Map2World: Segment Map Conditioned Text to 3D World Generation

Jaeyoung Chung[*1] Suyoung Lee[*†1,3] Jianfeng Xiang[†2,3]
Jiaolong Yang[3] Kyoung Mu Lee[1]

[1] Seoul National University, {robot0321,esw0116,kyoungmu}@snu.ac.kr
[2] Tsinghua University
[3] Microsoft Research Asia, {t-jxiang,jiaoyan}@microsoft.com

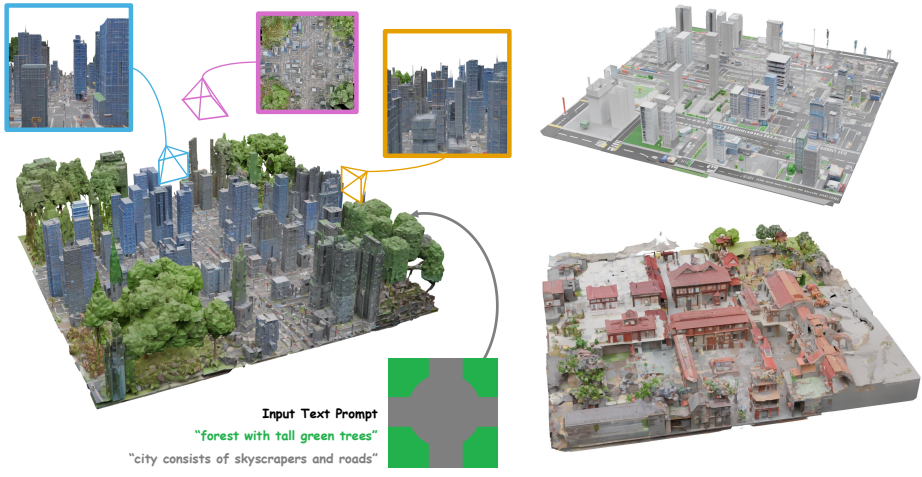


Fig. 1: (Left) Our model takes as input a segment map with text prompts for each segment and produces a corresponding 3D world with high quality. (Right) More examples of generated worlds in large-scale.

Abstract. High-quality world-scale 3D data is scarce, making complete and generalizable 3D world generation difficult. Existing approaches typically rely on either generated images/videos or domain-specific 3D world datasets, resulting in worlds that are often incomplete or restricted to narrow domains. In this paper, we present Map2World, a map-conditioned framework that converts semantic layouts into complete and generalizable large-scale 3D worlds. Given a segment map with per-region text prompts, Map2World uses the map as a controllable world-level interface and leverages TRELLIS, a powerful pretrained 3D asset generator, as a general 3D prior. To scale asset-level priors to world-level synthesis, Map2World enables arbitrary spatial expansion via MultiDiffusion, controls global scale through initial noise optimization, and enhances local geometry and appearance with a dedicated enhancer. Experiments demonstrate that

[*] indicates equal contribution.
[†] works done during internship at Microsoft Research.

Map2World generates complete, layout-controllable, scale-consistent, and detailed 3D worlds across diverse layouts and semantics.

Keywords: 3D World Generation · Structured Latent · 3DGS

1 Introduction

Three-dimensional (3D) world generation plays a pivotal role in a wide range of industrial applications, including immersive content creation, gaming, autonomous driving simulation, and virtual realities. While recent years have witnessed remarkable progress in 3D asset or object generation [17, 48, 56, 59], extending these capabilities to the world scale remains challenging. The primary bottleneck is the lack of high-quality world-level datasets, which are far more difficult to construct than object-centric collections [7, 8].

We view recent efforts toward 3D world generation through two representative paradigms: view-centric generation and world-centric generation. The first repurposes image or video generation models as 3D generators [6, 28, 32, 39, 54]. Benefiting from world priors learned from massive-scale visual data [12, 22, 44], these methods render novel views, estimate depth, and lift the resulting observations into 3D. While this strategy offers strong scalability through abundant 2D supervision, it inherits a fundamental weakness of view-centric synthesis: independently generated views are often geometrically inconsistent, and unobserved or occluded regions cannot be reliably completed. As a result, such methods struggle to produce coherent and complete 3D worlds. The second category trains generative models directly on 3D data. By operating in a world-centric 3D space, these models can in principle synthesize complete scenes beyond any particular camera trajectory. However, they are bottlenecked by the scarcity and limited diversity of large-scale 3D world datasets. Consequently, recent methods such as BlockFusion [47], NuiScene [23], LT3SD [33], WorldGrow [27], SCube [36], and CityDreamer [49] achieve impressive results within their respective target domains, such as medieval villages, forests, urban scenes, and indoor environments. However, this domain specialization limits each model to the particular type of world on which it is designed or trained, preventing general-purpose 3D world generation.

In this paper, we aim to build a 3D world generator that is both complete and generalizable. To this end, we present **Map2World**, a map-conditioned framework that converts semantic layouts into large-scale 3D worlds. Our key idea is to use a segment map as a world-level control interface and instantiate each region with a strong pretrained 3D asset generator, thereby avoiding camera-dependent reconstruction while inheriting broad 3D priors beyond narrow world-scale datasets. Given a segment map with per-region text prompts, Map2World generates a coherent 3D field aligned with the specified layout and semantics. To scale from assets to worlds, we enable arbitrary spatial expansion via MultiDiffusion and introduce initial noise optimization for global scale control. Finally, an detail enhancer enriches local geometry and appearance,

producing worlds that are spatially controllable, scale-consistent, complete, and detailed. Through extensive experiments, we demonstrate that our framework significantly improves both structural fidelity and perceptual realism, paving the way toward scalable and versatile 3D scene generation. In particular, compared with SynCity [11], which leverages TRELLIS to generate grid-wise worlds, we demonstrate that our method achieves better contextual smoothness and scale consistency in 3D worlds.

Our contributions can be summarized as follows:

- Flexible segment map conditioning: generates 3D worlds from any type of user-defined segment maps, not limited to grid-based layouts.
- Consistent detail enhancement: adds fine details to the assets while preserving overall structure.
- Domain-generalized world generation: leverages powerful asset generator priors to achieve robust generation across domains with limited data.

