Extend3D: Town-Scale 3D Generation

Seungwoo Yoon Jinmo Kim Jaesik Park*
Seoul National University
{dotori000, jmkim1012, jaesik.park}@snu.ac.kr

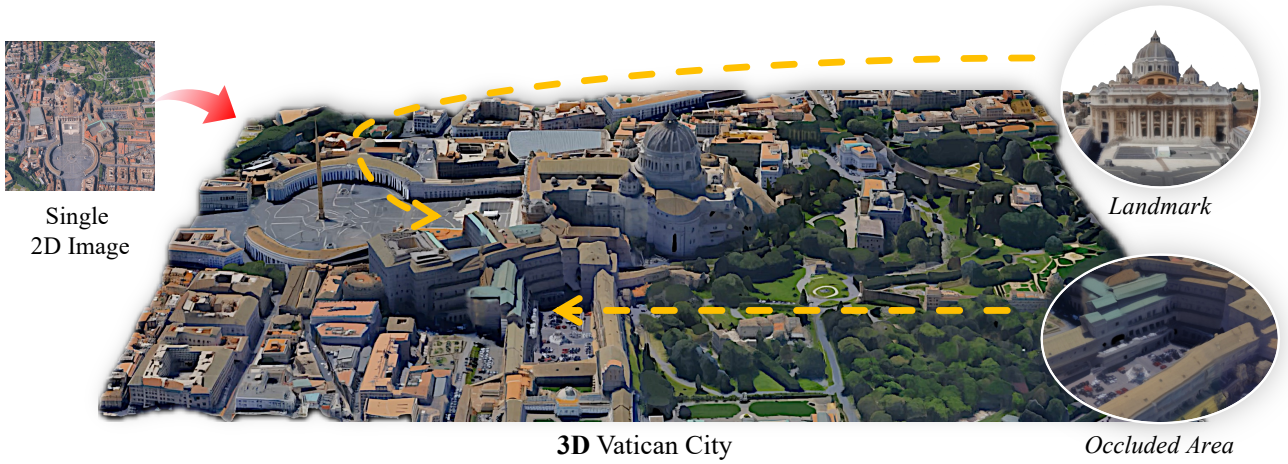

Figure 1. **The result of Extend3D.** We generated a large-scale 3D scene from an image of Vatican City captured from Google Earth [9].

Abstract

In this paper, we propose Extend3D, a training-free pipeline for 3D scene generation from a single image, built upon an object-centric 3D generative model. To overcome the limitations of fixed-size latent spaces in object-centric models for representing wide scenes, we extend the latent space in the x and y directions. Then, by dividing the extended latent space into overlapping patches, we apply the object-centric 3D generative model to each patch and couple them at each time step. Since patch-wise 3D generation with image conditioning requires strict spatial alignment between image and latent patches, we initialize the scene using a point cloud prior from a monocular depth estimator and iteratively refine occluded regions through SDEdit. We discovered that treating the incompleteness of 3D structure as noise during 3D refinement enables 3D completion via a concept, which we term under-noising. Furthermore, to address the sub-optimality of object-centric models for sub-scene generation, we optimize the extended latent during denoising, ensuring that the denoising trajectories remain consistent with the sub-scene dynamics. To this end, we introduce 3D-aware optimization objectives for improved geometric structure and texture fidelity. We demonstrate that our method yields better results than prior methods, as evidenced by human preference and quantitative experiments.

*Corresponding author.

1. Introduction

In the modern era, 3D scene assets are essential across fields such as game development, filmmaking, animation, simulation, and other areas of content production. Creating detailed 3D scenes requires substantial human effort and resources, even with the provided 3D assets. Therefore, a tailored generative model for 3D scenes would help reduce such costs and enhance productivity in industries.

Despite recent advances in 3D generative models, which have enabled the creation of production-ready high-quality 3D objects, generating large-scale 3D scenes remains challenging. One of the main challenges is that most current 3D datasets [3, 4, 8] consist of object-centric data, and lack cases with complex arrangements of multiple objects and a background. Consequently, previous data-centric approaches were unable to generate large general scenes.

Moreover, existing latent generative models [44, 47] represent 3D data with a fixed latent size, thereby limiting the level of detail of generated results. As the 3D scene grows in size, the output becomes blurry due to the limited latent dimensionality, resembling a low-resolution image. To adequately represent the scene's details, the latent size should be adapted to the scale of the result.

Therefore, research has been conducted to develop training-free pipelines for generating 3D scenes using object-centric models. Previous work has explored generating 3D scene blocks through an outpainting process [7, 49]. However, results from these approaches indicate that outpainting can degrade block consistency, particularly in large-scale scenes, making seams visible. Moreover, they rely entirely on the sub-scene generation capabilities of object-centric models, which are not sufficient.

In this paper, we introduce **Extend3D**, a novel training-free pipeline for generating 3D scenes from a single image. To achieve greater detail and scalability in large-scale 3D scene generation, we have expanded the latent space of a pre-trained 3D object generation model. Inspired by training-free, high-resolution image generation methods, such as those presented in recent works [1, 6, 10, 17, 20, 21, 42], we divide the extended latent space into overlapping patches and generate them simultaneously. Unlike previous outpainting methods, our approach automatically refines fine object details within the scene. This is possible because neighboring overlapping patches can influence each other, increasing the likelihood of accurately reconstructing their 3D representations.

However, there are challenges in 2D-3D spatial alignment and the object-centrality of pretrained models. To overcome this, we used the input image and the point cloud extracted from the monocular depth estimator [39] as priors to initialize and optimize the extended latents. We initialize the structure from the point cloud and refine the occluded regions using SDEdit [27] with *under-noising*. We optimize the latents at each time step using *3D-aware optimization objectives* to align the image and point cloud, ensuring that the denoising paths remain consistent with the sub-scene dynamics.

