

# GEN3D: GENERATING DOMAIN-FREE 3D SCENES FROM A SINGLE IMAGE

*Yuxin Zhang<sup>1,2</sup>, Ziyu Lu<sup>1,2</sup>, Hongbo Duan<sup>1,2</sup>, Keyu Fan<sup>1</sup>  
 Pengting Luo<sup>2</sup>, Peiyu Zhuang<sup>2</sup>, Mengyu Yang<sup>1,2</sup>, Houde Liu<sup>1,\*</sup>*

<sup>1</sup> Shenzhen International Graduate School, Tsinghua University, Shenzhen, China

<sup>2</sup> Central Media Technology Institute, Huawei, Shenzhen, China

## ABSTRACT

Despite recent advancements in neural 3D reconstruction, the dependence on dense multi-view captures restricts their broader applicability. Additionally, 3D scene generation is vital for advancing embodied AI and world models, which depend on diverse, high-quality scenes for learning and evaluation. In this work, we propose Gen3d, a novel method for generation of high-quality, wide-scope, and generic 3D scenes from a single image. After the initial point cloud is created by lifting the RGBD image, Gen3d maintains and expands its world model. The 3D scene is finalized through optimizing a Gaussian splatting representation. Extensive experiments on diverse datasets demonstrate the strong generalization capability and superior performance of our method in generating a world model and Synthesizing high-fidelity and consistent novel views.

**Index Terms**— Novel View Synthesis, 3D Scene Generation, world model

## 1. INTRODUCTION

With the advent of commercial mixed reality platforms and the rapid innovations in 3D graphics technology, high-quality 3D scene generation has become one of the most important problem in computer vision for applications such as immersive media, robotics, autonomous driving, and embodied AI. Previous approaches mainly include Geometric reconstruction and video generation methods. Geometric reconstruction methods suffer from severe artifacts at viewpoints unseen in input views, while video generation methods lack good geometric consistency in their outputs.

In this work, we propose a pipeline called Gen3d. To address the artifact generation issue of geometric reconstruction methods, Gen3d adopts a hierarchical strategy and leverages Stable Diffusion [1] and 3D Gaussian splatting [2] to generate a wide variety of high-quality 3D scenes with geometric consistency from multiple types of inputs, such as text, RGB images, and RGBD images.

Gen3d first decomposes the input image into foreground and background layers via depth maps (input or from depth models). The background is inpainted via Lama-ControlNet inpainting[3] guided image inpainting, and initial point clouds are generated for both layers. The camera then moves along a predefined trajectory. At each pose, visible parts of the point cloud are projected onto the new view, and a Stable Diffusion-based inpainting network synthesizes a complete image from this partial projection. The result, along with its depth map, is lifted back to 3D space, with point positions refined via an alignment algorithm. Finally, the aggregated point cloud initializes a 3D Gaussian splatting representation, which resolves depth

inconsistencies and enables high-quality novel view synthesis beyond traditional point-based methods.

In summary, our contributions are as follows.

- We introduce **Gen3d**, a **domain-free, high-quality 3D scene generation method**, achieving better domain generalization in 3D scene generation by leveraging the power of Stable Diffusion, depth estimation, and explicit 3D representation.
- To generate multi-view images, we **adopt a hierarchical strategy** that first constructs a point cloud as a geometric prior for each novel view to generate novel view images. The generated images are then seamlessly integrated to form a coherent 3D scene.
- Our model provides users with the ability to **create 3D scenes in various ways by supporting different input types**, such as text, RGB, and RGBD.

## 2. RELATED WORK

**Novel view generation.** Given a single image, early methods infer 3D scene representations and use them to render novel views. These representations include point clouds [4, 5], multi-plane images [6, 7], depth maps[8], and meshes [9]. Despite enabling fast rendering, these representations limit camera movement due to their finite spatial extent. In addition, they only supported generating views within small viewpoint changes w.r.t. the input image, as they only built single static scene representations that do not go beyond the input image. Gen3d focus on a generative task to support creating many connected scenes rather than a single one.