2 Related work

3D world generation. The former approach uses diffusion models to generate the images or videos, and reconstructs a 3D scene from the generated views. 2D image diffusion model-based approaches [6, 16, 26, 41, 52, 53] lift the pixels of the initial image into 3D with a monocular depth estimator. The image is then outpainted, and the generated region is lifted and stitched to the original scene to expand the world. The video diffusion model-based approaches [31, 45, 50, 58] generate a video that navigates a virtual environment and reconstructs the scene from the video frames. Yet, these methods cannot guarantee the 3D consistency since the diffusion models are not trained to be aware of such consistency.

The other approach directly creates the scene with explicit 3D representations, achieving perfect 3D consistency. Some methods, such as BlockFusion [47], NuiScene [23], and LT3SD [33], train the generator from scratch to learn the distribution of cubes that correspond to a scene part. However, they share two major drawbacks resulting from the limited availability of scene datasets. First, these models can only generate geometry, and the generated scene does not have textures. Second, the domain of the generation is limited to the dataset they used for training. SCube [36] and InfiniCube [32] add color estimation and use sparse-voxel-hierarchy [35] to improve global consistency, but they are also constrained to representing only driving scenes, which poses a significant limitation. Several works [11,60] leverage off-the-shelf 3D asset generators [5,29,35,48,56] and expand the scope of generation to 3D world. Although the quality of each asset can be prominent, the overall quality of the scene is highly dependent on the dataset. The strong prior knowledge of a high-quality 3D asset generator enhances both the quality and diversity of generated results; however, the spatial resolution of the generator's output is too limited to represent an entire world. SynCity addresses this issue by dividing the space into multiple grid tiles, generating assets for each tile, and then merging them. Nevertheless, this approach suffers from weak connectivity between tiles and is unable to create large objects with

arbitrary shapes. In contrast, our model supports seamless world generation from arbitrary segment maps using the latent fusion strategy.

Scaling diffusion models to large spatial extents. The idea of large-scale 3D generation can originate from the 2D diffusion literature, where training-free techniques have been developed to expand the resolution of pretrained models beyond their limits. Patch-based approaches synthesize large images by denoising overlapping patches and stitching them together, effectively bypassing the base model's resolution and memory constraints [9, 18]. However, these methods typically rely on rigid grid tilings and heuristic blending, which can introduce boundary artifacts and offer limited flexibility for region-wise semantic control. Outpainting-based methods progressively extend an input canvas beyond its original field of view, enabling directional or large-aspect-ratio image expansion [4, 21,42,46]. Despite this flexibility, they remain confined to 2D planar domains and are usually driven by a single global context, providing only coarse control over the semantics of newly generated regions. Multi-window diffusion frameworks treat a large canvas as a collection of overlapping windows that are denoised jointly, allowing different prompts or conditions to be assigned to different spatial regions while maintaining global coherence through latent or score fusion [2,10,19,24,25]. Building on this idea, we extend the multi-window paradigm to volumetric 3D, operating on arbitrary-shaped, user-defined 3D regions and fusing their denoising trajectories in a shared 3D latent space to construct large, coherent 3D scenes.

3 Preliminary: Structured Latent and Generation Pipeline

We summarize TRELLIS [48], a state-of-the-art 3D asset generation model, using the new representation, structured latent, and the two-stage generation process. The structured latent (or SLAT) encodes geometry and appearance with a set of local latents on a 3D grid,

$$\boldsymbol{s} = \{(\boldsymbol{z}_i, \boldsymbol{p}_i)\}_{i=1}^{L}, \tag{1}$$

where $\boldsymbol{p}_i \in \{0, 1, \ldots, N\text{-}1\}^3$ is the positional index of an active voxel in the N^3 3D grid, $\boldsymbol{z}_i \in \mathbb{R}^C$ denotes a latent vector that encodes geometry and appearance at the corresponding position $\boldsymbol{p}_i$, and L is the number of active voxels.

The structured latent is generated by a two-stage generation pipeline, from geometry to texture, based on the rectified flow model. Here, we assume the input text prompt y is given as a condition. For the first stage, they generate sparse structure $\{\boldsymbol{p}_i\}_{i=1}^{L}$ by iteratively applying the flow Transformer $\boldsymbol{\mathcal{G}}_S$ conditioned by y to the Gaussian noise. The fully denoised sparse structure latent is passed through the decoder, denoted as $\boldsymbol{\mathcal{D}}_S$, and transformed into a signed 3D scalar field where the voxels with positive values are filled with contents and called active. We record a set of position indices of all active voxels as $\{\boldsymbol{p}_i\}_{i=1}^{L}$. In the second stage, the set of latent features $\{\boldsymbol{z}_i\}_{i=1}^{L}$ is also estimated through iterative denoising steps using the Transformer called $\boldsymbol{\mathcal{G}}_L$, similar to the first stage. The structured latent can be transformed with various types of 3D representations, including 3D Gaussian splatting [20], radiance fields [13], and mesh [40], by

putting the latent to the corresponding decoder ($\boldsymbol{\mathcal{D}}_L$). In this experiment, we mainly focus on generating 3DGS as a final 3D representation.

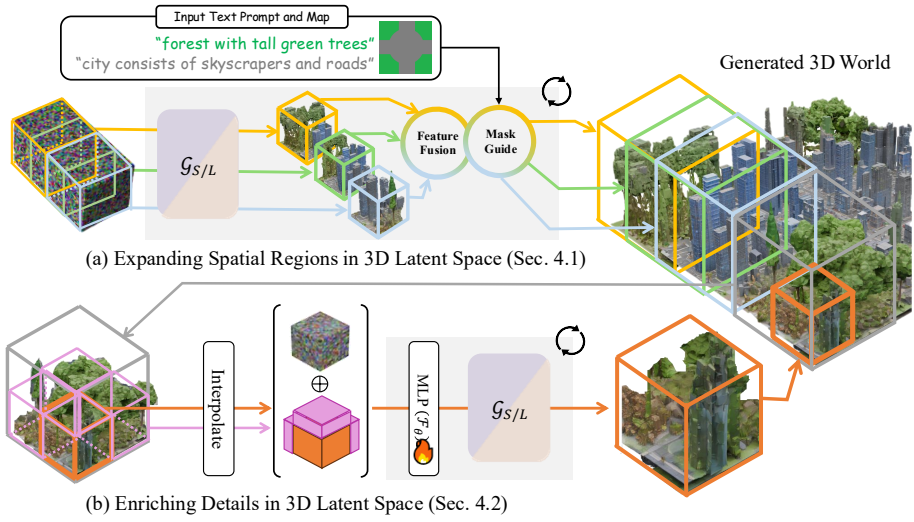


Fig. 2: Visualization of the overall generation pipeline. First, we estimate the structured latent for a large world conditioned by a segmentation map and a text prompt for each segment using latent fusion (top). Then, we further upscale the resolution of the generated scene using the detail enhancer. Detail enhancer includes an MLP layer to incorporate condition latent and noise, as well as a flow Transformer (bottom).