The qualitative results show that our method is scalable and generalizable. Through human preference and quantitative experiments, we demonstrate that our method outperforms state-of-the-art models in terms of geometry, appearance, and completeness, and is more faithful to the given image. Through an ablation study, we also demonstrate that overlapping patch-wise flow, initialization, and optimization are crucial for training-free 3D scene generation.

The main contributions of this paper are:

- We extend the latent space to integrate object-centric models into 3D scene generation, enabling a more generalizable and scalable generation pipeline.
- We introduce an overlapping patch-wise flow with image conditioning that captures local information and mitigates errors arising from object-centric models.
- We incorporate an iterative under-noised SDEdit process and 3D-aware optimization to complete occluded regions in the monocular depth point cloud and to overcome the deviation of object-centric models from scene dynamics.

2. Related Work

3D generative models. There have been numerous recent studies on generative models that can generate 3D objects conditioned on text or images. Currently, their main approach is the latent flow model [22, 33] applied to voxel-based or set-based latents.

Trellis [44] generates 3D Gaussians [13], radiance field [29], and mesh, using two steps of latent flow models where each generates a voxelized sparse structure and structured latents. Hunyuan3D [47] utilizes the latent flow model to generate shapes with set-based latents, as proposed in [45]. TripoSG [18] also uses the set-based latent representation of [45] to generate a mesh. These models have the limitation that they are trained with object-centric datasets. Moreover, structurally, current flow-based approaches suffer from the limitation that their latent size is predefined, so the output 3D can only have a confined range of details. We solve these problems by extending the latents to represent a large-scale scene.

To overcome the issues of object-centric models, some attempts have been made to train models using 3D scene datasets. BlockFusion [43] trains a diffusion model to generate cropped sub-scenes and generate the scene by extrapolation. PDD [23] trains a multi-scale diffusion model for coarse-to-fine scene generation. LT3SD [28] generates a 3D scene hierarchically with a latent tree representation. NuiScene [16] trains an autoregressive model with chunk VAE and vector sets. Nevertheless, since all of these methods are trained on limited datasets, they can generate 3D scenes with fewer categories than object-centric models. They also do not consider detailed model conditioning, such as image conditions, when designing hierarchical frameworks. Unlike them, our method can generate general 3D scenes with detailed image conditioning.

Training-free 3D scene generation. Recent advances in object-centric 3D generative models and the shortage of 3D scene datasets have led researchers to develop training-free 3D scene generation pipelines using these object-centric models.

SynCity [7] generates tiles of 3D sub-scenes sequentially with Trellis from a text using Flux inpainting [15]. Because SynCity attaches separate 3D sub-scenes, there are inconsistencies between tiles, and seams are visible. An image-to-3D scene generation pipeline, 3DTown [49], initializes

a scene with the point cloud from VGGT [37] and then completes it patch by patch using RePaint [24] and Trellis. Although 3DTown can generate 3D towns from images with high fidelity, it can only be used with restricted input due to the limitations of object-centric models (*e.g.*, vanishing floors). Also, regardless of initialization, some objects in the scene ignore certain input information, such as rotation. EvoScene [48] further leverages a video diffusion model [36] on 3DTown, but suffers from similar problems.

To address the problems of separate and sequential 3D sub-scene generation, we simultaneously generate 3D sub-scenes with interacting denoising paths. With small transitions between overlapping patches, the generation process can effectively capture local information and prevent geometrical errors through simultaneous generation. Also, unlike previous works that rely solely on sub-scene generation using an object-centric model, we optimize the latent representation at each step to prevent paths from transitioning from sub-scene to object dynamics.

Training-free high-resolution image generation. In the field of image generation, training-free high-resolution image generation has been widely researched and has led to massive discoveries on the dynamics of the scaled-up latent denoising process. The primary purpose of this area is to generate high-resolution images from pre-trained models trained on relatively low-resolution data.

MultiDiffusion [1] generates a high-resolution image from text with an extended 2D latent with overlapping patches. DemoFusion [6] solves the object repetition problem of Multidiffusion with two ideas: progressive upsampling and dilated sampling. Later research [20, 21], additionally refines dilated sampling.

When these methods are naively applied to extended 3D latent generation, however, we found that they fail to generate 3D scenes with high fidelity due to the unique dynamics of the model's image-condition, 3D, and object centrality. For instance, the floor vanishes, or poorly correlated patches lead to repeated objects. We therefore provide structure priors to generate a high-fidelity 3D scene.

Generation with priors. Several studies are trying to provide priors for pre-trained generative models for various purposes. SDEdit [27] is a representative method of image editing that can be applied to [11, 22, 33, 34]. SDEdit partially noise the original image, producing an edited image whose perturbed distributions retain the original image's style, meeting the intended image style. Readout Guidance [25] trains a small neural network to extract properties (*e.g.*, pose, depth, or edges) from the intermediate latent representation. Then, it computes the loss with respect to the property and provides a loss gradient as guidance, similar to the classifier guidance [5].

We apply SDEdit in our Extend3D to refine the initialized structure. Unlike image editing, we propose an under-noising technique designed for the 3D completion task. Also, instead of guidance, we optimize the intermediate latent with a loss explicitly designed for 3D scene generation, assuming that the priors have ground-truth knowledge of 3D structure and texture.

3. Preliminaries

3.1. Latent Flow Model for 3D Generation

A modern approach for high-quality 3D generative models is the latent flow model. They use voxelized latents of fixed size or set-based latents (*e.g.*, point clouds) within a confined region to represent 3D space. While our approach is not restricted to a specific generative model, it can be applied to general voxel-based latents or set-based latent flow models. We illustrate our idea using Trellis [44], which is one of the leading 3D generative models.

