InSpace: Structure-Aware 3D Indoor Scene Generation from a Single 360° Image

Gwanhyeong Koo[*1,2], Hyunsu Kim[2], Youngji Kim[2], Taejae Lee[2], Siwoo Lim[1], Sunjae Yoon[3], Suyong Yeon[†2], and Chang D. Yoo[†1]

[1]Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
{kookie, siu4450, cd_yoo}@kaist.ac.kr
[2]NAVER LABS, Seongnam-si, Republic of Korea
{hyunsu.xr, youngji.b.kim, taejae.lee, suyong.yeon}@naverlabs.com
[3]Chung-Ang University, Seoul, Republic of Korea
{sunjaeyoon}@cau.ac.kr

Fig. 1: InSpace generates complete 3D indoor scenes with structural layout and individual assets from a single 360° image. Leveraging the full 360° context of panoramic images, it produces spatially coherent scenes across diverse room layouts.

Abstract. Recent advances in single image-to-3D generation have enabled high-quality asset synthesis, yet extending these capabilities to indoor scene generation remains challenging. Existing methods focus on asset-level generation while neglecting the structural layout, which is essential for downstream applications and serves as the spatial anchor for grounding assets. However, a single image with a limited field of view lacks the spatial coverage to recover a coherent global layout. To this end, we use a 360° image represented in equirectangular projection (ERP) and propose **InSpace**, a structure-aware framework for 3D indoor scene generation. InSpace comprises three stages: (1) estimating partial scene geometry as spatial priors, (2) generating coarse scene

[*]Work done during an internship at NAVER LABS. [†]Co-corresponding authors

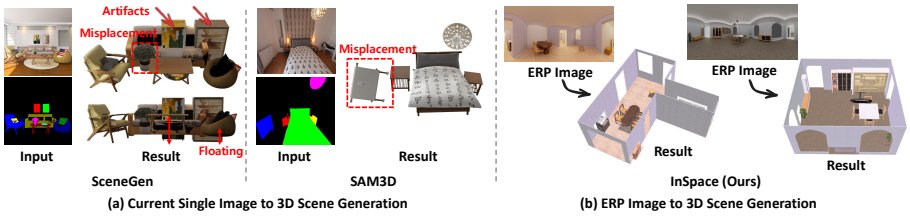

Fig. 2: (a) Existing single-image methods generate individual assets without structural layout, causing floating, misplacement, and artifacts. (b) InSpace uses an ERP image to generate complete indoor scenes with structural layout and well-grounded assets.

structure with view-selective cross-attention, and (3) producing detailed layout and asset geometry with textures through a global-local hybrid attention, using flow matching. We also propose ERP-FRONT, a paired ERP-Image-to-3D indoor scene dataset based on 3D-FRONT. Experiments show that InSpace generates complete 3D indoor scenes with structural layout, along with separate textured assets from a single ERP image, achieving strong performance across 3D and 2D metrics.

Keywords: 3D Scene Generation · ERP Imaging · Generative Models

1 Introduction

Recent advances in generative modeling have dramatically accelerated 3D content creation from visual inputs. In particular, single-image-to-3D generation models [6, 18, 20, 31, 34–36, 41, 54–57, 61, 65–67] have achieved remarkable quality and geometric fidelity, enabling rapid synthesis of an object from a single image. This success has naturally motivated extending these capabilities beyond isolated objects to entire 3D scene synthesis. Among these, indoor scene generation [15, 39, 46, 58, 60] has attracted growing attention due to its broad practical impact, from synthetic training environments for robotics and simulation, to digital twin creation for real estate and spatial planning, to enabling embodied agents to perceive and navigate 3D environments from a single observation.

However, existing methods focus on asset-level generation, producing individual assets and estimating their placement, while neglecting the structural layout of the indoor scene, including floors, walls, and built-in structures. This structural layout is a fundamental component of indoor scene generation. First, it reveals the overall shape and extent of the space, which is essential for any downstream task that requires scene-level spatial understanding. Second, it serves as a spatial anchor that constrains every object in the scene. Furniture rests on the floor, shelves lean against walls, and the structural geometry determines permissible placements. Without it, generated assets often exhibit spatial misalignment [19,46], floating above the ground or interpenetrating walls (Fig. 2(a)), necessitating post-hoc physics-aware optimization [39, 58, 60].

To address this limitation, we reformulate the problem by expanding the input modality from a single image to a 360° image represented in equirectangular projection (ERP). A single image provides only a limited field of view, making full structural layout estimation fundamentally ill-posed. In contrast, an ERP image captures the complete 360° field of view of an indoor environment [1, 2], offering comprehensive information about the scene geometry. Leveraging this richer spatial context, we introduce **InSpace**, a structure-aware framework for 3D indoor scenes, including both structural layout and individual assets from a single ERP image (Fig. 2 (b)).

InSpace comprises three cascaded stages. First, given a single ERP image, we estimate a depth map [33] and lift it to 3D via equirectangular back-projection, producing an initial point cloud. This is then normalized into $[-0.5, 0.5]^3$, termed the Partial Scene Geometry (PSG), along with a calibrated camera center. Both serve as geometric priors for spatially grounding the subsequent generation stages. Second, we convert the ERP image into six cubemap face images and use them as conditioning inputs to a flow-matching model [55] that generates a coarse 3D voxel representation of the scene, termed the Coarse Scene Geometry (CSG). Unlike conventional 3D generation models, where every voxel latent is conditioned globally by a single image feature, our model introduces **view-selective cross-attention**, where each voxel in the latent is conditioned only on the cubemap faces visible from its 3D position based on the calibrated camera center. This spatially grounded conditioning enables the model to coherently compose the scene from all six cubemap views with fine-grained spatial control. From the resulting CSG, a lightweight 3D bounding box detector built on a 3D U-Net [43, 62] predicts 3D oriented bounding boxes (3D OBBs) for each asset. Third, conditioned on the predicted 3D bounding boxes, we generate detailed geometry and texture for both the structural layout and each individual asset. This is achieved through a global-local hybrid attention mechanism. Global self-attention enables all scene components to interact for spatial coherence, while **asset-selective cross-attention** allows each component to attend only to its corresponding image region for fine-grained detail recovery. The resulting latents are finally decoded into a textured 3D mesh of the complete indoor scene.

To train and evaluate InSpace, we require paired ERP images and corresponding 3D indoor scene meshes. However, such data is unavailable in existing benchmarks. While several real-world panoramic datasets [5, 12, 53, 68] provide ERP images of indoor scenes, their reconstructed meshes are often incomplete or lack the structural fidelity necessary. To address this gap, we construct **ERP-FRONT**, a synthetic ERP-Image-to-3D indoor scene dataset based on 3D-FRONT [13]. Each indoor scene is defined at the room level, covering a wide range of room sizes, and paired with ERP observations rendered from within the scene. The resulting dataset contains 26.5K training and 2.5K test ERP-image-mesh pairs. In our experiments, InSpace shows strong performance on both 3D geometric and 2D rendering-based metrics. Moreover, as shown in the supplementary material, InSpace generalizes well to realistic panoramic datasets such as ReplicaPano [12], producing spatially coherent 3D indoor environments.