4 Proposed Method

We describe how Map2World exploits the knowledge of a 3D asset generator to synthesize a 3D world. In Sec. 4.1, we present our latent fusion strategy to expand generation to a wider scene and to support conditioning on multiple text-driven spatial regions of arbitrary shapes. Sec. 4.2 proposes a detail enhancing network to enrich the details of the world by manipulating the structured latent. Here, we explain the design choice of the detail enhancer to leverage the representational power of TRELLIS while preserving the content of the input large 3D scene. Finally, Sec. 4.3 introduces decoder fine-tuning to generate high-quality 3D scenes. The overall generation process is illustrated in Fig. 2.

4.1 Expanding Spatial Regions in 3D Latent Space

The active voxel space of assets generated by the pre-trained TRELLIS model is limited to a cube of size 64. While this resolution may be sufficient for representing a single object, it is inadequate for modeling a large world composed of numerous objects. To achieve spatial expansion using the pre-trained model, we draw inspiration from spatial expansion techniques in 2D diffusion [2] and apply the

idea to both 3D volumetric tensors and sparse structures in the rectified-flow Transformers, $\boldsymbol{\mathcal{G}}_S$ and $\boldsymbol{\mathcal{G}}_L$. Our method supports precise and controllable world generation conditioned on fine-grained, pixel-level spatial regions annotated with textual conditions. We further describe a 3D noise initialization strategy that reinforces global scale consistency during progressive generation.

Latent fusion for rectified flow models. We conceptually split the space into a set of overlapping 3D cube windows $\{\Omega_j\}$, where each window spans a cubic region of size 64 and adjacent windows overlap by half of their spatial resolution. From a given position $\mathbf{x}$, we update its local latent by aggregating the velocity field predictions $v_j(\mathbf{x})$ from all windows spatially covered as $\mathcal{A}(\mathbf{x}) = \{\, j \mid \mathbf{x} \in \Omega_j \,\}$. Using a shared 3D Gaussian kernel $W(\cdot)$, the fused velocity for the position $\mathbf{x}$ is:

$$v_t(\mathbf{x}|y) = \frac{\sum_{j \in \mathcal{A}(\mathbf{x})} W(\mathbf{x} - \mathbf{c}_j)\, v_{t,j}(\mathbf{x}|y)}{\sum_{j \in \mathcal{A}(\mathbf{x})} W(\mathbf{x} - \mathbf{c}_j)}, \tag{2}$$

where $\mathbf{c}_j$ is the center of Ω_j. This strategy is applied to both $\boldsymbol{\mathcal{G}}_S$ and $\boldsymbol{\mathcal{G}}_L$, and the latent at $\mathbf{x}$ is updated at every step according to the rectified-flow formulation:

$$\boldsymbol{s}_{t-1}(\mathbf{x}|y) = \boldsymbol{s}_t(\mathbf{x}|y) - \Delta t \cdot v_t(\mathbf{x}|y). \tag{3}$$

Segment-map-guided latent fusion. Map2World supports 3D world generation conditioned by multiple text prompts with a segment map. Assuming that we have K labels, we define M_k as a binary map that indicates the region of the k-th label, and y_k as the corresponding text prompt. The segment map is provided as a 2D layout, and M_k is obtained as a 3D mask by expanding this 2D region uniformly along the height axis. Here, the velocity at the position $\mathbf{x}$ is calculated by the weighted sum of velocities estimated for each label:

$$\tilde{v}_t(\mathbf{x}) = \frac{\sum_{k=1}^{K} \left(M_k(\mathbf{x}) \odot G(\sigma_t)\right) \cdot v_t(\mathbf{x}|y_k)}{\sum_{k=1}^{K} \left(M_k(\mathbf{x}) \odot G(\sigma_t)\right)}. \tag{4}$$

$G(\sigma_t)$ is a normalized 3D Gaussian kernel with zero mean and standard deviation σ_t to ensure smooth transition across regions, improving denoising stability. Here, σ_t is controlled by the diffusion time step, starting from a large value to produce soft boundaries and gradually changing toward a sharp mask as t decreases. The smooth transitions of the segment mask across regions improve stability during the diffusion process. It is noteworthy that this pipeline can be applied to any shape of M_k, whereas the existing frameworks can only generate objects of the same type in a square-shaped region.

Optimization for scale-aware initial latent. We observe two consistent behaviors in the noisy latent space. First, coarse geometric structures vary

significantly with different initial noise samples. Second, similar initial noise samples tend to produce similar outputs. These observations indicate that the mapping from the initial latent x_T to the final generation is locally smooth yet highly sensitive to the initial condition.

Although TRELLIS does not explicitly split the scale of the scene, we empirically find that sparse structures exhibit different scene scales, and structures with similar scales tend to appear near each other in the latent space, as illustrated in Fig. 4. This suggests that valid sparse structures lie on a scale-dependent manifold. To steer generation toward a desired scale, we optimize the initial noisy latent inspired by the idea of initial noise optimization [1]. For the sparse structure S_T and rectified flow models $\mathcal{G}_S$, we approximate the denoising trajectory by

$$S(t) \approx S_T + (1 - \tfrac{t}{T})[\mathcal{G}_S(S_T) - S_T]_{\text{sg}}, \tag{5}$$

to calculate the gradient without backpropagating through the full denoising trajectory. Here, $[\cdot]_{\text{sg}}$ denotes the stop-gradient operator. Under this approximation, the initialization is optimized via

$$\mathcal{L}_{\text{linear}} = \left\| y - \mathcal{M}([\mathcal{G}_L(S_T) - S_T]_{\text{sg}} + S_T) \right\|_2^2, \tag{6}$$

where $\mathcal{M}$ denotes the target mask and y represents the target constraint for guiding the scale. Through empirical observation of generated samples, we find that the empty spaces and the floors form consistent patterns that can be grouped by scene scale, as illustrated in Fig. 4. Accordingly, we categorize the samples by scale and, for each scale level, designate the region that is common across samples, namely the excluded regions and the ground, as the optimization target y. The loss mask $\mathcal{M}$ then restricts the optimization in Eq. (6) to this common region so that only y is constrained. Because y is defined on such a common region, a single representative sparse structure empirically selected per scale level is sufficient to align the initial noise: the optimization moves the initial latent onto the manifold that satisfies the given constraint, so it does not need to be repeated for every prompt [1]. We emphasize that this loss mask $\mathcal{M}$ is distinct from the segment mask M_k introduced in Sec. 4.1: $\mathcal{M}$ confines the scale-control loss to the common region y, whereas M_k is the 3D segment mask obtained by expanding the 2D segment map along the height axis. Further, to stabilize optimization under large learning rates required for early structural updates, we parameterize the sparse structure feature S in the spectral domain using a 3D FFT. This parameterization stabilizes the optimization trajectory and enables the use of a larger learning rate.