Trellis generates 3D representations with two steps of latent flow models given a condition $C_{\mathcal{I}}$ encoded from an image $\mathcal{I}$ by DINOv2 [31], and both steps are generalizable to flow models for voxelized or set-based latents. The first step of the model ***generates a sparse structure*** (SS) $\{\boldsymbol{p}_i\} \subset [M]^3$ (where $[M] := \{0, 1, ..., M-1\}$), which represents a set of occupied coordinates in a voxel grid. In sparse structure generation, low-resolution voxelized noise $\mathbf{Z}_1^{\mathrm{SS}} \in \mathbb{R}^{N\times N\times N}$ is denoised to $\mathbf{Z}_0^{\mathrm{SS}}$ with vector field $\boldsymbol{v}_{\mathrm{SS}}$, decoded with decoder $\mathcal{D}$, and activated voxel coordinates are collected as:

$$\mathbf{Z}_1^{\mathrm{SS}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad \frac{d}{dt}\mathbf{Z}_t^{\mathrm{SS}} = \boldsymbol{v}_{\mathrm{SS}}(\mathbf{Z}_t^{\mathrm{SS}}, C_{\mathcal{I}}, t), \tag{1}$$

$$\{\boldsymbol{p}_i\} = \{\boldsymbol{p} : \mathcal{D}(\mathbf{Z}_0^{SS})_{\boldsymbol{p}} > 0\}. \tag{2}$$

As the decoder is trained as a VAE, there is a trained encoder $\mathcal{E}$ that encodes the occupancy grid $O \in \mathbb{R}^{M\times M\times M}$ into a low-resolution latent representation. The second step of the model conducts ***denoising on a structured latent*** (SLAT), where a set-based latent feature is matched to a coordinate of sparse structure as:

$$\mathbf{Z}_t^{\mathrm{SLAT}} = \{(\boldsymbol{p}_i, \mathbf{z}_{i,t})\} \subset [M]^3 \times \mathbb{R}^l, \tag{3}$$

$$\mathbf{z}_{i,1} \overset{\mathrm{iid}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad \frac{d}{dt}\mathbf{Z}_t^{\mathrm{SLAT}} = \boldsymbol{v}_{\mathrm{SLAT}}(\mathbf{Z}_t^{\mathrm{SLAT}}, C_{\mathcal{I}}, t), \tag{4}$$

with invariant $\boldsymbol{p}_i$ and vector field $\boldsymbol{v}_{\mathrm{SLAT}}$. SLAT is then decoded to 3D representations such as 3D Gaussians, radiance field, or mesh by sparse decoders ($\mathcal{D}_{\mathrm{GS}}$, $\mathcal{D}_{\mathrm{NeRF}}$, and $\mathcal{D}_{\mathrm{mesh}}$), and usually. In this paper, we will use the notations $\mathbf{Z}_t$ that can refer to both $\mathbf{Z}_t^{\mathrm{SS}}$ and $\mathbf{Z}_t^{\mathrm{SLAT}}$, and $\boldsymbol{v}$ for $\boldsymbol{v}_{\mathrm{SS}}$ and $\boldsymbol{v}_{\mathrm{SLAT}}$ for simplicity.

3.2. SDEdit

We introduce SDEdit to refine the initialized structure, treating scene generation as a 3D sub-scene editing task. SDEdit

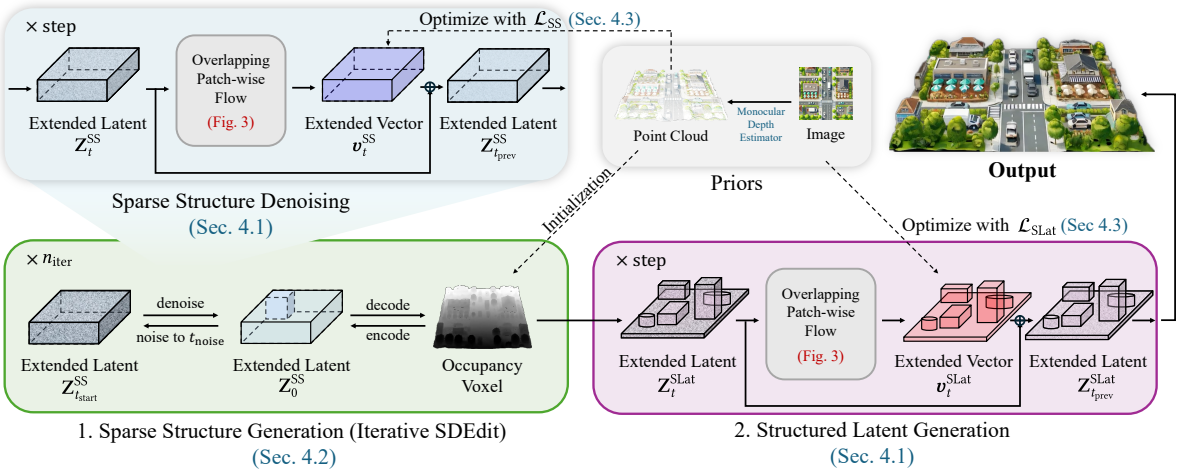

Figure 2. **An overall pipeline of our Extend3D.** Extend3D consists of two parts: sparse structure generation and structured latent generation. In the denoising part of both steps, an overlapping patch-wise flow was used (Sec. 4.1 and Fig. 3). In sparse structure generation, iterative SDEdit is used to initialize the structure (Sec. 4.2). Vector fields in both steps are optimized with priors (Sec. 4.3).

noises latent of a "guide" (*e.g.*, image to be edited) $\mathbf{Z}_0^{(g)}$ to $\mathbf{Z}_{t_{\text{start}}}$ and denoises it to $\mathbf{Z}_0$ to get the edited result. With the added noise, the perturbed distribution meets the intended distribution while preserving information in the guidance. Although SDEdit was designed for diffusion models [35], we can integrate it into flow models, with the following equations:

$$\mathbf{Z}_{t_{\text{start}}} = (1 - t_{\text{start}}) \cdot \mathbf{Z}_0^{(g)} + t_{\text{start}} \cdot \epsilon;\ \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad (5)$$

$$\frac{d}{dt}\mathbf{Z}_t = \boldsymbol{v}(\mathbf{Z}_t, C, t), \quad (6)$$

where C refers to the editing condition. When t_{start} increases, the denoising path gets longer, causing the effect of conditioning and generative models to be enlarged.