2 Related Work

3D Asset Generation Following the success of latent diffusion in 2D [29, 50], recent 3D asset generation methods adopt VAEs [23, 48] to encode 3D geometry into latent spaces, where diffusion [47] or flow-based models [40] learn to generate. Depending on the VAE design, existing approaches fall into two categories: vector-set (VecSet) and sparse voxel representations. **VecSet representations**, introduced in 3DShape2VecSet [64], encode 3D shapes as unordered latent token sets derived from surface point sampling. A set of learnable queries [4, 21] interacts with point embeddings via cross-attention to form an orthogonal basis of the shape latent space, capturing relative geometric relationships rather than absolute spatial coordinates. This design is efficient and transformer-friendly, leading to wide adoption in large-scale generation models [20, 31, 34–36, 65–67]. However, since tokens are not anchored to fixed spatial locations, VecSet lacks explicit spatial locality, making position-aware control and structured test-time scaling challenging. By contrast, **sparse voxel representations** encode 3D shapes as structured grids with explicit spatial coordinates. Introduced in XCube [49], sparse voxel VAEs [17, 37] compress 3D occupancy or feature grids via 3D convolutional encoders, maintaining spatially organized latent tensors aligned with 3D coordinates rather than learning an abstract token basis. This structured design enables locality for position-aware conditioning and local editing, and has been adopted in high-fidelity models [6, 32, 54–56, 61]. However, sparse voxel methods are more computationally expensive and tightly coupled to grid resolution, limiting scalability compared to VecSet-based approaches. Recent work [7, 22, 30] explores hybrid representations combining structured voxel anchors with VecSet-based latents to balance efficiency and spatial locality.

3D Scene Generation Early methods [14, 16, 27, 28] relied on retrieving and aligning existing 3D models to the input image, but were limited by dataset diversity and sparse geometric cues. The advent of image-to-3D models enabled a new paradigm of generating objects individually and assembling them via language/vision-guided spatial reasoning [15, 39, 58], and physics-based optimization [39, 58, 60] proposed to improve placement. However, these multi-stage pipelines are susceptible to error accumulation and require costly optimization for object placement. Recent simultaneous multi-instance generation methods [19, 38, 46] improve inter-object coherence through cross-instance interaction, yet remain focused on asset-level synthesis without modeling structural layouts (e.g., floors, walls, ceilings), causing spatial misalignment such as floating or interpenetrating objects. HiScene [11] addresses this by converting inputs into isometric views and generating compositional structures to ground objects on architectural surfaces, but the resulting layouts remain approximate and do not capture precise indoor geometry. Most related to our work, PanoContext-Former [12] is the first attempt at ERP-Image-to-indoor-scene generation, but produces inaccurate geometry, lacks texture support, and its code is unavailable.

3 Preliminary

Our method requires explicit spatial control over 3D voxel latents using ERP image features, necessitating a spatially structured latent space rather than an unordered VecSet representation. We therefore build upon TRELLIS.2 [55], which adopts the O-Voxel formulation to encode geometry and appearance within a compact, sparse voxel latent space.

O-Voxel Representation. O-Voxel encodes both geometry and appearance as a set of feature tuples associated with active voxels on a regular N^3 grid:

$$\mathbf{f} = \{(\mathbf{f}_i^{\text{shape}}, \mathbf{f}_i^{\text{mat}}, \mathbf{p}_i)\}_{i=1}^{L}, \tag{1}$$

where $\mathbf{f}_i^{\text{shape}}$ encodes local geometry via a Flexible Dual Grid formulation, $\mathbf{f}_i^{\text{mat}}$ encodes PBR material properties with structural layout and textured assets (*i.e.*, base color, metallic, roughness, and opacity), $\mathbf{p}_i \in \{0, \dots, N-1\}^3$ is the voxel coordinate, and $L \ll N^3$ is the number of active (surface-intersecting) voxels.

Sparse Compression VAE. To obtain a compact latent space, the framework trains two decoupled Sparse Compression VAEs (SC-VAEs) $\{\mathcal{E}_{\text{shape}}, \mathcal{D}_{\text{shape}}\}$ and $\{\mathcal{E}_{\text{mat}}, \mathcal{D}_{\text{mat}}\}$ for shape and material, respectively. Both are fully sparse-convolutional networks with a spatial downsampling factor s, compressing each active voxel's features into a latent vector of $c_{sc} = 32$ channels while preserving the sparse coordinate structure at the downsampled resolution $\frac{N}{s}$.

Three-Stage Generation Pipeline. In TRELLIS.2, given a conditioning image $\mathbf{I}$, 3D assets are generated through three sequential stages, each using a DiT-based flow matching model [40, 47] that denoises Gaussian noise $\boldsymbol{\epsilon}$ over timestep t:

$$\textit{Sparse Structure: } \mathcal{F}_{\text{ss}} : (\boldsymbol{\epsilon}, t, \mathbf{I}) \to \mathbf{z}_{\text{ss}}; \quad \mathbf{p} = \mathcal{D}_{\text{ss}}(\mathbf{z}_{\text{ss}}), \tag{2}$$

$$\textit{Geometry: } \mathcal{F}_{\text{shape}} : (\boldsymbol{\epsilon}_{\text{shape}}, t, \mathbf{I}) \to \mathbf{z}_{\text{shape}}; \quad \mathbf{f}^{\text{shape}} = \mathcal{D}_{\text{shape}}(\mathbf{z}_{\text{shape}}), \tag{3}$$

$$\textit{Material: } \mathcal{F}_{\text{mat}} : (\boldsymbol{\epsilon}_{\text{mat}}, t, \mathbf{I}, \mathbf{z}_{\text{shape}}) \to \mathbf{z}_{\text{mat}}; \quad \mathbf{f}^{\text{mat}} = \mathcal{D}_{\text{mat}}(\mathbf{z}_{\text{mat}}). \tag{4}$$

In the *sparse structure generation* stage (Eq. 2), DiT $\mathcal{F}_{\text{ss}}$ generates a structure latent $\mathbf{z}_{\text{ss}} \in \mathbb{R}^{c_0 \times \frac{N_0}{s_0}^3}$ from noise $\boldsymbol{\epsilon}$, where c_0 is the latent channel dimension, N_0 is the grid resolution, and s_0 is the downsampling factor (*e.g.*, $c_0 = 8$, $N_0 = 64$, $s_0 = 4$). The decoder $\mathcal{D}_{\text{ss}}$ maps $\mathbf{z}_{\text{ss}}$ into a coarse occupancy grid of resolution N_0^3 to determine the set of active voxel coordinates $\mathbf{p}$. In the *geometry generation* stage (Eq. 3), structured noise $\boldsymbol{\epsilon}_{\text{shape}} = \{(\boldsymbol{\epsilon}_i, \mathbf{p}_i)\}_{i=1}^{L}$ is formed as a SparseTensor by assigning random Gaussian vectors $\boldsymbol{\epsilon}_i \in \mathbb{R}^{c_{sc}}$ to each active coordinate, and a sparse DiT $\mathcal{F}_{\text{shape}}$ denoises it into the geometry latent $\mathbf{z}_{\text{shape}}$, where voxel coordinates $\{p_i\}_{i=1}^{L}$ are injected via positional embeddings. The *material generation* stage (Eq. 4) follows the same sparse formulation but additionally conditions on $\mathbf{z}_{\text{shape}}$ via channel-wise concatenation. The decoded O-Voxel is then instantly converted into a textured mesh.

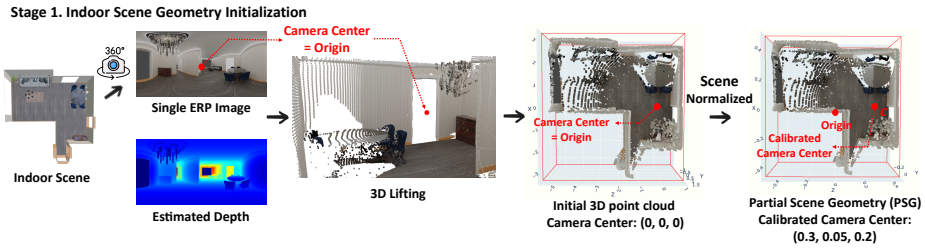

Fig. 3: Indoor Scene Geometry Initialization. From a single ERP image and its estimated depth, we lift the scene to 3D via equirectangular back-projection, producing an initial point cloud with the camera center at the origin $(0, 0, 0)$. The point cloud is then normalized into the canonical space $[-0.5, 0.5]^3$, yielding the Partial Scene Geometry (PSG) and shifting the camera center to a calibrated position $\mathbf{c}$.