4.2 Enriching Details in 3D Latent Space

The generated structured latent represents the geometry and the texture of an entire world. Since the entire world consists of numerous objects with a ground, it is almost impossible to encode all the details of the world into the latent space with limited capacity. Thus, we propose a detail-enhancing framework

which increases the resolution of the world conditioned by the latent of the input world. It is challenging to train a network that directly adds details in 3D space since collecting such data pairs for detail enhancement is infeasible. Instead, we design the detail enhancer to learn the relationship between coarse and fine scenes in the latent space, thereby maximizing the utilization of TRELLIS' prior knowledge. To construct dataset pairs for training, we segment existing 3D scene data into cubes. Each cube is designed to contain a sufficient amount of content. Subsequently, each cube is divided into two parts along each axis, resulting in a total of eight smaller cubes. Both the large cube and the eight small cubes are then passed through the TRELLIS encoder. Since each latent corresponding to a small cube encodes a relatively smaller spatial region, it contains more detailed information than an input scene latent. The detail enhancer is trained to estimate the eight latent tensors, where each corresponds to a small cube, conditioned on the latent tensor of the input scene. While existing auto-regressive pipelines such as BlockFusion [47] and NuiScene [23] aim to predict latents for rendering individual cubes, they face challenges in maintaining global consistency, as these models rely solely on the information of neighboring cubes. In contrast, our method incorporates a latent variable that encapsulates the entire scene as a condition, and it can achieve high consistency across the whole scene.

We denote the input 3D representation inside a large cube as $\mathcal{C}^{\mathcal{O}}$ and the split small cubes as $\mathcal{C}^j$, where $j = 0, 1, ..., 7$. The structured latent of the large cube is denoted as $\boldsymbol{s}^{\mathcal{O}} = \{(\boldsymbol{z}_i^{\mathcal{O}}, \boldsymbol{p}_i^{\mathcal{O}})\}_i$, and the structured latent of the small cube at index j is denoted as $\boldsymbol{s}^j = \{(\boldsymbol{z}_i^j, \boldsymbol{p}_i^j)\}_i$.

Network architecture. The goal of the detail enhancer is to generate the structured latent of small cubes ($\{\boldsymbol{s}^j\}$) conditioned by the structured latent of the input scene ($\boldsymbol{s}^{\mathcal{O}}$). Instead of estimating the structured latent for all small cubes at once, the detail enhancer receives the cube index (j) and predicts the corresponding structured latent($\boldsymbol{s}^j$). When designing the network for detail enhancement, we propose an additional module that accommodates our new conditions and integrates it with the existing flow Transformer of TRELLIS. The rationale behind this design choice is twofold: first, due to the scarcity of world-level data, it is impractical to develop an entirely new architecture and train all parameters from scratch; second, since the input conditions also reside within TRELLIS's latent manifold, leveraging the same structure facilitates more efficient processing and integration of condition information.

We employ structured latents from two different cubes as conditioning inputs for detail enhancing network. First, the structured latent of a large cube, $\boldsymbol{s}^{\mathcal{O}}$, provides rough information about the target cube. To leverage the information, we extract only the part of the latent corresponding to the spatial location of our target cube, which we denote the truncated latent as $\boldsymbol{s}^{\mathcal{O}|j}$. This structured latent plays a role analogous to a low-resolution image in image super-resolution. Thus, we concatenate the noise and the condition latent along the channel axis and define an MLP layer ($\boldsymbol{F}_\theta$) to mix the information, which is proven effective in a diffusion-based image enhancement field [37, 38]. We denote the channel

dimensions of the noise feature and latent feature tensor as c and C, respectively; then, the input dimension of the MLP layer is $(c+C)$, and the output dimension is c, which is the same as the dimension of the noise feature.

Next, while $\boldsymbol{s}^{\mathcal{O}|j}$ can be sufficient to enhance the details of the corresponding small target cube, relying solely on this condition does not guarantee that the geometry and textures between adjacent cubes will be seamlessly connected. To achieve the seamless connection, we also incorporate the structured latent of adjacent cubes facing the target cube. The set of structured latent for adjacent cubes with target index j is denoted as $\boldsymbol{s}^{Adj(j)}$. Here, $Adj(j)$ denotes the position indices of cubes that are adjacent to the j-th cube. In order to concatenate the latent features with the noise, as we did with the large cube latent ($\boldsymbol{s}^{\mathcal{O}|j}$), we temporarily expand the spatial size of the noise. The expanded noise is concatenated channel-wise with the adjacent latent and then compressed through the MLP layer. We note that the MLP layers for $\boldsymbol{s}^{\mathcal{O}|j}$ and $\boldsymbol{s}^{Adj(j)}$ share the same parameters. The mixed feature after the MLP layer is then fed into the flow Transformer of the original model($\boldsymbol{\mathcal{G}}_S$ and $\boldsymbol{\mathcal{G}}_L$). From the predicted flow, we crop the expanded region and retain only the portion corresponding to the target cube. The overall process can be summarized as follows:

$$\boldsymbol{v}_\theta\left(\boldsymbol{s}^j, t\right) = \boldsymbol{\mathcal{G}}_{S/L}\left(\boldsymbol{F}_\theta\left(\boldsymbol{s}_t^j, \boldsymbol{s}^{\mathcal{O}|j}, \boldsymbol{s}^{Adj(j)}\right), t\right), \tag{7}$$

where $\boldsymbol{s}_t^j = (1-t)\boldsymbol{s}^j + t\boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \in \mathcal{N}(0, \boldsymbol{I})$.

