SceneGen: Single-Image 3D Scene Generation in One Feedforward Pass

Yanxu Meng*, Haoning Wu*, Ya Zhang, Weidi Xie
School of Artificial Intelligence, Shanghai Jiao Tong University

Figure 1. **Overview.** Our proposed **SceneGen** framework takes a single scene image and its corresponding object masks as inputs, and efficiently generates multiple 3D assets with coherent geometry, texture, and spatial arrangement in a single feedforward pass.

Abstract

3D content generation has recently attracted significant research interest, driven by its critical applications in VR/AR and embodied AI. In this work, we tackle the challenging task of ***synthesizing multiple 3D assets within a single scene image****. Concretely, our contributions are fourfold: (i) we present* ***SceneGen****, a novel framework that takes a scene image and corresponding object masks as input, simultaneously producing multiple 3D assets with geometry and texture. Notably, SceneGen operates with no need for extra optimization or asset retrieval; (ii) we introduce a novel* ***feature aggregation*** *module that integrates local and global scene information from visual and geometric encoders within the* ***feature extraction*** *module. Coupled with a* ***position head****, this enables the generation of 3D assets and their relative spatial positions in a single feedforward pass; (iii) we demonstrate SceneGen's direct extensibility to multi-image input scenarios. Despite being trained solely on single-image inputs, our architecture yields improved generation performance when multiple images are provided; and (iv) extensive quantitative and qualitative evaluations confirm the efficiency and robustness of our approach. We believe this paradigm offers a novel solution for high-quality 3D content generation, potentially advancing its practical applications in downstream tasks. The code and model will be publicly available at:* https://mengmouxu.github.io/SceneGen.

"Everything you can imagine is real."

—— *Pablo Picasso*

*: These authors contribute equally to this work.

1. Introduction

The growing demand for immersive digital environments in applications such as virtual/augmented reality (VR/AR) and embodied AI has spurred significant advancements in 3D content generation [6, 8, 9, 13, 14, 28]. While early efforts primarily focus on synthesizing individual 3D assets [33, 58, 68], recent research attention has shifted to the more challenging task of 3D scene generation. Generating realistic 3D scenes [5, 11, 15, 16, 19, 63, 64], whether conditioned on input text or images, requires synthesizing multiple assets with accurate geometry, texture, and spatial relationships. This challenge fundamentally hinges on two key capabilities: (i) **3D asset generation** for creating plausible asset geometric topologies from limited textual or visual input; and (ii) **spatial arrangement** for managing inter-object spatial relationships to ensure physical plausibility, such as support, occlusion, and other interactions among assets.

In general, existing works fall into two paradigms: (i) retrieval-based methods [12, 38, 48, 60] typically employ LLMs for layout planning and retrieve matching 3D assets from existing libraries to assemble scenes. Though straightforward, their flexibility is constrained by the coverage of available asset libraries; (ii) two-stage approaches [18, 35, 61] first synthesize individual 3D assets and then employ vision-language models (VLMs) or optimization techniques to refine scene structure and spatial arrangement. While more flexible, their reliance on iterative optimization inevitably leads to inefficiency and error accumulation. The most relevant works to ours are PartCrafter [34] and MIDI [23], which generate parts or multiple assets from a single image. However, they still suffer from limited synthesis fidelity and inaccurate spatial relations among assets.

To tackle the aforementioned challenges, we propose **SceneGen**, a novel 3D scene generation model designed to simultaneously generate multiple assets, including their geometry, texture, and spatial positions, from a single scene image in a single feedforward pass (Figure 1). Concretely, our framework builds upon an existing single-asset generation model [58] and incorporates three key modules: **feature extraction**, **feature aggregation**, and **output**.

Specifically, the feature extraction module first strategically leverages off-the-shelf visual [42] and geometric [52] encoders to extract both asset-level and scene-level representations. Subsequently, our proposed feature aggregation module, composed of local and global attention blocks, effectively integrates these visual and geometric features while facilitating inter-asset interactions during generation to ensure geometrically plausible topologies. Finally, leveraging this comprehensive scene context, the output module can directly decode the generated latent features into the assets' relative position, geometry, and texture via a position head and a pre-trained structure decoder.

Moreover, despite being trained exclusively on single-image samples, SceneGen exhibits remarkable generalization to multi-image input scenarios, yielding even better generation quality, which primarily stems from our dedicated architectural design. To ensure a comprehensive and reliable evaluation of SceneGen, we systematically adopt multiple metrics focusing on both geometric and visual quality. Both quantitative and qualitative results demonstrate that our proposed SceneGen significantly outperforms previous methods in terms of generation quality and efficiency, which can generate a textured scene with four assets in approximately 2 minutes on a single A100 GPU.