4 Method

Given a single ERP image, InSpace generates a complete 3D indoor scene comprising both the scene layout (floors and walls) and individual assets with detailed geometry and texture. To enable spatial control over voxel latents conditioned on ERP features, we adopt a sparse voxel-based framework [55] and further decompose the scene into layout and asset-level components inspired by part-aware generation [10, 59]. Our framework consists of three stages. First, we estimate the scene layout by lifting the ERP depth map to a 3D point cloud and calibrating the camera center within a normalized voxel space (Sec. 4.1). Second, we generate a coarse scene structure via flow-based denoising in a dense voxel latent space, conditioned on cubemap features through view-selective cross-attention, and detect 3D oriented bounding boxes (OBBs) of assets from the generated structure (Sec. 4.2). Third, we produce detailed geometry and texture through a hybrid attention mechanism combining global self-attention with OBB-projected asset-selective cross-attention. Each asset attends to its relevant image region while preserving scene-level coherence (Sec. 4.3).

4.1 Stage 1. Indoor Scene Geometry Initialization

We begin by constructing the geometric foundation of the scene from a single ERP image. Using a pretrained depth estimator [33], we obtain a dense depth map and lift it to 3D via equirectangular projection, producing an initial point cloud. Since the ERP image is captured from a single viewpoint, the camera center (the position from which the panorama was rendered) lies at the origin $(0, 0, 0)$. To prepare for voxel-based generation, we normalize the point cloud into the canonical space $[-0.5, 0.5]^3$ via translation and uniform scaling. This normalization shifts the camera center from the origin to a new position $\mathbf{c}$ within the canonical space (see Fig. 3). We refer to the result as the *Partial Scene Geometry (PSG)*, as it captures only the surfaces visible from the single viewpoint. The calibrated camera center $\mathbf{c}$ and PSG are passed to subsequent stages as geometric priors for spatially grounding the generation process.

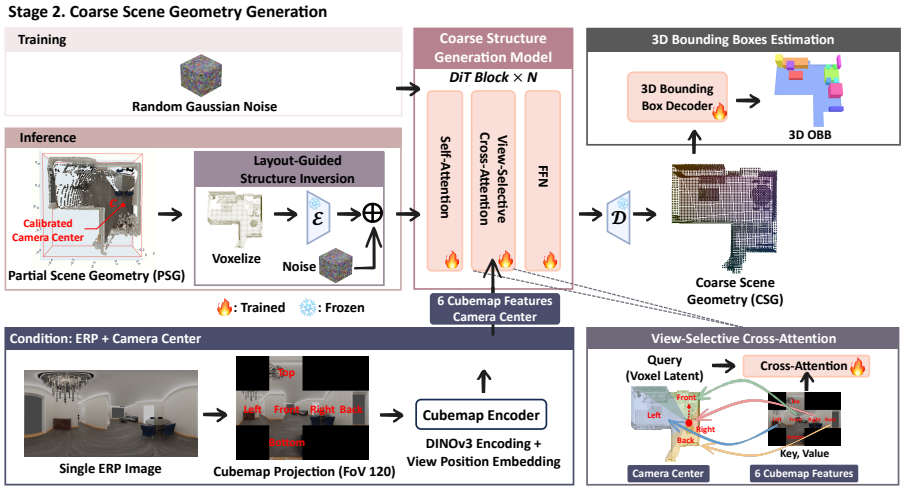

Fig. 4: Coarse Scene Geometry Generation. Six cubemap faces (FoV 120) are extracted from the ERP image and encoded with view position embeddings. The model denoises voxel latents via DiT blocks with view-selective cross-attention, where each voxel attends only to its visible faces based on camera center $\mathbf{c}$. At inference, Layout-Guided Structure Inversion optionally initializes the latent from the PSG. The decoded coarse geometry (CSG) is fed to a 3D bounding box detector to predict OBBs for assets.

4.2 Stage 2. Coarse Scene Geometry Generation

In this stage, we generate the coarse structure of the indoor scene as a dense voxel latent $\mathbf{z}_{\text{ss}} \in \mathbb{R}^{c_0 \times (N_0/s_0)^3}$ (Eq. 2) using a flow matching model [55]. The model consists of stacked DiT blocks, each applying self-attention among voxel latents, followed by view-selective cross-attention with the six cubemap features, and a feed-forward network (Fig. 4). It is trained with rectified flow matching (Sec. 4.4) to predict the velocity field along the path connecting Gaussian noise and the target voxel latent.

Cubemap Encoding. To condition the model on the input ERP image, we extract six cubemap faces along the six orthogonal viewing directions, with a field of view (FoV) set to 120 to introduce overlap between adjacent faces and prevent objects near boundaries from being clipped. Each face is independently encoded by a DINOv3 [52] image encoder, producing T_f tokens of dimension D_c per face, where f indexes the six cubemap faces. To distinguish face directions, we add a learnable view position embedding $\mathbf{e}_v \in \mathbb{R}^{6 \times D_c}$ to each face's features. The six encoded faces are concatenated along the token dimension, yielding a unified condition tensor $\mathbf{F} \in \mathbb{R}^{B \times T \times D_c}$, where B is the batch size and $T = 6T_f$.

View-Selective Cross-Attention. In existing 3D asset generation [20, 55], a single condition image influences all voxels in the latent uniformly during denoising. In our panoramic setting, however, the six cubemap faces each correspond

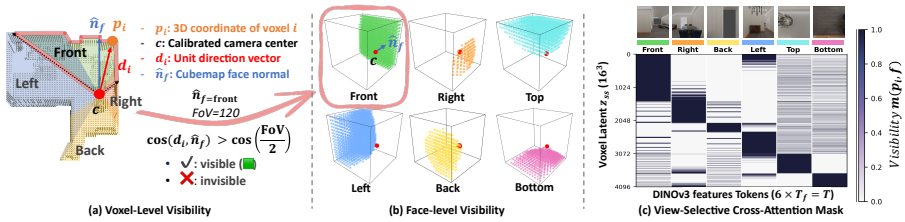

Fig. 5: View-Selective Cross-Attention. (a) For each voxel at position p_i, we compute the cosine similarity between the direction d_i from camera center c and each cubemap face normal $\hat{n}_f$ to determine visibility. (b) Voxels are partitioned by their visible faces, shown in corresponding colors. (c) The resulting mask M_{vs} assigns each voxel high attention weights (dark) only to its visible cubemap tokens.

to a distinct spatial region, and voxels in different regions of $\mathbf{z}_{\text{ss}}$ (Eq. 2) should be conditioned primarily on their corresponding faces rather than all condition tokens. To this end, we construct a view-selective (VS) cross-attention mask $\mathbf{M}_{\text{vs}}$ (Eq. 6) that establishes this spatially grounded correspondence.