Initialization, training, and sampling. Similar to many diffusion fine-tuning strategies [34, 57], our model is initialized to behave identically to the original TRELLIS before fine-tuning, and gradually incorporates the new conditioning as training progresses. Consequently, we use the pre-trained parameters of the original model for the flow Transformer. For the proposed MLP layer($\boldsymbol{F}_\theta$), the weight matrix is initialized such that its diagonal elements are set to 1, while all other elements and the bias are set to 0. This ensures that, before training, the output of $\boldsymbol{F}_\theta$ is identical to the noise regardless of the condition values. Then, the flow matching loss [30] is applied to fine-tune our model. The loss function to fine-tune the detail enhancer can be written as:

$$\mathcal{L}_\theta = \mathbb{E}_{\boldsymbol{s}^j, t}\|\boldsymbol{v}_\theta\left(\boldsymbol{s}^j, t\right) - (\boldsymbol{\varepsilon} - \boldsymbol{s}^j)\|_2^2. \tag{8}$$

We note that only the parameters of the MLP layer ($\boldsymbol{F}_\theta$) are fine-tuned.

After fine-tuning, we consecutively apply the model to sample the latent from the noise. We auto-regressively estimate the structured latent of small cubes from index 0 to 7. When estimating the latent of cube 0 ($\boldsymbol{s}^0$), only the large cube structured latent is used since there is no information on adjacent cubes. When generating latent for subsequent cubes, the latent of adjacent cubes that have already been generated are also utilized as conditioning inputs. The eight estimated structured latents are merged and compose a latent that can represent a scene with enhanced details.

It is noteworthy to mention that we do not use classifier-free guidance (CFG) [15] when fine-tuning or sampling the model. CFG encourages the model to generate same output for both conditional and unconditional settings, and use the difference to guide the denoising direction during sampling; however, in our case, the existence of conditions leads to substantial difference in reconstruction performance, thereby reducing the effectiveness of the CFG strategy.

4.3 Fine-tuning SLAT Decoder

Since the original TRELLIS decoder was trained exclusively on data representing complete objects, its performance degrades when tasked with representing cubes extracted from partial scenes. To overcome the distribution discrepancy, we fine-tuned the latent decoder ($\mathcal{D}_L$) using the small cubes used for training of the detail enhancer. Specifically, we extracted structured latents from the 3D meshes of these small cubes using the pre-trained encoder. Then we fine-tuned the decoder by minimizing the difference between the reconstructed 3D representation and the original mesh. The loss function follows the original TRELLIS formulation. Through decoder fine-tuning, we can obtain higher-quality 3D representations from the structured latents estimated by the detail enhancer.

5 Experiments

Dataset curation. Using the labels of NuiScene43 [23], we choose a set of filtered 43 high-quality scene meshes from Objaverse [8] to train the detail enhancers. We exclude eight scenes without textural information and extract cubes from the remaining 35 scenes. We randomly cropped 500 cubes with various sizes ($\in \{64, 128, 192, 256\}$) for each scene. Here, we checked that each cube includes more than 10,000 vertices. After a total of 17,500 cubes have been extracted, we treat each cube as a large cube and split it into eight identical small cubes. For a large cube and the corresponding small cubes, we extract sparse structure latent and structured latent tensors using the pre-trained encoder models of TRELLIS. We use 16,000 pairs to train the enhancer and 1,500 pairs for validation.

5.1 Comparison on World Generation

Map2World takes user-provided text prompts and a segmentation map comprising multiple segments labeled by each prompt as input and generates a large-scale world that satisfies these conditions. We note that our model is not restricted to the size of the voxel grid of the original asset generator and can create the world with any user-defined dimensions. Also, the region of each segment can be formed in an arbitrary shape. Fig. 3 displays the rendered samples of the world created by Map2World, given the segment map with free-form segments. The three examples shown in the figure differ not only in their input prompts but also in the number of segments, the segment shapes, and the overall dimensions of the target world. SynCity cannot generate an appropriate 3D world under

(a) Segment map **(b)** Top-view map **(c)** Generated world **(d)** Roaming view

Fig. 3: Qualitative results conditioned by arbitrary shaped segment maps with user-defined text prompts.

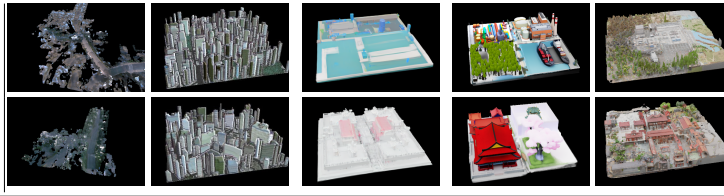
these conditions. In contrast, our model successfully produces a 3D world that aligns well with both the segment map shapes and the text prompts in all cases. Fig. 3b displays a top-down view of the generated world, where each region closely matches the input segment map. Furthermore, as illustrated in Figs. 3c and 3d, the structures of adjacent regions are seamlessly connected, ensuring smooth transitions between segments. A qualitative comparison with SynCity [11] under a grid-shaped segment map is provided in **??**.

Tab. 1 reports a quantitative comparison on world generation, where all methods are given the same text prompt and the resulting worlds are evaluated under three groups of metrics. We compare Map2World against four baselines: InfiniCube [32], CityDreamer [49], GaussianCube [55], and SynCity [11]. Among them, SynCity is the most direct baseline as it also builds on TRELLIS for map-conditioned scene generation, while InfiniCube and CityDreamer target domain-specific driving and urban scenes, and GaussianCube performs text-to-3D generation without layout conditioning. We evaluate the generated worlds using a GPTScore-based World Quality metric, distribution-based FID/KID, and a human preference study, which we describe in turn below.

World Quality (WQ). We evaluate generated environments using a composite GPTScore-based metric, *World Quality (WQ)*, defined as

$$WQ = 0.15S + 0.45W + 0.25C + 0.15R, \tag{9}$$

Table 1: Quantitative comparison of different methods and metrics.