**3D scene generation.** Recently, advancements in scene generation have predominantly centered on modeling isolated, local 3D environments. A significant number of these efforts explicitly target indoor settings [10, 11], while another line of research concentrates predominantly on outdoor scenarios [12]. This thematic specialization limits the generalizability of such approaches across diverse environmental contexts.

**Video generation.** Recent improvements in video generation [13] have led to interest in whether these models can also be used as scene generators. Several works have attempted to add camera control, allowing a user to “move” through the scene [14]. While these are promising, they currently do not guarantee 3D consistency.

**Fast 3D scene representations.** 3D Gaussian Splatting(3DGS) excels in high-fidelity real-time rendering by explicitly modeling scenes as anisotropic 3D Gaussians—each defined by position, covariance, opacity, and spherical harmonics coefficients. Unlike implicit volumetric methods requiring costly ray marching, 3DGS enables efficient GPU-based rasterization, achieving real-time high-resolution view synthesis. Its adaptive optimization dynamically adjusts Gaussians to capture fine details. Leveraging these advantages (efficiency, explicit control, and rendering quality), we adopt

\*Corresponding author: liu.hd@sz.tsinghua.edu.cn

3DGS as our core representation for sparse-input 3D reconstruction.

### 3. METHODOLOGY

Given an input image or text prompt, our purpose is to generate realistic and high-quality 3D scenes that are conditioned on this input. In scenarios where only a text prompt is provided, Gen3d is capable of producing scenes that are semantically related to the given text. Furthermore, Gen3d excels in creating specified scenes based on text prompts while preserving the stylistic elements of the input image.

#### 3.1. Single-view Layer Generation

To decompose a scene into layered representations from a single view, we propose a hybrid approach combining depth-prior-guided segmentation with diffusion-based inpainting. As illustrated in Fig. 1, our framework first generates foreground object masks  $\mathcal{M}_{fg}$  and background masks  $\mathcal{M}_{bg}$  using depth-aware Segment Anything Model (SAM)[15], then reconstructs occluded regions in the background layer through text-conditioned inpainting.

**Depth-guided Mask Generation.** Given an input image  $I \in \mathbb{R}^{H \times W \times 3}$ , we estimate its depth map  $D \in \mathbb{R}^{H \times W}$  using Moge2[16]. The SAM model produces  $N$  candidate masks  $\{\mathbf{M}_i\}_{i=1}^N$  with associated confidence scores  $\{s_i\}_{i=1}^N$ . We filter these masks through:

$$\mathcal{V} = \{\mathbf{M}_i | s_i > \tau_{iou} \wedge \text{area}(\mathbf{M}_i) \in [A_{min}, A_{max}]\} \quad (1)$$

where  $\tau_{iou} = 0.85$  and  $A_{min/max}$  are set as 0.5% / 60% of image area. Each valid mask is then classified by depth consistency:

$$\mathbf{M}_{fg} = \bigcup_{\mathbf{M}_i \in \mathcal{V}} \{\mathbf{M}_i \mid \text{median}(D|_{\mathbf{M}_i}) < \theta_d\} \quad (2)$$

with  $\theta_d$  dynamically set as the 35% of  $D$ . The background mask is derived as  $\mathbf{M}_{bg} = \mathbf{1} - \mathbf{M}_{fg}$ .

**Occlusion-aware Inpainting.** For regions  $\Omega = \mathbf{M}_{fg} \cap \{D > \theta_d\}$ , we employ Lama-ControlNet with two key modifications:

(1) Depth-conditioned attention: The cross-attention layers in Stable Diffusion are modulated by normalized depth values  $\tilde{D} = (D - D_{min})/(D_{max} - D_{min})$ .

(2) Textual prompting: Background inpainting uses the template “high-resolution [scene category] background with [dominant colors] colors, photorealistic, 8K” where bracketed terms are auto-filled by CLIP[17].