For the i-th voxel at coordinate $\mathbf{p}_i \in \{0, \ldots, N_0/s_0 - 1\}^3$ (Eq. 1), we compute the unit direction $\mathbf{d}_i = (\mathbf{p}_i - \mathbf{c})/\|\mathbf{p}_i - \mathbf{c}\|$ from the calibrated camera center $\mathbf{c}$ toward the voxel, and measure how well this direction aligns with each cubemap face f, whose outward normal is $\hat{\mathbf{n}}_f$ (e.g., $\hat{\mathbf{n}}_{\text{front}} = (0, 1, 0)$). The cosine similarity $\mathbf{d}_i \cdot \hat{\mathbf{n}}_f$ is compared against $\cos(\text{FoV}/2)$, the angular boundary of each face's visible cone. To produce a smooth, differentiable mask rather than a hard cutoff at face boundaries, we pass this comparison through a scaled sigmoid:

$$m(\mathbf{p}_i, f; \mathbf{c}) = \sigma\big(\alpha\big(\mathbf{d}_i \cdot \hat{\mathbf{n}}_f - \cos\big(\tfrac{\text{FoV}}{2}\big)\big)\big) \in (0, 1], \tag{5}$$

where α is a fixed scaling factor that yields a near-binary transition. The output m represents visibility (see Fig. 5(a)), where values near 1 indicate the voxel falls well within the face's FoV and values near 0 indicate it lies outside. This soft assignment effectively partitions $\mathbf{z}_{\text{ss}}$ into spatially coherent groups, each primarily conditioned on its corresponding cubemap face, while allowing smooth transitions at face boundaries due to the overlapping FoV. The face-level visibility is broadcast to all T_f tokens within each face to form the full mask (Fig. 5(b,c)):

$$\mathbf{M}_{\text{vs}}[i, j] = m(\mathbf{p}_i,\ f(j);\ \mathbf{c}), \quad \mathbf{M}_{\text{vs}} \in \mathbb{R}^{(N_0/s_0)^3 \times T}, \tag{6}$$

where i indexes voxel latents, j indexes condition tokens, and $f(j)$ maps the j-th token to its cubemap face. Each row encodes a single voxel's visibility across all T tokens, and each column reflects the visibility of its corresponding cubemap face. This mask is applied as an additive bias to the cross-attention logits:

$$\text{VS-CrossAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}; \mathbf{M}_{\text{vs}}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}} + \log \mathbf{M}_{\text{vs}}\right)\mathbf{V}. \tag{7}$$

where d_h is the attention head dimension, $\mathbf{Q} \in \mathbb{R}^{(N_0/s_0)^3 \times d_h}$ are voxel queries projected from $\mathbf{z}_{\text{ss}}$ (Eq. 2), and $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{T \times d_h}$ are keys and values projected

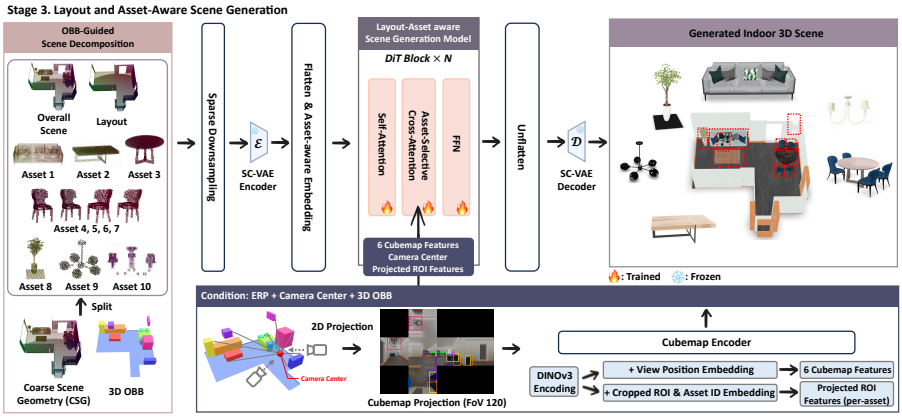

Fig. 6: Layout and Asset-Aware Scene Generation. The CSG is decomposed into layout and assets via detected OBBs, encoded with part embeddings, and concatenated into a unified token sequence. DiT blocks apply global self-attention, followed by asset-selective cross-attention where layout tokens use the view-selective mask and each asset attends to its projected ROI features. The output is decoded into a textured 3D scene.

from $\mathbf{F}$. Since $m \in (0, 1]$, the $\log \mathbf{M}_{\text{vs}}$ term acts as a soft penalty, where near-zero visibility yields large negative values that suppress attention weights, while full visibility contributes zero and leaves them unchanged.

Layout-Guided Structure Inversion. At inference, we leverage the PSG from Sec. 4.1 as a spatial prior. This is inspired by SDEdit [45] and other diffusion-based image editing methods [8, 25, 26, 44, 51, 63], which add noise to a guidance input and denoise from that intermediate noise level rather than from pure noise to keep the output faithful to the input. We voxelize the PSG and encode it via $\mathcal{E}_{\text{ss}}$ into $\mathbf{z}_{\text{psg}}$, then construct the noised initial latent $\mathbf{x}_{t_0} = (1 - t_0)\,\mathbf{z}_{\text{psg}} + t_0\,\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and denoise from t_0 to 0. A lower t_0 preserves more of the PSG layout, while a higher t_0 allows greater model freedom. This requires no change to training, as the model is optimized with the standard flow-matching objective over $t \sim \mathcal{U}(0, 1)$ and thus learns to denoise the target latent from any noise level, allowing inference to begin from the noised PSG at $t_0 < 1$ instead of pure noise ($t_0 = 1$). Since the PSG already encodes the geometry recovered from the input ERP, initializing from this information-rich state provides a strong spatial anchor and yields more accurate scene structure.

3D Bounding Box Detection. After denoising by the stacked DiT blocks, we obtain the generated structure latent $\mathbf{z}_{\text{ss}}$ and decode it via $\mathcal{D}_{\text{ss}}$ (Eq. 2) into the *Coarse Scene Geometry (CSG)*, a binary occupancy grid at resolution N_0^3. A 3D U-Net–based detector [43, 62] then predicts K OBBs $\{\mathbf{o}_k\}_{k=1}^{K}$ in a single pass, each parameterized by its center $\mathbf{o}_k^c \in [-0.5, 0.5]^3$, extents $\mathbf{o}_k^s \in [0, 0.5]^3$, and yaw angle $\theta_k \in [-\pi, \pi]$. Architectural details are in Sec. 2 of the supplementary.

4.3 Stage 3. Layout and Asset-Aware Scene Generation

The final stage generates detailed geometry and texture for the complete indoor scene (Eqs. 3–4). Inspired by part-aware 3D generation [59], we decompose the scene into layout and asset-level components and process them jointly through a hybrid attention mechanism. Shape and texture share an identical model architecture, where the texture model additionally conditions on the generated shape latent $\mathbf{z}_{\text{shape}}$ via concatenation (Eq. 4).

OBB-Guided Scene Decomposition. Using the detected OBBs from Sec. 4.2, we decompose the CSG into individual components (Fig. 6, left): the overall scene (all active voxels), the scene layout (voxels outside any OBB, representing floors and walls), and K individual assets (voxels enclosed within each $\mathbf{o}_k$). Each component is encoded by the SC-VAE encoder $\mathcal{E}$ (Sec. 3), flattened into a token sequence, and summed with a learnable part positional embedding $\mathbf{e}_p \in \mathbb{R}^D$ (DiT hidden dim) encoding its component identity. All sequences are concatenated into a unified token sequence and processed by stacked DiT blocks, where each block applies global self-attention followed by asset-selective cross-attention.

Global Scene Self-Attention. Within each DiT block, self-attention is first applied across all components in the unified token sequence, enabling the model to learn inter-object spatial relationships and physical placement constraints [59].