Method	InfiniCube [32]	CityDreamer [49]	GaussianCube [55]	SynCity [11]	Map2World
Example images					
GPTScore					
S (sharpness)	6.5	7.8	6.8	8.2	8.0
W (world completeness)	3.8	8.5	4.5	6.8	7.8
C (coherence)	3.2	6.2	5.0	7.6	7.9
R (realism)	4.6	5.9	5.1	7.3	7.6
WQ	4.05	7.36	5.08	7.25	**7.76**
Distribution-based Metrics					
FID (Incep.v2)$_{\downarrow}$	268.0	272.8	169.1	**143.4**	192.4
KID (Incep.v2)$_{\downarrow}$	0.2144	0.2376	**0.0906**	0.1060	0.1480
Human Evaluation					
Realism	0.9474	0.8596	0.9825	0.9474	Reference
Scene completeness	0.9825	0.8070	0.6667	0.9298	Reference
Scene coherence	0.9825	0.8772	0.8947	0.9474	Reference
Overall quality	0.9825	0.9123	0.9825	0.8947	Reference

where S, W, C, and R denote *sharpness*, *world completeness*, *coherence*, and *realism*, respectively, each rated by the GPT 5.3 model from the rendered images. The metric assigns the highest weight to world completeness so that it captures large-scale environmental structure rather than the fidelity of individual objects, making it suitable for evaluating world-scale generative models. We provide the full per-axis definitions and the design rationale of the weighting in **??**. GaussianCube produces environments with noticeably lower visual quality and incomplete scene structures, resulting in low scores across most evaluation criteria. While SynCity benefits from the strong image-to-3D expressive capability of TRELLIS, thereby achieving high sharpness, the generated assets are often connected in an unnatural manner. These inconsistencies reduce the structural completeness of the generated environments and lead to a relatively lower world completeness score. In contrast, Map2World achieves consistently high scores across all evaluation criteria, producing environments that are not only visually clear but also structurally coherent and complete at the world level.

Distribution-based metrics. We report distribution-based metrics, FID [14] and KID [3], computed with an Inception v2 [43] extractor between each method and a ground-truth reference of 1,050 images rendered from the 35 meshes. As shown in Tab. 1, Map2World does not achieve the best values. However, we argue that these distribution-based metrics are unreliable for scene-level evaluation: the reference set of only 35 scenes is far too small to represent the underlying 3D scene manifold, and the heavy computation of world generation prevents sampling enough diverse outputs to cover it. Moreover, 3D-aware properties

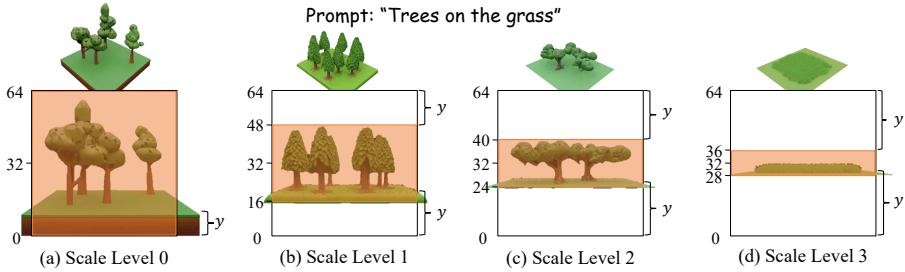


Fig. 4: Examples generated from the same prompt but different scale levels.

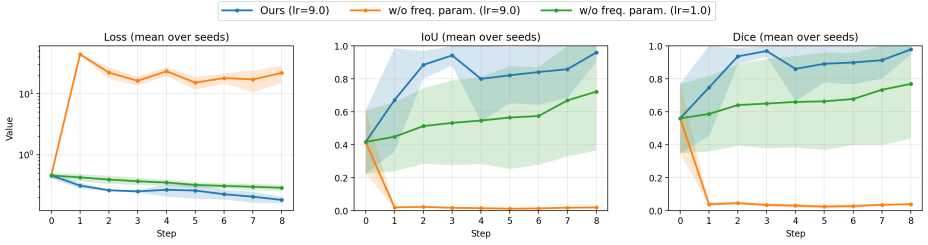


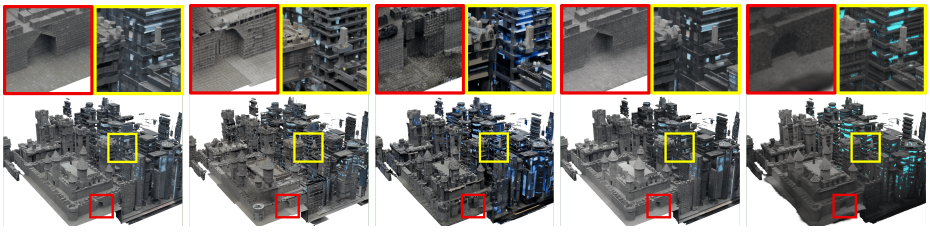
Fig. 5: Ablation on spectral-domain parameterization. This parameterization stabilizes the optimization process, enabling rapid convergence to the target within a few steps using a high learning rate.

such as geometric completeness and map alignment are not captured by image-distribution distances, so these scores should be interpreted only as a coarse reference rather than a faithful measure of world-level quality.

Human evaluation. To obtain a more direct perceptual signal, we further conduct a pairwise human evaluation along four axes: realism, scene completeness, scene coherence, and overall quality. We recruited 19 participants and collected A/B preferences over 21 randomly sampled pairs, and report in Tab. 1 the proportion of comparisons in which Map2World was preferred over each baseline (with Map2World itself serving as the reference). Map2World is preferred in 67–98% of comparisons across all baselines and axes, providing strong perceptual evidence that is consistent with the WQ trends and far better aligned with human judgment than the distribution-based metrics above.

5.2 Ablation Studies

Spectral parameterization for stable initial latent optimization. In the iterative optimization of initial noise for scale control illustrated in Fig. 4, the repeated generation increases both generation time and computational cost. Therefore, we introduce the spectral domain parameterization of the sparse structure S as described in Sec. 4.1, to stabilize the optimization trajectory, enable the use of a large learning rate, and enrich the target scale within a few optimization steps. We present the IoU and Dice plots measuring how well



(a) Ours **(b)** IP-Adapter **(c)** Use CFG **(d)** No $\mathcal{D}_L$ finetune **(e)** No detail enhan.

Fig. 6: Qualitative comparison of rendered scenes on design choices for the detail enhancer. *Best viewed when zoomed in.*

Table 2: Quantitative comparison for the detail enhancer design. The configuration of our choice and the best metric are written in **bold**.