#### 3.2. world generation

To generate a multi-view consistent 3D point cloud, we first create the initial point cloud. Subsequently, while moving the camera, we aggregate both the new points and initial points by moving back and forth between the 3D space and the camera plane, with the reconstruction results situated in a unified world coordinate system. The overall process of point cloud construction is illustrated in Figure 1.

**Initialization.** A point cloud generation starts from lifting the pixels of the initial image. If the user provides a text prompt as input, a latent diffusion model is utilized to generate an image relevant to the given text, and the metric depth map is estimated using Moge2. We denote the generated or received RGB image and the corresponding depth map as  $\mathbf{I}_0 \in \mathbb{R}^{3 \times H \times W}$  and  $\mathbf{D}_0 \in \mathbb{R}^{H \times W}$ , respectively, where  $H$  and  $W$  represent the height and the width of the image. The camera intrinsic matrix and the extrinsic matrix of  $\mathbf{I}_0$  are denoted as  $\mathbf{K}$  and  $\mathbf{P}_0$ , respectively. For the case where  $\mathbf{I}_0$  and  $\mathbf{D}_0$  are generated from the diffusion model, we set the values of  $\mathbf{K}$  and  $\mathbf{P}_0$  by convention according to the size of the image.

From the RGBD image  $[\mathbf{I}_0, \mathbf{D}_0]$ , we lift the pixels into the 3D space, where the lifted pixels form a point cloud. The generated initial point cloud using the first image is defined as  $\mathcal{P}_0$ :

$$\mathcal{P}_0 = f_{2d \rightarrow 3d}([\mathbf{I}_0, \mathbf{D}_0], \mathbf{K}, \mathbf{P}_0), \quad (3)$$

where  $f_{2d \rightarrow 3d}$  is the function that lifts pixels from the RGBD image  $[\mathbf{I}, \mathbf{D}]$  to the point cloud.

**Point Cloud Augmentation and aggregation.** We sequentially attach points to the original point cloud to construct a large-scale 3D scene. Specifically, we define a counterclockwise rotating camera trajectory trajectory of length  $N$ , where  $\mathbf{P}_i$  denotes the camera position and pose at the  $i$ -th index. At each step, we inpaint and lift the missing pixels. We leverage the representational power of Stable Diffusion for the image inpainting task. Specifically, At step  $i$ , we first move and rotate the camera from the previous position  $\mathbf{P}_{i-1}$  to  $\mathbf{P}_i$ . The coordinate system is transformed from world coordinates to the current camera coordinates, followed by projection onto the camera plane using the intrinsic matrix  $\mathbf{K}$  and the extrinsic matrix  $\mathbf{P}_i$ . We denote the projected image at camera  $\mathbf{P}_i$  as  $\hat{\mathbf{I}}_i$ . Due to the change in camera position and pose, certain regions in  $\hat{\mathbf{I}}_i$  cannot be filled from the existing point cloud. We define a binary mask  $\mathbf{M}_i$  to indicate the filled regions:  $\mathbf{M}_i$  equals 1 if the corresponding pixel is filled by existing points, and 0 otherwise. The Stable Diffusion inpainting model ( $S$ ) is employed to generate a realistic image  $\mathbf{I}_i$  from the incomplete image  $\hat{\mathbf{I}}_i$  and the mask  $\mathbf{M}_i$ . The corresponding depth map  $\hat{\mathbf{D}}_i$  is estimated using Moge2.

Note that the monocular depth estimation model provides metric depth values. We then lift pixels to 3D space using the inpainted image  $\mathbf{I}_i$  and its corresponding depth map  $\mathbf{D}_i$ . To save memory consumption and represent efficiently, only pixels in the inpainted regions ( $\mathbf{M}_i = 0$ ) are lifted.