4. Method

Extend3D is a training-free pipeline that generates a 3D scene from a single scene image. To implement 3D scene generation, we extended the 3D latents of a pre-trained object-centric 3D generative model [44] to represent more detailed, larger 3D scenes. We extend the latents in the x and y coordinates, and a portion of the extended latent serves as a conventional latent for the pre-trained object-centric 3D generative model.

To handle extended latents, we divide them into overlapping patches, generated simultaneously via separate but coupled denoising paths conditioned on image patches (Sec. 4.1). Additionally, to address the underlying issues of the object-centric model (*e.g.*, vanishing floor, inability to generate sub-scenes, and randomly rotated objects) and to mitigate the problems associated with patch-wise generation (*e.g.*, repeated objects and seams between patches), we incorporate priors into the generation process. We first initialize the scene with a point cloud from a depth estimator and perform iterative under-noised SDEdit. This completes the occluded area and refines the scene while generating the structure (Sec. 4.2). We then optimize the scene at every time step using the point cloud and an image of the entire scene. We also propose a loss function that treats the point cloud as a prior for the voxel-based latent (Sec. 4.3). The overall pipeline is illustrated in Fig. 2 and Sec. A.1.

4.1. Overlapping Patch-wise Flow

In order to generate a detailed 3D structure and texture, we introduce an extended latent for sparse structure $\mathbf{Z}_t^{\text{SS}} \in \mathbb{R}^{aN \times bN \times N}$ and an extended SLAT $\mathbf{Z}_t^{\text{SLAT}} \subset [aM] \times [bM] \times [M] \times \mathbb{R}^l$ where $\mathbf{Z}_t$ can refer to both. Here, a and b are extension factors. (From here, we will use $\downarrow$ to notate non-extended latents or vectors.)

We divide these latents into overlapping patches with a division factor d. We refer to the (i, j)-th latent patches as $\phi_{i,j}^{\text{SS}}(\mathbf{Z}_t^{\text{SS}})$ and $\phi_{i,j}^{\text{SLAT}}(\mathbf{Z}_t^{\text{SLAT}})$. This process can be described as a N^3 or M^3-sized sliding window $\mathbb{W}$ moving with stride N/d or M/d to sample patches, illustrated as sampling in Fig. 3. The patches can be mapped back to their original positions by setting the values at the other positions to zero (zero padding), thereby coupling them, as illustrated in Fig. 3. We represent these inverse mappings as $(\phi_{i,j}^{\text{SS}})^{-1}$ and $(\phi_{i,j}^{\text{SLAT}})^{-1}$. We leave the rigorous definitions of the mappings in Sec. A.5.

We also patchify the image condition with $\psi_{i,j}$, which crops the image region to exactly match the (i, j)-th 3D patch (see details in Sec. A.2). Similar to MultiDiffusion, we get the vector field of the extended latents by merging the vector fields for each patch, where the overlapping regions are averaged across the patches, as illustrated in the left side of Fig. 3. The entire overlapping patch sampling,

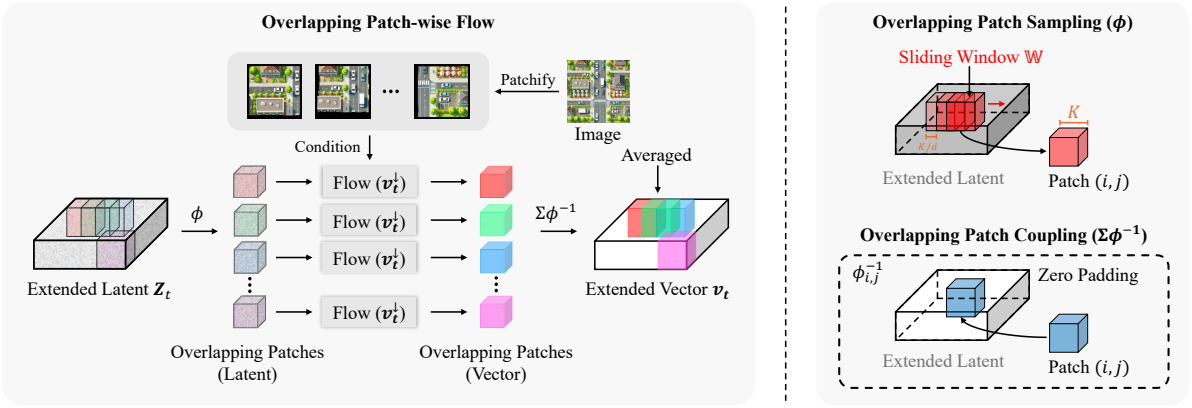

Figure 3. **Overlapping patch-wise flow.** The extended latent is divided into latent patches with the sliding window. We then obtain the patch vector for each latent patch and merge them into a single extended latent vector, thereby coupling the patches.

merging, and denoising process can be formulated as:

$$\boldsymbol{v}_{i,j}(\mathbf{Z}_t, \mathcal{I}, t) = \boldsymbol{v}^{\downarrow}\big(\phi_{i,j}(\mathbf{Z}_t), C_{\psi_{i,j}(\mathcal{I})}, t\big), \tag{7}$$

$$\boldsymbol{v}(\mathbf{Z}_t, \mathcal{I}, t) = \sum_{i,j} \phi_{i,j}^{-1}(\boldsymbol{v}_{i,j}) \oslash \sum_{i,j} \mathbf{1}_{\mathbb{W}_{i,j}}, \tag{8}$$

$$\frac{d}{dt}\mathbf{Z}_t = \boldsymbol{v}(\mathbf{Z}_t, \mathcal{I}, t), \tag{9}$$

where $\oslash$ is an element-wise division. Equation (7) can be calculated independently from the other patches and in parallel, but the dynamics of different patches, even far away, can be coupled by overlaps.