Asset-Selective Cross-Attention. While self-attention operates globally, cross-attention is scoped per component, directing each to its relevant image region. Since the view-selective mask (Eq. 5) depends only on the calibrated camera center $\mathbf{c}$ and voxel coordinates, it generalizes to any voxel latent beyond $\mathbf{z}_{\text{ss}}$. We leverage this to construct a unified cross-attention mask $\mathbf{M}_{\text{as}} \in \mathbb{R}^{N_{\text{total}} \times T}$, where N_{total} is the total number of voxels across all components and T is the number of cubemap condition tokens (Sec. 4.2), combining two masking strategies:

$$\mathbf{M}_{\text{as}}[i,j] = \begin{cases} \mathbf{M}_{\text{vs}}[i,j] \in \mathbb{R}^{(N_{\text{ov}}+N_{\text{la}})\times T} & \text{if voxel } i \in \text{overall or layout,} \\ \mathbf{M}_{\text{as}}^{(k)}[i,j] \in \{0,1\}^{N_k \times T} & \text{if voxel } i \in \text{asset } k, \end{cases} \tag{8}$$

where N_{ov}, N_{la}, and N_k denote the number of voxels in the overall scene, layout, and asset k respectively. The full mask $\mathbf{M}_{\text{as}}$ is formed by stacking $\mathbf{M}_{\text{vs}}$ for the overall and layout components with $\mathbf{M}_{\text{as}}^{(k)}$ for each asset along the voxel dimension, yielding $N_{\text{total}} = N_{\text{ov}} + N_{\text{la}} + \sum_{k=1}^{K} N_k$. For the overall scene and layout components, $\mathbf{M}_{\text{vs}}[i,j]$ reuses the view-selective visibility $m(\mathbf{p}_i, f(j); \mathbf{c})$ (Eq. 5) applied to the voxel coordinates of $\mathbf{z}_{\text{shape}}$ and $\mathbf{z}_{\text{mat}}$ (Eqs. 3–4), so that each voxel attends to the cubemap faces visible from its 3D position. For each asset k, $\mathbf{M}_{\text{as}}^{(k)}$ is defined via $m_{\text{proj}}(\mathbf{o}_k, j; \mathbf{c}) \in \{0,1\}$, obtained by projecting the eight corners of $\mathbf{o}_k$ onto the six cubemap planes using $\mathbf{c}$ and the FoV geometry (Fig. 6, bottom). The resulting 2D bounding rectangles crop the corresponding

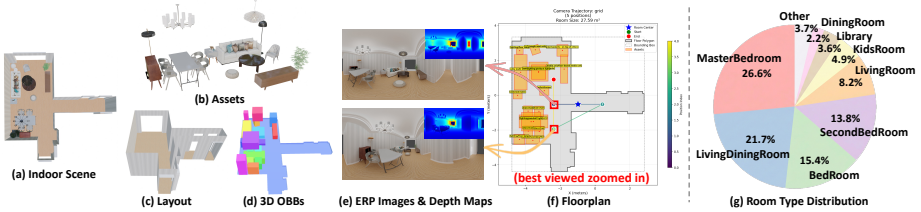

Fig. 7: ERP-FRONT Dataset. Each room from 3D-FRONT is decomposed into (a) the complete indoor scene, (b) individual assets, (c) structural layout, and (d) 3D oriented bounding boxes. (e) ERP images with depth maps are rendered from valid camera positions in (f) the floorplan. (g) Room type distribution across the dataset.

regions from $\mathbf{F}$, setting $m_{\text{proj}} = 1$ for tokens within the region and 0 otherwise. An asset identity embedding is added to the cropped tokens, so that when M_{as}^{k} is stacked into the unified M_{as}, the model can identify the asset associated with each token. Since 3D OBB projection depends only on $\mathbf{o}_k$, this mask is shared across all voxels within asset k. The mask M_{as} is added to the attention logits:

$$\text{AS-CrossAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}; \mathbf{M}_{\text{as}}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_h}} + \log \mathbf{M}_{\text{as}}\right)\mathbf{V}. \tag{9}$$

where $\mathbf{Q} \in \mathbb{R}^{N_{\text{total}} \times d_h}$ are voxel queries from the unified token sequence, and $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{T \times d_h}$ are keys and values projected from $\mathbf{F}$. This enables layout components to attend to their geometrically visible cubemap faces via $\mathbf{M}_{\text{vs}}$, while each asset k attends exclusively to its OBB-projected image region via $\mathbf{M}_{\text{as}}^{(k)}$, recovering fine-grained appearance details. The outputs are unflattened and decoded by $\mathcal{D}$ to produce the final textured 3D mesh.

4.4 Training Objective

All three models $\mathcal{F}_{\text{ss}}$, $\mathcal{F}_{\text{shape}}$, and $\mathcal{F}_{\text{mat}}$ (Eqs. 2–4) are trained with rectified flow matching [42]. Given a ground-truth latent $\mathbf{x}_0$ and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we construct the interpolant $\mathbf{x}_t = (1-t)\,\mathbf{x}_0 + t\,\boldsymbol{\epsilon}$ at $t \sim \mathcal{U}(0,1)$ and train the model to predict the velocity field: $\mathcal{L} = \mathbb{E}_{t,\,\mathbf{x}_0,\,\boldsymbol{\epsilon}}\Big[\big\| \mathbf{v}_\theta(\mathbf{x}_t,\, t,\, \mathbf{F},\, \mathbf{M}) - (\boldsymbol{\epsilon} - \mathbf{x}_0)\big\|^2\Big]$ where $\mathbf{F}$ denotes the cubemap condition features and $\mathbf{M}$ is the stage-specific attention mask: $\mathbf{M}_{\text{vs}}$ (Eq. 6) for $\mathcal{F}_{\text{ss}}$, and $\mathbf{M}_{\text{as}}$ (Eq. 8) for $\mathcal{F}_{\text{shape}}$ and $\mathcal{F}_{\text{mat}}$. For material generation, $\mathbf{v}_\theta$ additionally conditions on the generated shape latent $\mathbf{z}_{\text{shape}}$ via concatenation (Eq. 4).

5 ERP-FRONT Dataset

We construct ERP-FRONT, a paired ERP-Image-to-3D indoor scene dataset built on 3D-FRONT [13], rendered using BlenderProc [9]. Each house is decomposed at the room level (one or more rooms per house), and for each room we

extract the structural layout (floors and walls), individual assets with their 3D OBBs, and a floorplan encoding asset positions (Fig.7(a)-(f)). To render ERP images, we search for valid camera positions on a uniform grid within each room, requiring a minimum clearance of 30 cm from all assets to reduce occlusion. The resulting dataset comprises 5,962 houses and 13,292 rooms, totaling approximately 29K ERP-Image-mesh pairs. The average room area is 19.8 m^2 with 8.4 assets per room. We split the data into 12,121 training rooms (26.5K ERP-image-mesh pairs) and 1,144 test rooms (2.5K pairs). Room type distribution is shown in Fig. 7(g). More details are provided in Sec. 1 of the supplementary.

6 Experiments

6.1 Implementation Details

Our framework is built upon TRELLIS.2 [55], using all VAE encoders and decoders with pretrained weights without finetuning. For sparse structure generation (Eq. 2), we set $c_0 = 8$, $N_0 = 64$, $s_0 = 4$, yielding a latent resolution of 16^3. For shape and texture generation (Eqs. 3–4), the SC-VAE uses $c_{\text{sc}} = 32$ channels with downsampling factor $s = 16$ on an $N = 512$ grid, producing sparse latents at 32^3 resolution. All stages use a frozen DINOv3 ViT-L/16 [52] encoder (512×512 input, $T_f = 1029$ tokens, $D_c = 1024$). The scaling factor in Eq. 5 is set to α=50. Each generation model has 30 DiT blocks ($D = 1536$, 12 heads, $d_h = 128$) finetuned from pretrained weights with AdamW (lr 5×10^{-5}). $\mathcal{F}_{\text{ss}}$, $\mathcal{F}_{\text{shape}}$, and $\mathcal{F}_{\text{mat}}$ are trained for 120, 60, and 24 epochs respectively on 6× NVIDIA A100 GPUs with batch size 4 per GPU. The 3D bounding box detector is a lightweight 3D U-Net [62] (12.5M parameters) trained on the decoded CSG.