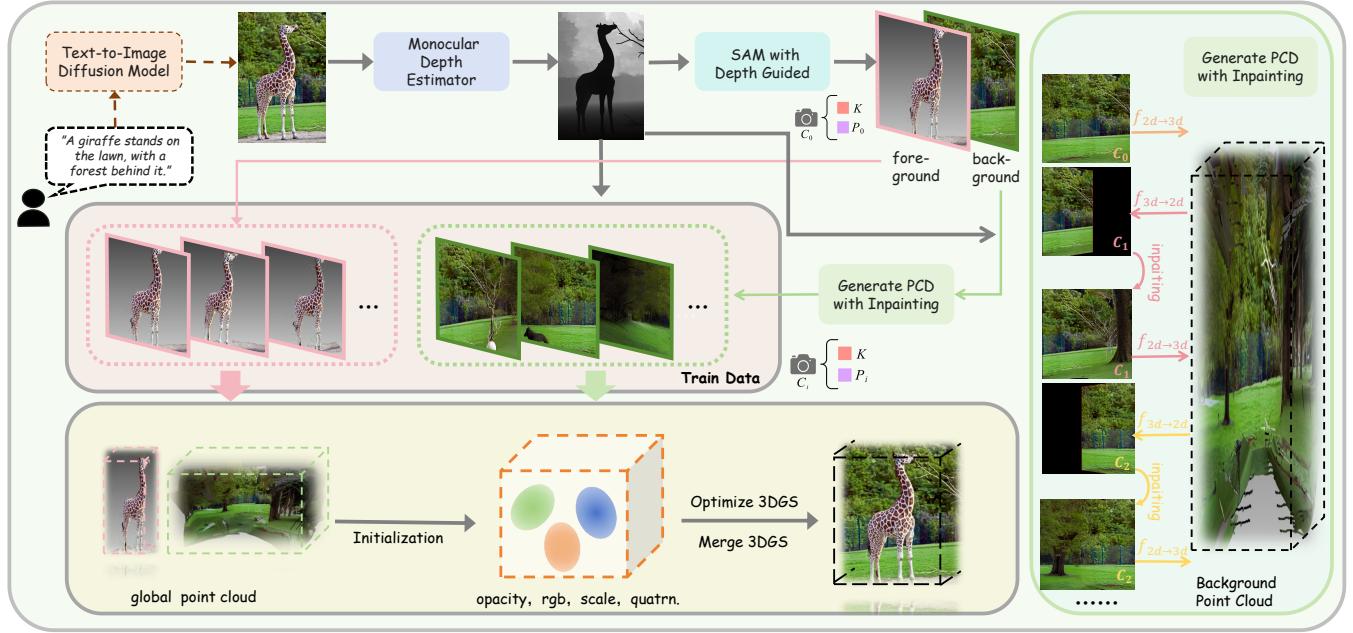
Compared to approaches that train generative models to simultaneously produce both RGB and depth maps, such as RGBD2[18] employing off-the-shelf depth estimation methods yields more accurate and generalizable depth maps, as these models are trained on large and diverse datasets. However, since  $\mathbf{D}_0, \mathbf{D}_1, \dots, \mathbf{D}_{i-1}$  are not considered when estimating  $\mathbf{D}_i$ , an inconsistency arises when integrating new points  $\hat{\mathcal{P}}_i$ .

To address this issue, we adjust the points in  $\hat{\mathcal{P}}_i$  within the 3D space to ensure smooth integration between the existing point cloud  $\mathcal{P}_{i-1}$  and the new points  $\hat{\mathcal{P}}_i$ . Specifically, we extract the region where the mask boundary changes ( $|\nabla \mathbf{M}_i| > 0$ ) to identify corresponding points in both  $\mathcal{P}_{i-1}$  and  $\hat{\mathcal{P}}_i$ . We then compute a displacement vector from  $\hat{\mathcal{P}}_i$  to  $\mathcal{P}_{i-1}$ . However, naively moving points may distort the geometry of the lifted point cloud and cause misalignment with the inpainted image. To mitigate this, we impose constraints on point movement and employ an interpolation algorithm to preserve the overall structure.