	$\mathcal{G}_{S/L}$ Architecture	CFG	$\mathcal{D}_L$ F.T.	PSNR$_\uparrow$	LPIPS$_\downarrow$	FID$_\downarrow$ Incep.v3	DINOv2	CLIP
(a)	**Concatenation**	**No**	**Yes**	**22.53**	**0.2137**	**16.98**	**32.67**	**11.79**
(b)	IP-Adapter	No	Yes	20.28	0.2499	29.62	80.81	19.85
(c)	Concatenation	Yes	Yes	21.95	0.2174	19.06	38.15	21.32
(d)	Concatenation	No	No	22.08	0.2165	17.89	32.94	13.17

the target geometry constraint is satisfied in Fig. 5. In our setting, the blue curve shows that with a high learning rate of 9.0, the IoU and Dice scores reach approximately 0.9 on average within five optimization steps. In contrast, the orange curve, which directly optimizes the sparse structure without spectral-domain parameterization under the same setting, exhibits unstable behavior with large loss spikes and fails to converge. Reducing the learning rate to 1.0 (green curve) avoids divergence but requires significantly more optimization steps, increasing the computational burden.

Design choices for the detail enhancer. We conduct ablation studies on the design of the detail enhancer and decoder fine-tuning to justify our design choice. Fig. 6 shows rendered images from generated scenes by changing the options while keeping the condition, seed, and camera parameters the same. Fig. 6e is the rendered sample without using the detail enhancer; the generated structured latent with latent fusion is decoded and rendered.

First, we apply different network architectures to compare the performance. Specifically, we adapt the architecture from IP-Adapter [34, 51] to fine-tune $\mathcal{G}_{S/L}$. The IP-adapter is also designed for fine-tuning the model to the new condition in a parameter-efficient manner. However, as shown in Fig. 6b, IP-Adapter fails to generate the appropriate structure and shows disconnectedness at the boundary. Also, we use **classifier-free guidance** during fine-tuning and sampling. The sampled result is shown in Fig. 6c. In our setting, the difference in generation quality between the conditional and unconditional models would be substantial. When denoising with CFG, the difference between the denoised features in the

conditional and unconditional settings becomes excessively large, causing severe geometric distortions and overly saturated colors. Finally, we analyze the effect of **SLAT decoder fine-tuning** in Fig. 6d. Although the improvement was less pronounced compared to the changes in the detail enhancer, we observed that the fine-tuned decoder produced sharper geometry and more refined textures.

To quantitatively compare the options, we use the test meshes and consider the rendered images from the mesh as ground truth. For the metrics, we compute PSNR, LPIPS, and FID scores on the rendered images from the test-set generated scenes and record the results in Tab. 2. For FID, we measured the scores across three baseline models: Inceptionv3, DINOv2, and CLIP. We note that our detail enhancer is a generator; thus, PSNR and LPIPS are not ideal metrics for evaluating quality. Still, our model achieves the best PSNR and LPIPS among the options, indicating the highest fidelity to the input condition. Our choice also yields the best FID metrics across all three baselines, suggesting that our model generates the most natural and high-quality detailed structures and textures. Furthermore, in a user study comparing rendered views with and without the detail enhancer, 79% of respondents preferred the enhanced results for their superior overall quality.

6 Conclusion

Despite the growing demand for creating worlds, existing methods fail to achieve both flexible and scalable 3D world generation due to the shortage of appropriate data. In this work, we present Map2World, a novel text-guided 3D world generation pipeline by exploiting the prior knowledge of TRELLIS [48], a popular 3D object generation model. Our model supports segment-map-guided sampling and allows users to generate the world from a user-defined map composed of arbitrary-shaped regions, which is not available in any existing works. Also, we impose constraints on noise initialization and employ a latent fusion strategy [2] during sampling to share features across the world to ensure global consistency. Next, we propose a detail enhancing module to further improve the quality of the generated world. The detail enhancer incorporates the global structure latent as a condition to preserve the consistency of generated details on a global scale. Our model successfully generates worlds that are well aligned with the segment maps and exhibit smooth boundary transitions, even for segment maps of diverse sizes and shapes.

Acknowledgements

This research was supported by the MSIT(Ministry of Science, ICT), Korea, under the Global Research Support Program in the Digital Field program(RS-2024-00436680) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). This project is supported by Microsoft Research Asia.

References

1. Baek, S., Dong, E., Namazifard, S., Matthews, M.J., Yi, K.M.: Sonic: Spectral optimization of noise for inpainting with consistency. arXiv preprint arXiv:2511.19985 (2025)
2. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: Multidiffusion: Fusing diffusion paths for controlled image generation. In: ICML (2023)
3. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans. In: ICLR (2018)
4. Chen, Q., Ma, Y., Wang, H., Yuan, J., Zhao, W., Tian, Q., Wang, H., Min, S., Chen, Q., Liu, W.: Follow-your-canvas: Higher-resolution video outpainting with extensive content generation. arXiv preprint arXiv:2409.01055 (2024)
5. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In: ICCV. pp. 22246–22256 (2023)
6. Chung, J., Lee, S., Nam, H., Lee, J., Lee, K.M.: Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. IEEE TVCG (2025)
7. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. NeurIPS (2023)
8. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: CVPR. pp. 13142–13153 (2023)
9. Ding, Z., Zhang, M., Wu, J., Tu, Z.: Patched denoising diffusion models for high-resolution image synthesis. In: ICLR (2023)
10. Du, R., Chang, D., Hospedales, T., Song, Y.Z., Ma, Z.: Demofusion: Democratising high-resolution image generation with no $$$. In: CVPR (2024)
11. Engstler, P., Shtedritski, A., Laina, I., Rupprecht, C., Vedaldi, A.: Syncity: Training-free generation of 3d worlds. arXiv preprint arXiv:2503.16420 (2025)
12. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
13. Gao, Q., Xu, Q., Su, H., Neumann, U., Xu, Z.: Strivec: Sparse tri-vector radiance fields. In: ICCV (2023)
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
15. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS Workshop (2021)
16. Höllein, L., Cao, A., Owens, A., Johnson, J., Nießner, M.: Text2room: Extracting textured 3d meshes from 2d text-to-image models. In: ICCV. pp. 7909–7920 (2023)
17. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. In: ICLR. vol. 2024, pp. 50678–50702 (2024)
18. Jiang, W., Jangid, D.K., Lee, S.J., Sheikh, H.R.: Latent patched efficient diffusion model for high resolution image synthesis. In: CVPR (2025)
19. Jiménez, Á.B.: Mixture of diffusers for scene composition and high resolution image generation. arXiv preprint arXiv:2302.02412 (2023)
20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM TOG **42**(4) (2023)