The rest of this paper is organized as follows: Sec. 2 provides a comprehensive review and discussion of related literature. Sec. 3 elaborates on our proposed SceneGen framework. Sec. 4 presents extensive quantitative and qualitative evaluations. Finally, Sec. 5 concludes with key insights and contributions. To our knowledge, SceneGen is the first 3D scene generation model capable of **simultaneously synthesizing geometry, texture, and relative positions of multiple 3D assets in a single feedforward pass, without requiring per-scene optimization**. We believe this work will inspire future advances in high-quality, efficient 3D content generation and facilitate diverse downstream applications.

2. Related Work

3D visual perception. Extensive research has advanced 3D visual perception, where traditional methods like SfM [47, 51] rely on computationally intensive optimization for 3D reconstruction. Notably, emerging feedforward methods [3, 27, 30, 50, 52–54, 67, 70] have demonstrated efficient 3D perception, with DUSt3R [54] pioneering this trend and VGGT [52] establishing a minimalist yet powerful paradigm that distills geometric priors from large-scale data without explicit 3D inductive biases or optimizations.

3D asset synthesis. Typically, 3D asset synthesis aims to generate object-centric geometry and texture from text or image inputs. The recent success of diffusion models [22] in 2D generation [37, 43, 46, 55, 56] has inspired the development of learning-based, scalable 3D content [6, 8, 9, 13, 14, 28] generation, which produce 3D asset in various representations, including explicit forms such as point clouds [40], voxels [25, 41], and SDFs [4, 32], as well as implicit ones like 3D Gaussians [20, 66] and NeRFs [1, 31, 59]. Subsequent advances leverage VAEs [29] for compressing 3D geometry or textures [33, 58, 68] and adopt hybrid mesh-texture pipelines [24, 26, 57, 65], with TRELLIS [58] demonstrating scalable, high-fidelity generation via structured latents. Nevertheless, these methods remain restricted to single-asset synthesis and fundamentally lack the capability to model complex multi-asset scenes.

3D scene generation. Beyond single-asset synthesis, 3D scene generation is more challenging yet valuable, aiming

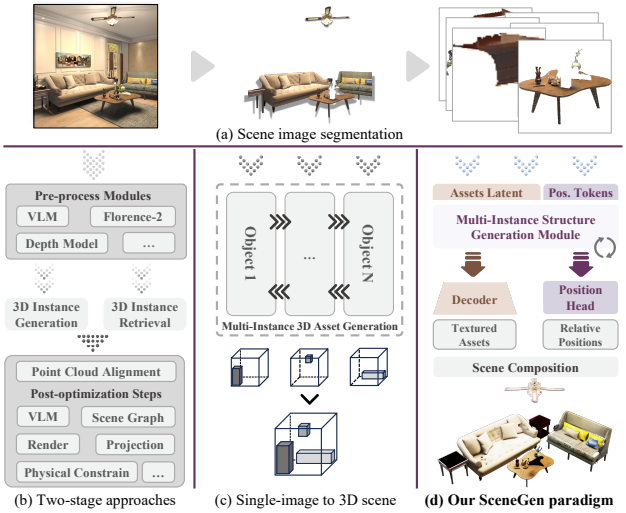

Figure 2. **3D Scene Generation.** (a) Existing methods typically require segmenting target objects from the scene image; (b) Two-stage methods like CAST [61] sequentially retrieve or generate individual assets, then assemble them via post-processing; (c) Methods such as MIDI [23] directly generate multiple assets from a single image, but suffer from blurry details and unreasonable spatial layouts; (d) In contrast, our SceneGen jointly synthesizes the geometry, texture, and spatial positions of multiple assets in a single feedforward pass, producing plausible 3D scenes.

to produce multiple coordinated, physically plausible assets within a scene. Prior text-based approaches primarily leverage LLMs for layout planning [12, 38, 48, 60] and retrieve suitable assets from existing libraries. Subsequent image-based methods employ segmentation [5, 10, 15, 19, 63], scene graphs [11, 16, 64] and depth/point cloud alignment [10, 49, 61] to assist in multi-asset generation and arrangement. As depicted in Figure 2 (b), recent optimization-based methods [18, 35, 61] adopt VLMs for post-processing, refining scene structures via image- or text-guided adjustments, but inevitably suffer from inefficiency. Other works (Figure 2 (c)), such as MIDI [23] and PartCrafter [34], explore scene generation conditioned on a single image, but inherently sacrifice reconstruction fidelity due to their reliance on canonical-space representations. To overcome these limitations, our proposed **SceneGen** uniquely integrates asset-level and scene-level features, enabling robust and efficient 3D scene generation.