First, we constrain each point in  $\hat{\mathcal{P}}_i$  to move along the ray originating from the camera center to its corresponding pixel. We locate the closest point in  $\mathcal{P}_{i-1}$  along this ray and quantify the required depth adjustment. This constraint ensures that the visual content of the RGB image  $\mathbf{I}_i$  remains consistent despite 3D point adjustments. Second, we assume depth values remain unchanged on the opposite side of the mask boundary. For points without ground-truth correspondences (i.e., where  $\mathbf{M}_i = 0$ ), we compute depth changes via linear interpolation. Smooth interpolation alleviates artifacts caused by abrupt movements.

The aligned point cloud is combined with the original as follows:

$$\mathcal{P}_i = \mathcal{P}_{i-1} \cup \mathcal{W}(\hat{\mathcal{P}}_i), \quad (4)$$



**Fig. 1. Gen3d pipeline.** Initially, the input 2D image is segmented into two distinct components: the foreground objects and the background. We adopt methodologies such as the Stable Diffusion model and monocular depth estimation to enhance point cloud coverage and facilitate the construction of larger-scale scenes. Subsequently, we employ the point cloud alongside the reprojected images to optimize a set of Gaussian splats, further refining the resulting 3D scene.

where  $\mathcal{W}$  denotes the movement and interpolation operation. This process is repeated  $N$  times to construct the final point cloud  $\mathcal{P}_N$ . Through reprojection,  $\mathcal{P}_N$  delivers high-quality, multi-view consistent images. The overall procedure for constructing  $\mathcal{P}_N$  from  $[\mathbf{I}_0, \mathbf{D}_0], \mathbf{K}$ .

### 3.3. Rendering with gaussian splatting

After the point cloud is constructed, we train a 3D Gaussian Splatting model using the point cloud and the projected images. The centers of the Gaussian splats are initialized from the input point cloud, while the volume and position of each point are optimized under the supervision of the ground truth projected images. The loss function is constructed as a weighted combination of L1 loss and SSIM loss.

Initializing with  $\mathcal{P}_N$  accelerates network convergence and encourages the model to focus on reconstructing fine-grained details. For training, we use an additional set of  $M$  images alongside the original  $(N+1)$  images employed in point cloud generation, as the initial set alone is insufficient for producing plausible results. These  $M$  new images and their corresponding masks are generated by re-projecting  $\mathcal{P}_N$  under a new camera sequence  $\mathbf{P}_{N+1}, \dots, \mathbf{P}_{N+M}$ :

$$\mathbf{I}_i, \mathbf{M}_i = f_{3d \rightarrow 2d}(\mathcal{P}_N, \mathbf{K}, \mathbf{P}_i), \quad i = N+1, \dots, N+M. \quad (5)$$

Note that we do not perform inpainting on  $\mathbf{I}_i$  during the optimization of the Gaussian splats. Instead, when computing the loss function, we only consider the valid image regions where the mask value is 1. This prevents the model from learning artifacts in the reprojected images. Since each point is represented as a Gaussian distribution, missing pixels are naturally filled during rendering, and the resulting rasterized image becomes plausible after training.

## 4. EXPERIMENTS

### 4.1. Implementation details.

The modules we used to construct Gen3d can be either trained using manual design or brought from off-the-shelf models. We use pretrained large-scale off-the-shelf models to compose the whole network to maximize the generalization capability of the network. Specifically, we use the same text prompt input for the Stable Diffusion if the first image is generated from the text. If the input format is a RGB(D) image without text, we use LAVIS[24] to generate the caption according to the image and place it in the diffusion inpainting model to generate consistent content. For the camera trajectory that we use to construct the point cloud ( $\{\mathbf{P}_i\}_{i=0}^N$ ), we create several types of camera trajectory presets in advance, and different types of trajectories were used for different tasks.