6.2 Evaluation Setup

Model Configuration. Our default model is trained with view-selective cross-attention (Stage 2) and asset-selective cross-attention (Stage 3), with Layout-Guided Structure Inversion at $t_0 = 0.7$ during inference.

Compared Methods. While a few prior works have explored panoramic 3D scene understanding [12], their code is unavailable, making direct comparison infeasible. We therefore compare against recent single-image scene generation methods: SceneGen [46], MIDI [19], and SAM3D [6]. For each test scene, we extract the perspective view that maximizes asset visibility from the ERP image by projecting 3D bounding boxes onto candidate views. This image, along with its segmentation mask [24], is provided as input. As these methods operate in their own coordinate systems, we manually align the generated scenes to the ground truth before refining with ICP [3]. Comparisons with existing methods use 20 samples due to manual alignment. Quantitative and qualitative comparisons with these methods are provided in Sec. 4 of the supplementary.

Metrics. For 3D evaluation, we measure Voxel IoU, Chamfer Distance (CD), and F1-Score on the coarse structure (Stage 2, Fig. 9(a)) at the scene level, as

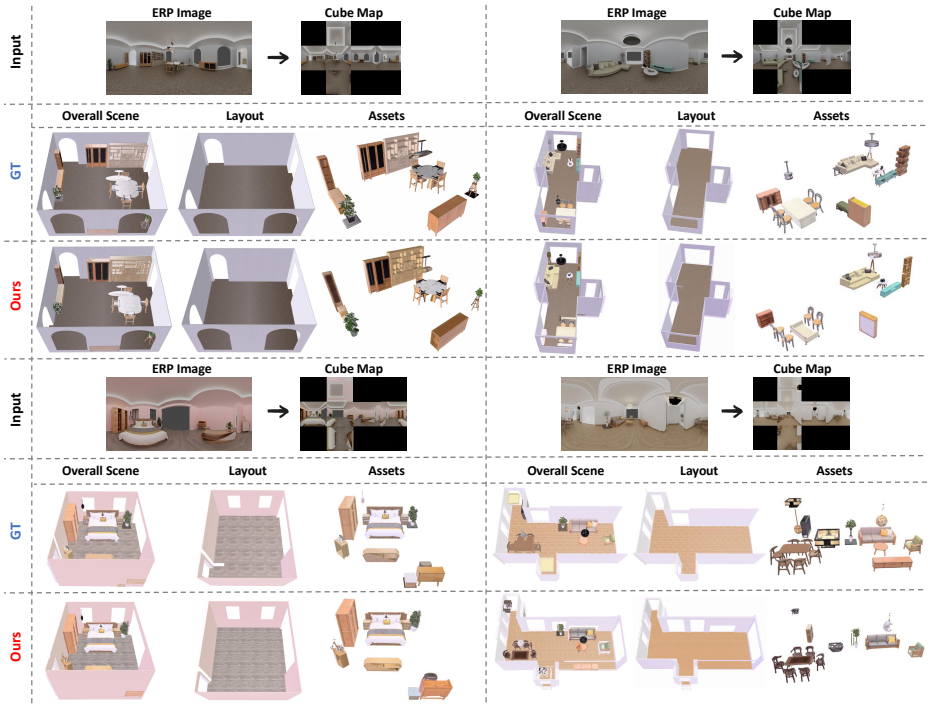

Fig. 8: Qualitative results on ERP-FRONT. Given an ERP image and its cubemap decomposition (top), InSpace generates the overall scene, structural layout, and individual assets. Four examples are shown with ground truth for comparison.

this stage captures overall room geometry before asset decomposition. On the generated meshes (Stage 3), we evaluate CD and F1-Score at both scene and asset level, where assets are matched to ground truth via estimated and GT bounding box alignment. For 2D evaluation, we render Stage 3 outputs from a top-down orthographic view and six interior views along cubemap directions (Fig. 9(b)), using PSNR and LPIPS.

6.3 Results

For qualitative results, Fig. 1 and 8 show generated scenes on ERP-FRONT, where InSpace produces complete scenes whose layout and assets match the ground truth. To assess generalization beyond the synthetic ERP-FRONT, we test on ReplicaPano [12], which provides ERP images extracted from the realistic Replica dataset [53]. Additional results on ERP-FRONT and ReplicaPano, along with comparisons against single-image scene generation methods, are provided in Sec. 3–4 of the supplementary. For quantitative results, Table 1 shows performance under different configurations. $t_0 = 0.5$ achieves the best IoU in Stage 2, while in Stage 3, $t_0 = 0.7$ performs marginally better across most metrics.

Table 1: Quantitative results on ERP-FRONT test set. (Left) Coarse structure generation (Stage 2) under different configurations. VS: view-selective cross-attention. Inv: Layout-Guided Structure Inversion. (Right) Full scene generation (Stage 3) at scene/asset level. All CD values multiplied by 10^3. Best and second-best highlighted.

	Stage 2: Coarse Structure					Stage 3: Full Scene Generation							
										PSNR↑		LPIPS↓	
Configuration	IoU↑	F1↑	Prec.↑	Rec.↑	CD↓ ($\times 10^3$)	Level	CD↓ ($\times 10^3$)	F1@.01↑	F1@.02↑	Top -down	Int -erior	Top -down	Int -erior
Trained w/o VS-CrossAttn													
w/o VS	44.36	55.61	55.69	55.76	15.92	-	-	-	-	-	-	-	-
Trained w/ VS-CrossAttn						*Trained w/ AS-CrossAttn*							
w/ VS ($t_0=1.0$)	57.54	69.07	69.19	69.12	7.68	Scene	1.52	35.26	79.89	19.02	11.22	0.228	0.653
						Asset	1.92	53.42	76.28	-	-	-	-
+ Inv ($t_0=0.3$)	58.06	69.66	70.07	69.49	7.41	Scene	1.48	36.63	81.48	19.17	11.22	0.219	0.651
						Asset	1.80	55.42	78.65	-	-	-	-
+ Inv ($t_0=0.5$)	58.48	69.97	70.33	69.82	7.66	Scene	0.79	36.01	81.69	19.16	11.22	0.224	0.654
						Asset	1.02	53.77	77.59	-	-	-	-
+ Inv ($t_0=0.7$)	57.09	68.78	68.97	68.79	7.76	Scene	0.63	36.69	82.10	19.17	11.22	0.220	0.650
						Asset	1.72	56.71	79.52	-	-	-	-

6.4 Ablation Study

View-Selective Cross-Attention. To evaluate the effectiveness of view-selective (VS) cross-attention, we train a model with standard cross-attention (w/o VS) alongside our default model (w/ VS). Table 1 (left) compares the two models on Stage 2. Without VS, all voxels attend uniformly to the full cubemap, yielding IoU of 44.36. VS cross-attention raises IoU to 57.54 and halves CD, confirming the importance of spatially grounded conditioning (Fig. 10(a-2 vs. a-3)).