The advantage of divided but coupled dynamics is the ability to refine errors in other patches. By detecting slight movement of the sliding window, our method can identify local information from changes in the image and in latent features between patches. Additionally, because some objects are at the centers of patches, we can leverage the object-centric model more effectively. The beneficial effect of overlapping patch-wise flow can be found in Fig. 7 (A).

Noted in DemoFusion [6], AccDiffusion [21], and CutDiffusion [20], dilated sampling is crucial for generating a consistent global structure. We apply dilated sampling during the sparse structure generation phase and leave the details to Sec. A.3.

4.2. Initialize with Prior

When directly denoising sparse structure from pure Gaussian noise using Eq. (9), all patches fail to initialize each sub-scene due to the inherent limitation of the object-centric models. Moreover, the coarse structure is determined during the early denoising stage [42], before the patches are sufficiently coupled, so that the image condition and the 3D latent are not well spatially aligned. Consequently, the output becomes noisy, fragmented, and unstable as in Fig. 7 (B). This motivates the need for a robust structural prior at initialization.

Inspired by 3DTown [49], we initialize the scene structure with a point cloud $\mathbb{P}$ extracted from a monocular depth estimator. Specifically, we adopt MoGe-2 [38, 39] for our Extend3D. The predicted point cloud is voxelized into an occupancy grid $\boldsymbol{O}_0 \in \mathbb{R}^{aM \times bM \times M}$. Because the monocular depth estimator cannot infer the occluded regions, the resulting occupancy voxel grid contains empty areas that should be rectified using the pre-trained generative model. To address this, with an encoded voxel grid $\mathbf{Z}_0^{(g)} = \mathcal{E}(\boldsymbol{O}_0)$, our Extend3D performs SDEdit. Unlike standard SDEdit, which applied Eq. (5), we introduce *under-noising*:

$$\mathbf{Z}_{t_{\text{start}}} = (1 - t_{\text{noise}}) \cdot \mathbf{Z}_0^{(g)} + t_{\text{noise}} \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{10}$$

where $t_{\text{start}} > t_{\text{noise}}$, ensuring that the latent is denoised more aggressively than it was originally noised. By under-noising the guide structure, the pre-trained model may treat missing or occluded parts as additional noise, illustrated as the arrow ② in Fig. 4. Finally, the denoising process, represented as arrow ③, allows such areas to be filled. This is similar to adding high-frequency noise to enhance image detail in image super-resolution [12]. We empirically validate this choice in Sec. 5.4.

SDEdit can fill the unwanted empty areas. However, it often fails to fully complete the scene, leaving some holes.

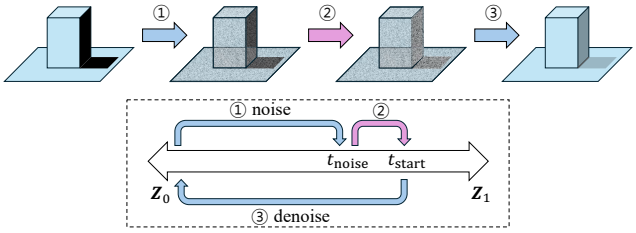

Figure 4. **Motivation of under-noising**. The blue arrows represent actual noising or denoising, while the purple arrow illustrates how the model is presumed to perceive.

To mitigate this, as a single SDEdit process partially refines the structure, we apply SDEdit iteratively as: $\boldsymbol{O}_n = \mathrm{SDEdit}(\boldsymbol{O}_{n-1})$, represented in Fig. 2. This process iteratively fills the occluded region of $\boldsymbol{O}_0$ and eventually completes the scene.

4.3. Optimize with Prior

During denoising, sub-scenes deviate from a scene-like structure toward an object-like structure due to the object-centric model's properties, leading to distortion or a vanishing floor, even with proper initialization. To prevent deviation and to align the denoising paths with the conditioning, we optimize the extended latents over time steps using the point cloud and the image. When solving Eq. (9) with the discrete ODE solver, instead of moving directly along $\boldsymbol{v}(\mathbf{Z}_t, \mathcal{I}, t)$, we use $\hat{\boldsymbol{v}}_t$, an optimized vector starting from $\boldsymbol{v}(\mathbf{Z}_t, \mathcal{I}, t)$. By optimization, we can leverage the pre-trained model for the occluded region while optimizing on the seen region, as in Readout Guidance [25]. In addition, optimizing the vector field can improve consistency across patches by simultaneously optimizing the entire scene, as in [17]. We introduce two optimization losses, one for sparse structure generation and one for structured latent generation, as explained in Sec. 3.1 and illustrated in Fig. 2.

In the ***sparse structure generation step***, we define:

$$\mathcal{L}_{\mathrm{SS}} = -\frac{1}{|\mathbb{P}|}\sum_{\boldsymbol{p}\in\mathbb{P}} \log \sigma((\mathcal{D}(\mathbf{Z}_t^{\mathrm{SS}} - t\cdot\hat{\boldsymbol{v}}_t))_{\boldsymbol{p}}), \tag{11}$$

where σ is a sigmoid function. The loss function is designed to enforce that the initialized voxels do not disappear during the denoising process, motivated by binary cross-entropy loss. It gives a positive signal on predicted voxels where points exist. Voxels with dense point clouds will have more weight in the loss. While this loss can be minimized by increasing the number of voxels, combined with the pre-trained model every time step, it merely prevents the desired voxels from disappearing, rather than creating undesired voxels. Moreover, for the same reason, it can smoothly connect the point cloud priors and generated voxels, not just by attaching two distinct voxel grids. With the $\mathcal{L}_{\mathrm{SS}}$, we optimize $\hat{\boldsymbol{v}}_t$ with Adam optimizer [14].