### 4.2. World Generation

We test Gen3d on WorldScore [25] static benchmark on world generation. WorldScore consists of 2,000 static test examples that span diverse worlds. The metrics evaluate the controllability and quality of video generation. Specifically, we use “Camera Control”, “Object Control”, and “Content Alignment” to judge how the model adhere to viewpoint instructions and text prompts. We use “3D Consistency”, “Photometric Consistency”, “Style Consistency”, and “Subjective Quality” to evaluate the consistency and quality of generated content. Finally, an average score is presented to show the overall performance. We compare six 3D generation methods in the existing benchmark. The scores are reported in Table 1. Gen3d achieves the highest score on this benchmark. The score shows that Gen3d has competitive performance on camera control and 3D consistency, compared with 3D Models. Our Photometric consistency score is the

**Table 1. WorldScore Benchmark Comparison.** Abbreviations: Ctrl=Controllability, Align=Alignment, Consist=Consistency, Photo=Photometric. The top three rankings from highest to lowest are marked with .

| Models              | Worldscore | Controllability |             |               | Quality    |               |               |                 |
|---------------------|------------|-----------------|-------------|---------------|------------|---------------|---------------|-----------------|
|                     |            | Camera Ctrl     | Object Ctrl | Content Align | 3D Consist | Photo Consist | Style Consist | Subjective Qual |
| SceneScape[19]      | 50.73      | 84.99           | 47.44       | 28.64         | 76.54      | 62.88         | 21.85         | 32.75           |
| Text2Room[11]       | 62.10      | 94.01           | 38.93       | 50.79         | 88.71      | 88.36         | 37.23         | 36.69           |
| LucidDreamer[20]    | 70.40      | 88.93           | 41.18       | 75.00         | 90.37      | 90.20         | 48.10         | 58.99           |
| WonderJourney[21]   | 63.75      | 84.60           | 37.10       | 35.54         | 80.60      | 79.03         | 62.82         | 66.56           |
| InvisibleStitch[22] | 61.12      | 93.20           | 36.51       | 29.53         | 88.51      | 89.19         | 32.37         | 58.50           |
| WonderWorld[23]     | 72.69      | 92.98           | 51.76       | 71.25         | 86.87      | 85.56         | 70.57         | 49.81           |
| Gen3d(ours)         | 75.05      | 99.77           | 41.00       | 66.73         | 91.79      | 91.11         | 77.01         | 57.92           |



**Fig. 2. Comparisons of multi-view generation results across different methods.** The images are sourced from the COCO dataset, the WorldScore Benchmark, and web-sourced images.

highest among all methods, further demonstrating the visual quality of our generated videos.

To comprehensively evaluate our Gen3d model for multi-view 3D generation, we conducted qualitative comparisons with top-performing baselines in our benchmark: WonderWorld and LucidDreamer. These models were selected for their state-of-the-art performance. WonderWorld is optimized only for open-landscape outdoor scenes and fails at indoor generation. Thus, indoor comparisons only include LucidDreamer and Gen3d.

As shown in Fig. 2, LucidDreamer has two critical flaws under large camera movements: noticeable geometric holes in complex occlusions, and inconsistent generation quality across views. In contrast, Gen3d via our layered strategy decomposes scenes into layers and optimizes their consistency independently, effectively synthesizing occluded regions and maintaining texture and shape uniformity. For outdoor scenes, WonderWorld, despite its strengths in clear-sky outdoor environments, still produces obvious floating artifacts during camera motion owing to the use of semantic hard-coding for re-

construction. Gen3d, however, maintains stable, artifact-free generation and accurate geometry in these challenging outdoor scenarios.

## 5. CONCLUSION

This study addresses the limitation of traditional neural 3D reconstruction—reliance on dense multi-view captures—by proposing Gen3d, a novel pipeline for high-quality, wide-scope 3D scene generation from a single input.

Gen3d includes: Stable Diffusion for photorealistic 2D synthesis, 3DGS for efficient 3D representation, and point cloud geometric guidance for cross-view consistency. It proceeds in steps: segmentation with initial point cloud generation, iterative point cloud expansion, diffusion-based novel view synthesis, 3DGS optimization to eliminate holes and enable real-time rendering.