Layout-Guided Structure Inversion. Using the w/ VS model, we vary t_0 at inference (Table 1, left). $t_0 = 0.5$ achieves the best IoU, and all inversion

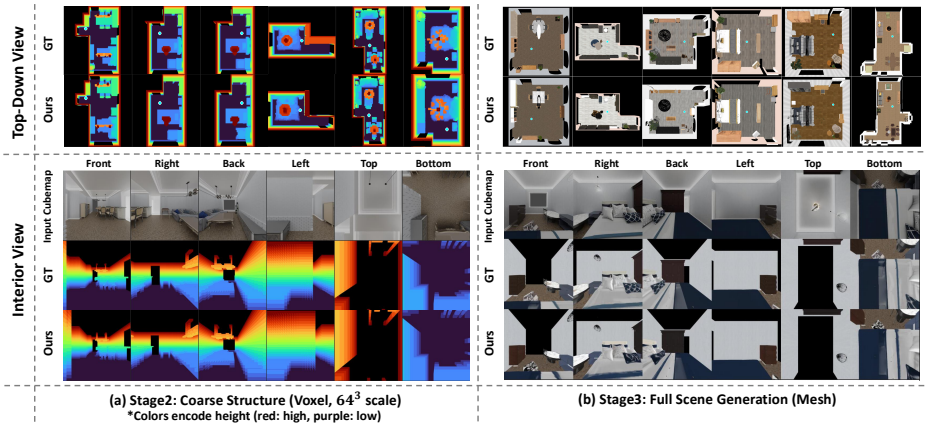

Fig. 9: Rendered evaluation views on ERP-FRONT. (a) Coarse structure (Stage 2) and (b) full scene generation (Stage 3), rendered from a top-down view and six interior views from the camera center along cubemap directions for quantitative evaluation. Cyan dots indicate the calibrated camera center.

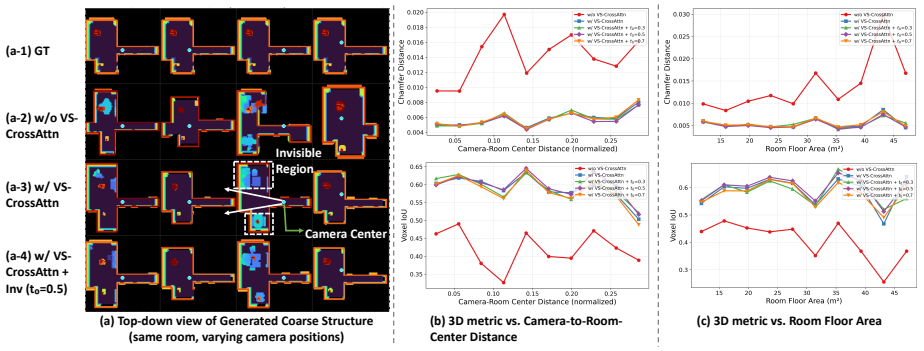

Fig. 10: Ablation on Stage 2 across camera positions. (a) Generated coarse structure from different camera positions in the same room. (b, c) Quantitative metrics across multiple rooms: w/o VS degrades sharply with camera-to-room-center distance (b) and room floor area (c), while VS with Inversion remains robust.

settings yield consistently strong results with marginal differences, confirming the robustness of Layout-Guided Structure Inversion.

Camera Position Robustness. Fig. 10(a) shows coarse structures from different camera positions in the same room. The w/o VS model predicts inconsistent and incorrect layouts across positions (a-2), while the w/ VS model consistently recovers accurate room layouts (a-3,4). Notably, when the camera position introduces occluded regions, the model often generates assets in invisible areas. Fig. 10(b) measures Voxel IoU and CD as a function of camera-to-room-center distance. The w/ VS model maintains performance, while the w/o VS model degrades rapidly. Fig. 10(c) shows that the w/ VS model maintains robust performance across room sizes, while the w/o VS model degrades rapidly.

7 Conclusion

We presented InSpace, a framework for generating complete 3D indoor scenes from a single ERP image. Leveraging full 360° coverage, InSpace introduces view-selective and asset-selective cross-attention for spatially grounded generation, with Layout-Guided Structure Inversion to incorporate geometric priors at inference. Trained on ERP-FRONT, a paired ERP Image-to-3D dataset based on 3D-FRONT, experiments show strong performance across 3D and 2D metrics.

Acknowledgements

This work was supported by the Korea Planning & Evaluation Institute of Industrial Technology(KEIT) and the Ministry of Trade, Industry & Resources (MOTIR) of the Republic of Korea (RS-2024-00417108), and supported by the Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2021-II211381, Development of Causal AI through Video Understanding and Reinforcement Learning, and Its Applications to Real Environments).

References

1. Ai, H., Cao, Z., Wang, L.: A survey of representation learning, optimization strategies, and applications for omnidirectional vision: H. ai et al. International Journal of Computer Vision **133**(8), 4973–5012 (2025)
2. Ai, H., Cao, Z., Zhu, J., Bai, H., Chen, Y., Wang, L.: Deep learning for omnidirectional vision: A survey and new perspectives. arXiv preprint arXiv:2205.10468 (2022)
3. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor fusion IV: control paradigms and data structures. vol. 1611, pp. 586–606. Spie (1992)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
5. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. In: 2017 International Conference on 3D Vision (3DV). pp. 667–676. IEEE Computer Society (2017)
6. Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., et al.: Sam 3d: 3dfy anything in images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7220–7232 (2026)
7. Chen, Y., Li, Z., Wang, Y., Zhang, H., Li, Q., Zhang, C., Lin, G.: Ultra3d: Efficient and high-fidelity 3d generation with part attention. arXiv preprint arXiv:2507.17745 (2025)
8. Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. In: ICLR 2023 (Eleventh International Conference on Learning Representations) (2023)
9. Denninger, M., Sundermeyer, M., Winkelbauer, D., Zidan, Y., Olefir, D., Elbadrawy, M., Lodhi, A., Katam, H.: Blenderproc. arXiv preprint arXiv:1911.01911 (2019)
10. Ding, L., Dong, S., Li, Y., Gao, C., Chen, X., Han, R., Kuang, Y., Zhang, H., Huang, B., Huang, Z., Wang, Z., Xu, D., Xue, T.: Fullpart: Generating each 3d part at full resolution (2025), `https://arxiv.org/abs/2510.26140`
11. Dong, W., Yang, B., Yang, Z., Li, Y., Hu, T., Bao, H., Ma, Y., Cui, Z.: Hiscene: creating hierarchical 3d scenes with isometric view generation. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 9783–9792 (2025)
12. Dong, Y., Fang, C., Bo, L., Dong, Z., Tan, P.: Panocontext-former: Panoramic total scene understanding with a transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28087–28097 (2024)
13. Fu, H., Cai, B., Gao, L., Zhang, L.X., Wang, J., Li, C., Zeng, Q., Sun, C., Jia, R., Zhao, B., et al.: 3d-front: 3d furnished rooms with layouts and semantics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10933–10942 (2021)
14. Gao, D., Rozenberszki, D., Leutenegger, S., Dai, A.: Diffcad: Weakly-supervised probabilistic cad model retrieval and alignment from an rgb image. ACM Transactions on Graphics (TOG) **43**(4), 1–15 (2024)
15. Gu, Z., Cui, Y., Li, Z., Wei, F., Ge, Y., Gu, J., Liu, M.Y., Davis, A., Ding, Y.: Artiscene: Language-driven artistic 3d scene generation through image intermediary. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2891–2901 (2025)