In the ***structured latent generation step***, we apply the extended rendering loss [40, 46] as follows:

$$\hat{\mathcal{I}} = \mathrm{Render}(\mathcal{D}_{\mathrm{GS}}(\mathbf{Z}_t^{\mathrm{SLAT}} - t\cdot\hat{\boldsymbol{v}}), \boldsymbol{P}), \tag{12}$$

$$\mathcal{L}_{\mathrm{SLAT}} = \mathrm{LPIPS}(\hat{\mathcal{I}}, \mathcal{I}) - \mathrm{SSIM}(\hat{\mathcal{I}}, \mathcal{I}), \tag{13}$$

where Render is a differentiable renderer (such as Gaussian splatting) and $\boldsymbol{P}$ is a camera parameter of an image viewpoint provided by the depth estimator. This optimizes the entire scene with an image in the original camera view. Because an object-centric model often loses details in scene textures, this optimization helps refine them. Also, it makes the seams invisible because the boundary is optimized at every time step, ensuring paths are more consistent with each other. With $\mathcal{L}_{\mathrm{SLAT}}$, we also optimize $\hat{\boldsymbol{v}}_t$ with Adam.

Please refer to Sec. A.1 for the algorithm details.

5. Experiments

5.1. Human Preference

Table 1. **Human preference win rate (%) of our method.**

versus.	Geometry	Faithfulness	Appearance	Completeness
Trellis [44]	50.0	66.4	67.1	62.1
Hunyuan3D [47]	73.6	75.7	75.0	75.0
EvoScene [48]	87.1	87.9	87.1	87.1

To score the visual aestheticity of 3D scenes, we conducted a human preference study. We compared our method with Trellis [44] and Hunyuan3D-2.1 [47], the current best open-sourced 3D generation models, and EvoScene, which is specifically designed for large-scene generation. Human annotators (10 participants) ranked the methods on four criteria: geometry, faithfulness, appearance, and completeness, given 14 images and 3D scenes. As a result (Tab. 1), our Extend3D outperformed previous methods in four criteria.

5.2. Quantitative Results

Table 2. **Quantitative results.**

	LPIPS ↓	SSIM ↑	PSNR ↑	CD ↓	F-score (0.05) ↑
Trellis [44]	0.650	0.239	10.0	0.0315	0.442
Hunyuan3D [47]	0.683	0.255	10.4	0.0192	0.567
EvoScene [48]	0.482	0.310	13.2	0.0188	0.498
Ours w/o $\mathcal{L}_{\mathrm{SLAT}}$	0.400	0.333	13.8	0.0078	0.708
Ours	0.240	0.611	20.4	0.0086	0.694

We render the 3D scene into the camera view of the input image using the camera parameter estimator [39], and obtain LPIPS, SSIM, and PSNR scores on 100 input images [9, 15, 19, 30, 41] spanning diverse wide scenes. As shown in Tab. 2, our method achieved the best scores across three metrics, indicating that it is most faithful to the input image in terms of structure and texture. Using 45 images and ground-truth mesh pairs from the UrbanScene3D dataset [19], we evaluated the geometric results using the Chamfer Distance (CD) and F-score with a threshold of 0.05. Table 2 shows that our method surpasses the results of the previous methods.

Table 3. **Comparison between 3D scene generation methods.**

	CLIP ↑	HPSv3 ↑	Intra-LPIPS ↓
SynCity [7]	0.251	3.254	0.631
Ours	**0.276**	**3.519**	**0.571**

We also compared our method with the state-of-the-art training-free 3D scene generation pipeline, SynCity.

Figure 5. **Qualitative result of our Extend3D.** Our 3D scene generation result (with $a = b = 2$) is compared to the results of state-of-the-art 3D generative models. While previous methods may not accurately represent the image or lose scene details, our method effectively expresses the image condition in 3D. The input image is generated using Flux.1 [dev] [15]. We provide additional results in Sec. A.7.

Figure 6. **Qualitative comparison with SynCity.** The results are generated from the text prompt, *medieval market*.

Since SynCity is text-conditioned, whereas ours is image-conditioned, we first generated scene images from keywords using ChatGPT [30] and then used Extend3D. We render the results of SynCity and Extend3D to get CLIP score [32], HPSv3 [26], and Intra-LPIPS [17]. Here, Intra-LPIPS refers to LPIPS between patches within a single scene, measuring the patch consistency. Table 3 shows that our Extend3D is superior to SynCity in text compatibility, quality, and patch-wise consistency.

5.3. Qualitative Results

Figure 1 shows the scalability of our method. Given a town-scale scene image, the extended latent can fully capture details, including landmarks and small buildings, and produce a 36× larger result than the original latent space. We present more examples of wide scenes in Fig. 17 and Fig. 18. Also, our Extend3D can represent general scenes. It can generate a town, a table of foods, a study scene, and an indoor room, illustrated in Fig. 5 and Sec. A.7. In diverse cases, our method outperformed previous 3D generative models. Also, compared with SynCity, our method can generate scenes without patch boundaries, as shown in Fig. 6 and Fig. 15. Moreover, because SynCity generates a scene in an outpainting approach, it cannot refine the unnatural edge of the water. Compared to EvoScene in Fig. 16, our results had less distorted geometry and more detailed textures.

5.4. Ablation Study

We conducted an ablation study on three proposed methods in our Extend3D and presented the results in Fig. 7. When we obtained the results for Fig. 7 (A) and (B), we did not optimize the latent to emphasize the structural difference.