21. Kim, K., Yun, Y., Kang, K.W., Kong, K., Lee, S., Kang, S.J.: Painting outside as inside: Edge guided image outpainting via bidirectional rearrangement with progressive step learning. In: WACV (2021)
22. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
23. Lee, H.H., Han, Q., Chang, A.X.: Nuiscene: Exploring efficient generation of unbounded outdoor scenes. In: ICCV. pp. 26509–26518 (2025)
24. Lee, J., Jung, D.S., Lee, K., Lee, K.M.: Streammultidiffusion: real-time interactive generation with region-based semantic control. CVPR (2025)
25. Lee, Y., Kim, K., Kim, H., Sung, M.: Syncdiffusion: Coherent montage via synchronized joint diffusions. In: NeurIPS (2023)
26. Li, H., Shi, H., Zhang, W., Wu, W., Liao, Y., Wang, L., Lee, L.h., Zhou, P.Y.: Dreamscene: 3d gaussian-based text-to-3d scene generation via formation pattern sampling. In: ECCV. pp. 214–230 (2024)
27. Li, S., Yang, C., Fang, J., Yi, T., Lu, J., Cen, J., Xie, L., Shen, W., Tian, Q.: Worldgrow: Generating infinite 3d world. In: AAAI. vol. 40, pp. 6433–6441 (2026)
28. Liang, H., Cao, J., Goel, V., Qian, G., Korolev, S., Terzopoulos, D., Plataniotis, K.N., Tulyakov, S., Ren, J.: Wonderland: Navigating 3d scenes from a single image. In: CVPR (2025)
29. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: CVPR (2023)
30. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023)
31. Liu, X., Zhou, C., Huang, S.: 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. In: NeurIPS. vol. 37, pp. 133305–133327 (2024)
32. Lu, Y., Ren, X., Yang, J., Shen, T., Wu, Z., Gao, J., Wang, Y., Chen, S., Chen, M., Fidler, S., et al.: Infinicube: Unbounded and controllable dynamic 3d driving scene generation with world-guided video models. In: ICCV. pp. 27272–27283 (2025)
33. Meng, Q., Li, L., Nießner, M., Dai, A.: Lt3sd: Latent trees for 3d scene diffusion. In: CVPR. pp. 650–660 (2025)
34. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023)
35. Ren, X., Huang, J., Zeng, X., Museth, K., Fidler, S., Williams, F.: Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In: CVPR (2024)
36. Ren, X., Lu, Y., Liang, H., Wu, J.Z., Ling, H., Chen, M., Fidler, S., Williams, F., Huang, J.: Scube: Instant large-scale scene reconstruction using voxsplats. In: NeurIPS (2024)
37. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
38. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. In: NeurIPS (2022)
39. Shen, T., Bahmani, S., He, K., Srinivasan, S.G., Cao, T., Ren, J., Li, R., Wang, Z., Sharp, N., Gojcic, Z., et al.: Lyra 2.0: Explorable generative 3d worlds. arXiv preprint arXiv:2604.13036 (2026)

40. Shen, T., Munkberg, J., Hasselgren, J., Yin, K., Wang, Z., Chen, W., Gojcic, Z., Fidler, S., Sharp, N., Gao, J.: Flexible isosurface extraction for gradient-based mesh optimization. ACM TOG **42**(4) (2023)
41. Shriram, J., Trevithick, A., Liu, L., Ramamoorthi, R.: Realmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion. In: 3DV (2025)
42. Song, D.Y., Yu, J.J., Cho, D.: Progressive artwork outpainting via latent diffusion models. In: ICCV (2025)
43. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. pp. 2818–2826 (2016)
44. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
45. Wang, H., Liu, F., Chi, J., Duan, Y.: Videoscene: Distilling video diffusion model to generate 3d scenes in one step. In: CVPR. pp. 16475–16485 (2025)
46. Wu, T., Zheng, C., Cham, T.J.: Panodiffusion: 360-degree panorama outpainting via diffusion. In: ICLR (2024)
47. Wu, Z., Li, Y., Yan, H., Shang, T., Sun, W., Wang, S., Cui, R., Liu, W., Sato, H., Li, H., Ji, P.: Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. ACM TOG **43**(4), 1–17 (2024)
48. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: CVPR. pp. 21469–21480 (2025)
49. Xie, H., Chen, Z., Hong, F., Liu, Z.: Citydreamer: Compositional generative model of unbounded 3d cities. In: CVPR (2024)
50. Yan, Y., Xu, Z., Lin, H., Jin, H., Guo, H., Wang, Y., Zhan, K., Lang, X., Bao, H., Zhou, X., et al.: Streetcrafter: Street view synthesis with controllable video diffusion models. In: CVPR. pp. 822–832 (2025)
51. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arxiv:2308.06721 (2023)
52. Yu, H.X., Duan, H., Herrmann, C., Freeman, W.T., Wu, J.: Wonderworld: Interactive 3d scene generation from a single image. In: CVPR. pp. 5916–5926 (2025)
53. Yu, H.X., Duan, H., Hur, J., Sargent, K., Rubinstein, M., Freeman, W.T., Cole, F., Sun, D., Snavely, N., Wu, J., et al.: Wonderjourney: Going from anywhere to everywhere. In: CVPR. pp. 6658–6667 (2024)
54. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. TPAMI (2025)
55. Zhang, B., Cheng, Y., Yang, J., Wang, C., Zhao, F., Tang, Y., Chen, D., Guo, B.: Gaussiancube: A structured and explicit radiance representation for 3d generative modeling. In: NeurIPS (2024)
56. Zhang, L., Wang, Z., Zhang, Q., Qiu, Q., Pang, A., Jiang, H., Yang, W., Xu, L., Yu, J.: Clay: A controllable large-scale generative model for creating high-quality 3d assets. ACM TOG **43**(4), 1–20 (2024)
57. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023)
58. Zhang, Q., Zhai, S., Martin, M.A.B., Miao, K., Toshev, A., Susskind, J., Gu, J.: World-consistent video diffusion with explicit 3d modeling. In: CVPR. pp. 21685–21695 (2025)

59. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., et al.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. arXiv preprint arXiv:2501.12202 (2025)
60. Zheng, K., Zhang, R., Gu, J., Yang, J., Wang, X.E.: Constructing a 3d town from a single image. arXiv preprint arXiv:2505.15765 (2025)