16. Gümeli, C., Dai, A., Nießner, M.: Roca: Robust cad model retrieval and alignment from a single image. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4022–4031 (2022)
17. He, X., Zou, Z.X., Chen, C.H., Guo, Y.C., Liang, D., Yuan, C., Ouyang, W., Cao, Y.P., Li, Y.: Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14822–14833 (2025)
18. 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: International Conference on Learning Representations. vol. 2024, pp. 50678–50702 (2024)
19. Huang, Z., Guo, Y.C., An, X., Yang, Y., Li, Y., Zou, Z.X., Liang, D., Liu, X., Cao, Y.P., Sheng, L.: Midi: Multi-instance diffusion for single image to 3d scene generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23646–23657 (2025)
20. Hunyuan3D, T., Yang, S., Yang, M., Feng, Y., Huang, X., Zhang, S., He, Z., Luo, D., Liu, H., Zhao, Y., et al.: Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. arXiv preprint arXiv:2506.15442 (2025)
21. Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., Carreira, J.: Perceiver: General perception with iterative attention. In: International conference on machine learning. pp. 4651–4664. PMLR (2021)
22. Jia, T., Yan, D., Hao, D., Li, Y., Zhang, K., He, X., Li, L., Wang, Y., Chen, J., Jiang, L., Yin, Q., Quan, L., Chen, Y.C., Yuan, L.: Ultrashape 1.0: High-fidelity 3d shape generation via scalable geometric refinement (2025), https://arxiv.org/abs/2512.21185
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
24. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)
25. Koo, G., Yoon, S., Hong, J.W., Yoo, C.D.: Flexiedit: Frequency-aware latent refinement for enhanced non-rigid editing. In: European Conference on Computer Vision. pp. 363–379. Springer (2024)
26. Koo, G., Yoon, S., Lee, Y., Hong, J.W., Yoo, C.D.: Flowdrag: 3d-aware drag-based image editing with mesh-guided deformation vector flow fields. In: International Conference on Machine Learning. pp. 31467–31485. PMLR (2025)
27. Kuo, W., Angelova, A., Lin, T.Y., Dai, A.: Mask2cad: 3d shape prediction by learning to segment and retrieve. In: European Conference on Computer Vision. pp. 260–277. Springer (2020)
28. Kuo, W., Angelova, A., Lin, T.Y., Dai, A.: Patch2cad: Patchwise embedding learning for in-the-wild shape retrieval from a single image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12589–12599 (2021)
29. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., et al.: Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. arXiv preprint arXiv:2506.15742 (2025)
30. Lai, Z., Zhao, Y., Zhao, Z., Liu, H., Lin, Q., Huang, J., Guo, C., Yue, X.: Lattice: Democratize high-fidelity 3d generation at scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19982–19992 (2026)

31. Lai, Z., Zhao, Y., Zhao, Z., Liu, H., Wang, F., Shi, H., Yang, X., Lin, Q., Huang, J., Liu, Y., et al.: Unleashing vecset diffusion model for fast shape generation. arXiv preprint arXiv:2503.16302 (2025)
32. Li, D.Y., Zhao, W., Chen, Y., Hu, W., Guo, M.H., Zhang, F.L., Shan, Y., Hu, S.M.: Pixal3d: Pixel-aligned 3d generation from images. arXiv preprint arXiv:2605.10922 (2026)
33. Li, H., Zheng, W., He, J., Liu, Y., Lin, X., Yang, X., Chen, Y.C., Guo, C.: Da^2: Depth anything in any direction (2025), https://arxiv.org/abs/2509.26618
34. Li, W., Liu, J., Chen, R., Liang, Y., Chen, X., Tan, P., Long, X.: Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. arXiv preprint arXiv:2405.14979 (2024)
35. Li, W., Zhang, X., Sun, Z., Qi, D., Li, H., Cheng, W., Cai, W., Wu, S., Liu, J., Wang, Z., et al.: Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. arXiv preprint arXiv:2505.07747 (2025)
36. Li, Y., Zou, Z.X., Liu, Z., Wang, D., Liang, Y., Yu, Z., Liu, X., Guo, Y.C., Liang, D., Ouyang, W., et al.: Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
37. Li, Z., Wang, Y., Zheng, H., Luo, Y., Wen, B.: Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. Advances in Neural Information Processing Systems **38**, 118582–118600 (2026)
38. Lin, Y., Lin, C., Pan, P., Yan, H., Feng, Y., Mu, Y., Fragkiadaki, K.: Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers (2025), https://arxiv.org/abs/2506.05573
39. Ling, L., Lin, C.H., Lin, T.Y., Ding, Y., Zeng, Y., Sheng, Y., Ge, Y., Liu, M.Y., Bera, A., Li, Z.: Scenethesis: A language and vision agentic framework for 3d scene generation (2025), https://arxiv.org/abs/2505.02836
40. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
41. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9298–9309 (2023)
42. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: The Eleventh International Conference on Learning Representations (ICLR) (2023)
43. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: European conference on computer vision. pp. 531–548. Springer (2022)
44. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11461–11471 (2022)
45. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021)
46. Meng, Y., Wu, H., Zhang, Y., Xie, W.: Scenegen: Single-image 3d scene generation in one feedforward pass. In: 2026 International Conference on 3D Vision (3DV). pp. 543–553. IEEE (2026)
47. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)

48. Razavi, A., Van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. Advances in neural information processing systems **32** (2019)
49. Ren, X., Huang, J., Zeng, X., Museth, K., Fidler, S., Williams, F.: Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4209–4219 (2024)
50. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
51. Shi, Y., Xue, C., Liew, J.H., Pan, J., Yan, H., Zhang, W., Tan, V.Y., Bai, S.: Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8839–8849 (2024)
52. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. arXiv preprint arXiv:2508.10104 (2025)
53. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
54. Wu, S., Lin, Y., Zhang, F., Zeng, Y., Yang, Y., Qian, J., Zhu, S., Cao, X., Torr, P., Yao, Y., et al.: Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. Advances in Neural Information Processing Systems **38**, 170778–170804 (2026)
55. Xiang, J., Chen, X., Xu, S., Wang, R., Lv, Z., Deng, Y., Zhu, H., Dong, Y., Zhao, H., Yuan, N.J., et al.: Native and compact structured latents for 3d generation. arXiv preprint arXiv:2512.14692 (2025)
56. 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: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21469–21480 (2025)
57. Xu, J., Cheng, W., Gao, Y., Wang, X., Gao, S., Shan, Y.: Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. arXiv preprint arXiv:2404.07191 (2024)
58. Yang, Y., Jia, B., Zhang, S., Huang, S.: Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. Advances in neural information processing systems **38**, 140319–140351 (2026)
59. Yang, Y., Zhou, Y., Guo, Y.C., Zou, Z.X., Huang, Y., Liu, Y.T., Xu, H., Liang, D., Cao, Y.P., Liu, X.: Omnipart: Part-aware 3d generation with semantic decoupling and structural cohesion. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–12 (2025)
60. Yao, K., Zhang, L., Yan, X., Zeng, Y., Zhang, Q., Xu, L., Yang, W., Gu, J., Yu, J.: Cast: Component-aligned 3d scene reconstruction from an rgb image. ACM Transactions on Graphics (TOG) **44**(4), 1–19 (2025)
61. Ye, C., Wu, Y., Lu, Z., Chang, J., Guo, X., Zhou, J., Zhao, H., Han, X.: Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 25050–25061 (2025)
62. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11784–11793 (2021)

63. Yoon, S., Koo, G., Hong, J.W., Yoo, C.D.: Dni: Dilutional noise initialization for diffusion video editing. In: European Conference on Computer Vision. pp. 180–195. Springer (2024)
64. Zhang, B., Tang, J., Niessner, M., Wonka, P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. ACM Transactions On Graphics (TOG) **42**(4), 1–16 (2023)
65. 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 Transactions on Graphics (TOG) **43**(4), 1–20 (2024)
66. 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)
67. Zhao, Z., Liu, W., Chen, X., Zeng, X., Wang, R., Cheng, P., Fu, B., Chen, T., Yu, G., Gao, S.: Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. Advances in neural information processing systems **36**, 73969–73982 (2023)
68. Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., Zhou, Z.: Structured3d: A large photo-realistic dataset for structured 3d modeling. In: European Conference on Computer Vision. pp. 519–535. Springer (2020)