Overlapping Patch-wise Flow. We claim that the coupled paths of patches can mutually rectify and effectively capture local information. To validate this argument, we compare results for varying division factors. As illustrated in Fig. 7 (A), $d = 2$ distorted local structure, while $d = 4$ did not. These results demonstrate that patch interactions correct each other, and the sliding window's stride enables the extended latent to capture finer details.

Initialize with Prior. In the first part of Fig. 7 (B), the results with and without initialization are compared. Without initialization (*i.e.*, $t_{\text{start}} = 1$), the structure is totally broken with important objects disappeared, buildings not in proper position, etc. We therefore conclude that proper initialization is essential for extended latents. In the second part, we compared three results with different t_{noise} and t_{start}. When $t_{\text{noise}} = t_{\text{start}}$ (usual SDEdit), the structure maintained the holes in the initial point cloud or was destroyed due to the t_{start} trade-off in SDEdit. However, with $t_{\text{noise}} < t_{\text{start}}$ (under-noising), the occluded region of the initial structure is completed naturally.

Optimize with Prior. Figure 7 (C) shows the ablation study

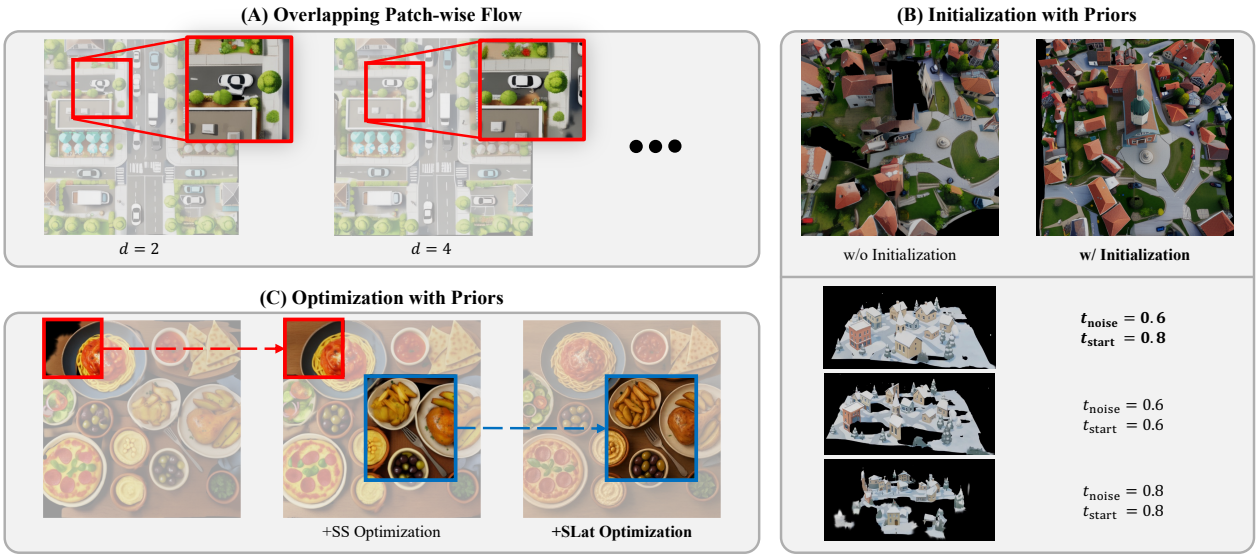

Figure 7. **Ablation study.** All the images, except for the ablation of under-noising, are taken from the input image camera viewpoint. We set $a = b = 2$ to generate the 3D scenes in this figure.

Table 4. **Ablation study for varying division factor d.**

	LPIPS ↓	SSIM ↑	PSNR ↑	CD ↓	F-score (0.05) ↑
$d = 2$	0.251	0.598	19.8	0.0088	0.692
$d = 4$	0.240	0.611	20.4	0.0086	0.694
$d = 8$	0.237	0.615	20.5	0.0079	0.699

Table 5. **Ablation study on prior initialization and optimization**

	LPIPS ↓	SSIM ↑	PSNR ↑	CD ↓	F-score (0.05) ↑
p.w. flow only	0.606	0.209	9.63	0.0348	0.261
+ initialize	0.425	0.312	13.0	0.0083	0.693
+ SS optim.	0.400	0.333	13.8	0.0078	0.708
+ SLAT optim.	0.240	0.611	20.4	0.0086	0.694

for optimization with priors. Starting from the base model, we sequentially added sparse structure and SLAT optimization. Without sparse structure optimization, the floor and parts of the objects vanished, as in Sec. A.7. Structured latent optimization could refine seams and distortion between patches compared to those without optimization. Furthermore, the overall quality of the scene's structure and texture improved (*e.g.*, the fork and chips in the figure).

In addition, as in Sec. 5.2, we quantitatively evaluated the generated scenes. Table 4 and Tab. 5 show the effectiveness of our proposed methods, consistent with the qualitative observation that increasing d and providing priors enhances the quality of the scenes. From the geometric results in Tab. 5, we find that initialization and SS optimization refine the geometry, whereas SLAT optimization sometimes degrades it. Since SLAT optimization usually enhances a 3D scene's texture, there is a trade-off between geometry and texture when optimizing the SLAT. Table 6, conducted without optimization and with $n_{\text{iter}} = 1$ as in the qualitative experiment, demonstrates that under-noising is the best choice of t_{start} and t_{noise} in 3D completion.

Table 6. **Ablation study on under-noising.**

t_{noise} / t_{start}	LPIPS ↓	SSIM ↑	PSNR ↑	CD ↓	F-score (0.05) ↑
0.6 / 0.6	0.388	0.324	13.4	0.0081	0.657
0.8 / 0.8	0.550	0.216	9.91	0.0292	0.378
0.8 / 0.6 (over-noise)	0.622	0.219	9.79	0.0518	0.249
0.6 / 0.8 (**under-noise**)	**0.387**	**0.327**	**13.5**	**0.0078**	**0.680**

6. Conclusion

We propose a training-free 3D scene generation pipeline, Extend3D. By the extended latent space of the pre-trained object-centric model, we enabled scalable 3D scene generation. We demonstrate that our method (overlapping patch-wise flow, initialization, and optimization) and its schemes (iterative SDEdit, under-noising, and 3D-aware optimization objectives) achieve notable improvements in image-guided 3D scene generation.

