation of the 3D scene and object.



**Ziwei Liu** (Member, IEEE) received the PhD degree from the Chinese University of Hong Kong. He is currently an assistant professor with Nanyang Technological University, Singapore. Previously, he was a senior research fellow with the Chinese University of Hong Kong and a post-doctoral researcher with the University of California, Berkeley. His research interests include revolves around computer vision, machine learning, and computer graphics. He has published extensively on top-tier conferences and journals in relevant fields, including CVPR, ICCV, ECCV, NeurIPS, ICLR, ICML, TPAMI, TOG and Nature - Machine Intelligence. He is the recipient of Microsoft Young Fellowship, Hong Kong PhD Fellowship, ICCV Young Researcher Award and HKSTP Best Paper Award. He also serves as an Area Chair of ICCV, NeurIPS, and ICLR.



**Jing Tan** received the B.Sc. degree and M. Sc. degree from Nanjing University, Nanjing, China, in 2020 and 2023. She is currently pursuing a Ph.D. degree in the Multimedia Laboratory, the Chinese University of Hong Kong. Her research focuses on perceiving and recreating the world system from understanding the foreground events to the creation of the 3D scene surroundings. She has served as a reviewer for CVPR, ICCV, ECCV, NeurIPS, ICLR and IJCV.



**Dahua Lin** received the BEng degree from the University of Science and Technology of China, Hefei, China, in 2004, the MPhil degree from the Chinese University of Hong Kong, Hong Kong, in 2006, and the PhD degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2012. From 2012 to 2014, he was a research assistant professor with Toyota Technological Institute at Chicago, Chicago, IL, USA. He is currently an associate professor with the Department of Information Engineering, Chinese University of Hong Kong (CUHK), and the leading scientist with Shanghai AI Laboratory.



**Mengchen Zhang** received her B.Sc. degree from Nanjing University, Nanjing, China, in 2023. She is currently pursuing a Ph.D. degree at Zhejiang University, Zhejiang, China. Her research interests lie in the field of 3D Vision, particularly 6D pose estimation and 3D generation.