Future work will focus on accelerating point cloud expansion and extending Gen3d to dynamic 3D scene generation.

## 6. REFERENCES

- [1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis, “3d gaussian splatting for real-time radiance field rendering.,” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.
- [2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10684–10695.
- [3] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky, “Resolution-robust large mask inpainting with fourier convolutions,” *arXiv preprint arXiv:2109.07161*, 2021.
- [4] Chris Rockwell, David F Fouhey, and Justin Johnson, “Pixel-synth: Generating a 3d-consistent experience from a single image,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14104–14113.
- [5] Jing Yu Koh, Harsh Agrawal, Dhruv Batra, Richard Tucker, Austin Waters, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson, “Simple and effective synthesis of indoor 3d scenes,” in *Proceedings of the AAAI conference on artificial intelligence*, 2023, vol. 37, pp. 1169–1178.
- [6] Richard Tucker and Noah Snavely, “Single-view view synthesis with multiplane images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 551–560.
- [7] Tewodros Amberbir Habtegebrial, Varun Jampani, Orazio Gallo, and Didier Stricker, “Generative view synthesis: From single-view semantics to novel-view images,” *Advances in neural information processing systems*, vol. 33, pp. 4745–4755, 2020.
- [8] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang, “3d photography using context-aware layered depth inpainting,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8028–8038.
- [9] Ronghang Hu, Nikhila Ravi, Alexander C Berg, and Deepak Pathak, “Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12528–12537.
- [10] Nantheera Anantrasirichai and David Bull, “Artificial intelligence in the creative industries: a review,” *Artificial intelligence review*, vol. 55, no. 1, pp. 589–656, 2022.
- [11] Lukas Höller, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner, “Text2room: Extracting textured 3d meshes from 2d text-to-image models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7909–7920.
- [12] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi, “Procthor: Large-scale embodied ai using procedural generation,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 5982–5994, 2022.
- [13] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [14] Zheng Zhu, Xiaofeng Wang, Wangbo Zhao, Chen Min, Niachen Deng, Min Dou, Yuqi Wang, Botian Shi, Kai Wang, Chi Zhang, et al., “Is sora a world simulator? a comprehensive survey on general world models and beyond,” *arXiv preprint arXiv:2405.03520*, 2024.
- [15] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädl, Chloe Rolland, Laura Gustafson, et al., “Sam 2: Segment anything in images and videos,” *arXiv preprint arXiv:2408.00714*, 2024.
- [16] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang, “Moge-2: Accurate monocular geometry with metric scale and sharp details,” *arXiv preprint arXiv:2507.02546*, 2025.
- [17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
- [18] Tackgeun You, Mijeong Kim, Jungtaek Kim, and Bohyung Han, “Generative neural fields by mixtures of neural implicit functions,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 20352–20370, 2023.
- [19] Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel, “Scenescape: Text-driven consistent scene generation,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 39897–39914, 2023.
- [20] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee, “Luciddreamer: Domain-free generation of 3d gaussian splatting scenes,” *arXiv preprint arXiv:2311.13384*, 2023.
- [21] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al., “Wonderjourney: Going from anywhere to everywhere,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6658–6667.
- [22] Paul Engstler, Andrea Vedaldi, Iro Laina, and Christian Rupprecht, “Invisible stitch: Generating smooth 3d scenes with depth inpainting,” in *2025 International Conference on 3D Vision (3DV)*. IEEE, 2025, pp. 457–468.
- [23] Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu, “Wonderworld: Interactive 3d scene generation from a single image,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5916–5926.
- [24] Dongxu Li, Junnan Li, Hung Le, Guangsen Wang, Silvio Savarese, and Steven C. H. Hoi, “Lavis: A library for language-vision intelligence,” 2022.
- [25] Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu, “Worldscore: A unified evaluation benchmark for world generation,” *arXiv preprint arXiv:2504.00983*, 2025.