Limitations. We found three limitations in our method. Firstly, occluded region completion is sometimes incomplete, for example, the one representing a room in Sec. A.7. Secondly, SLAT optimization requires considerable memory, especially for large scenes (computational cost analysis is provided in Sec. A.4). Lastly, our framework shows limited performance on street-level images. The problem is due to a significant mismatch between the scales of the x and y coordinates, arising from the vanishing points. It would be a direction for future work to implement 3D generation from a wider range of image types.

Acknowledgements

This work was supported by the IITP grant funded by MSIT [NO.RS-2021-II211343: AI Graduate School (Seoul National University) (5%), NO.RS-2025-25442338: AI Star Fellowship (45%), and NO.RS-2025-02303703: Real-world multi-space fusion and 6DoF free-viewpoint immersive visualization for extended reality (50%)].

References

[1] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. In *ICML*, 2023. 2, 3

[2] Hanke Chen, Yuan Liu, and Minchen Li. Trellisworld: Training-free world generation from object generators. *arXiv preprint arXiv:2510.23880*, 2025. 13

[3] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *CVPR*, 2022. 1

[4] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. 1

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 3

[6] Ruoyi Du, Dongliang Chang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Demofusion: Democratising high-resolution image generation with no $$$. In *CVPR*, 2024. 2, 3, 5

[7] Paul Engstler, Aleksandar Shtedritski, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Syncity: Training-free generation of 3d worlds. In *ICCV*, 2025. 2, 6, 11, 20

[8] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *IJCV*, 129(12):3313–3337, 2021. 1

[9] Google. Google earth. https://earth.google.com/web, 2025. 1, 6, 22, 23, 24

[10] Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. In *ICLR*, 2024. 2

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[12] Jinho Jeong, Sangmin Han, Jinwoo Kim, and Seon Joo Kim. Latent space super-resolution for higher-resolution image generation with diffusion models. In *CVPR*, 2025. 5

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 42(4), 2023. 2

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[15] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 2, 6, 7, 15, 24

[16] Han-Hung Lee, Qinghong Han, and Angel X. Chang. Nuiscene: Exploring efficient generation of unbounded outdoor scenes. In *ICCV*, 2025. 2

[17] Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. In *NeurIPS*, 2023. 2, 6, 7

[18] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025. 2

[19] Liqiang Lin, Yilin Liu, Yue Hu, Xingguang Yan, Ke Xie, and Hui Huang. Capturing, reconstructing, and simulating: the urbanscene3d dataset. In *ECCV*, 2022. 6, 21, 24, 25

[20] Mingbao Lin, Zhihang Lin, Wengyi Zhan, Liujuan Cao, and Rongrong Ji. Cutdiffusion: A simple, fast, cheap, and strong diffusion extrapolation method. *arXiv preprint arXiv:2404.15141*, 2024. 2, 3, 5

[21] Zhihang Lin, Mingbao Lin, Meng Zhao, and Rongrong Ji. Accdiffusion : An accurate method for higher-resolution image generation. In *ECCV*, 2024. 2, 3, 5, 11

[22] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 2, 3

[23] Yuheng Liu, Xinke Li, Xueting Li, Lu Qi, Chongshou Li, and Ming-Hsuan Yang. Pyramid diffusion for fine 3d large scene generation. In *ECCV*, 2024. 2

[24] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 3

[25] Grace Luo, Trevor Darrell, Oliver Wang, Dan B Goldman, and Aleksander Holynski. Readout guidance: Learning control from diffusion features. In *CVPR*, 2024. 3, 6

[26] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *ICCV*, 2025. 7

[27] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022. 2, 3

[28] Quan Meng, Lei Li, Matthias Nießner, and Angela Dai. Lt3sd: Latent trees for 3d scene diffusion. In *CVPR*, 2025. 2

[29] 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. In *ECCV*, 2020. 2

[30] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6, 7, 14, 15, 16, 17, 18, 19, 20, 21, 24

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr

Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 3

[32] 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 *ICML*, 2021. 7

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3

[34] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3

[35] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 4

[36] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 3

[37] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025. 3

[38] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *CVPR*, 2025. 5

[39] 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. In *NeurIPS*, 2025. 2, 5, 6

[40] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 6

[41] Joey Wilson, Jingyu Song, Yuewei Fu, Arthur Zhang, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. Motionsc: Data set and network for real-time semantic mapping in dynamic environments. *RA-L*, 7 (3):8439–8446, 2022. 6, 14, 24

[42] Haoning Wu, Shaocheng Shen, Qiang Hu, Xiaoyun Zhang, Ya Zhang, and Yanfeng Wang. Megafusion: Extend diffusion models towards higher-resolution image generation without further tuning. In *WACV*, 2025. 2, 5

[43] Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, and Pan Ji. Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. *TOG*, 43(4), 2024. 2

[44] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *CVPR*, 2025. 2, 3, 4, 6, 11

[45] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *TOG*, 42(4):1–16, 2023. 2

[46] Richard Zhang, Phillip Isola, Alexei Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[47] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. 2, 6, 11

[48] Kaizhi Zheng, Yue Fan, Jing Gu, Zishuo Xu, Xuehai He, and Xin Eric Wang. Self-evolving 3d scene generation from a single image. *arXiv preprint arXiv:2512.08905*, 2025. 3, 6, 11, 21

[49] Kaizhi Zheng, Ruijian Zhang, Jing Gu, Jie Yang, and Xin Eric Wang. Constructing a 3d town from a single image. *arXiv preprint arXiv:2505.15765*, 2025. 2, 5