3. Method

In this work, we present **SceneGen**, designed to jointly perform 3D asset generation within scenes and predict relative spatial positions among assets. Here, we first formally describe our problem formulation in Sec. 3.1; followed by elaboration on our model architecture and training method in Sec. 3.2 and Sec. 3.3, respectively; finally, we extend SceneGen to multi-view input scenarios in Sec. 3.4.

3.1. Problem Formulation

Our proposed **SceneGen** is a single-stage feedforward 3D scene generation model ($\boldsymbol{\mathcal{G}}_{\text{Scene}}$), which takes a scene image ($\boldsymbol{I}_{\text{Scene}}$) containing N objects and corresponding masks ($\{\boldsymbol{m}_i\}_{i=1}^N$) as input, simultaneously generating 3D asset structure and texture representations ($\{\boldsymbol{S}_i\}_{i=1}^N$), and their relative positions ($\{\boldsymbol{P}_i\}_{i=1}^N$), formulated as:

$$\{(\boldsymbol{S}_i, \boldsymbol{P}_i)\}_{i=1}^N = \boldsymbol{\mathcal{G}}_{\text{Scene}}(\boldsymbol{I}_{\text{Scene}}, \{\boldsymbol{m}_i\}_{i=1}^N)$$

Here, the position of each asset relative to a pre-selected query asset, is denoted as $\boldsymbol{P}_i = [\boldsymbol{t}_i, \boldsymbol{q}_i, \boldsymbol{s}_i] \in \mathbb{R}^8$, comprising $\boldsymbol{t}_i \in \mathbb{R}^3$ (translation), $\boldsymbol{q}_i \in \mathbb{R}^4$ (rotation quaternion), and $\boldsymbol{s}_i \in \mathbb{R}^1$ (scale factor). By default, we select the asset with $i = 1$ as the query asset, with its parameters fixed as: $\boldsymbol{t}_{\text{query}} = [0, 0, 0]$, $\boldsymbol{q}_{\text{query}} = [1, 0, 0, 0]$, $\boldsymbol{s}_{\text{query}} = 1$.

3.2. SceneGen

SceneGen framework ($\boldsymbol{\mathcal{G}}_{\text{Scene}}$) comprises three key stages: (i) **feature extraction**, employing a scene visual encoder (Φ_V) and a scene geometric encoder (Φ_G) to extract visual and structural features within the scene, implemented using pre-trained DINOv2 [42] and VGGT [52], respectively; (ii) **feature aggregation**, comprising M DiT [43] blocks, each integrating a *local attention* block, a *global attention* block, and a feedforward network; and (iii) **output** module, which introduces a position head (Ψ_{pos}) for predicting the spatial locations of assets and adopts off-the-shelf sparse structure (SS) and structured latents (SLAT) decoders [58] for decoding scene geometry structures. By integrating these complementary modules, our SceneGen effectively captures both local asset-level and global scene-level features, enabling it to simultaneously generate multiple 3D assets and predict their relative positions.

Feature extraction. SceneGen starts with extracting both local and global features from a given scene image ($\boldsymbol{I}_{\text{scene}}$) with the visual encoder (Φ_V) and geometric encoder (Φ_G). Specifically, for each object with its corresponding segmentation mask ($\boldsymbol{m}_i$), we obtain four complementary feature representations: (i) the object's individual visual features ($\mathcal{F}_i^V$); (ii) the visual features of its mask ($\mathcal{F}_i^{\text{mask}}$); (iii) scene global visual features ($\mathcal{F}_{\text{global}}^V$); and (iv) the global geometric features ($\mathcal{F}_{\text{global}}^{\text{geo}}$), formulated as:

$$\mathcal{F}_i^V = \Phi_V(\boldsymbol{I}_{\text{scene}} \otimes \boldsymbol{m}_i), \quad \mathcal{F}_i^{\text{mask}} = \Phi_V(\boldsymbol{m}_i),$$
$$\mathcal{F}_{\text{global}}^V = \Phi_V(\boldsymbol{I}_{\text{scene}}), \quad \mathcal{F}_{\text{global}}^{\text{geo}} = \Phi_G(\boldsymbol{I}_{\text{scene}})$$

Here, $\otimes$ denotes pixel-wise multiplication. These features are then concatenated along the sequence length dimension into a unified scene context ($\mathcal{F}_i^{\text{scene}}$), formulated as:

$$\mathcal{F}_i^{\text{scene}} = [\mathcal{F}_i^V; \mathcal{F}_i^{\text{mask}}; \mathcal{F}_{\text{global}}^V; \mathcal{F}_{\text{global}}^{\text{geo}}]$$

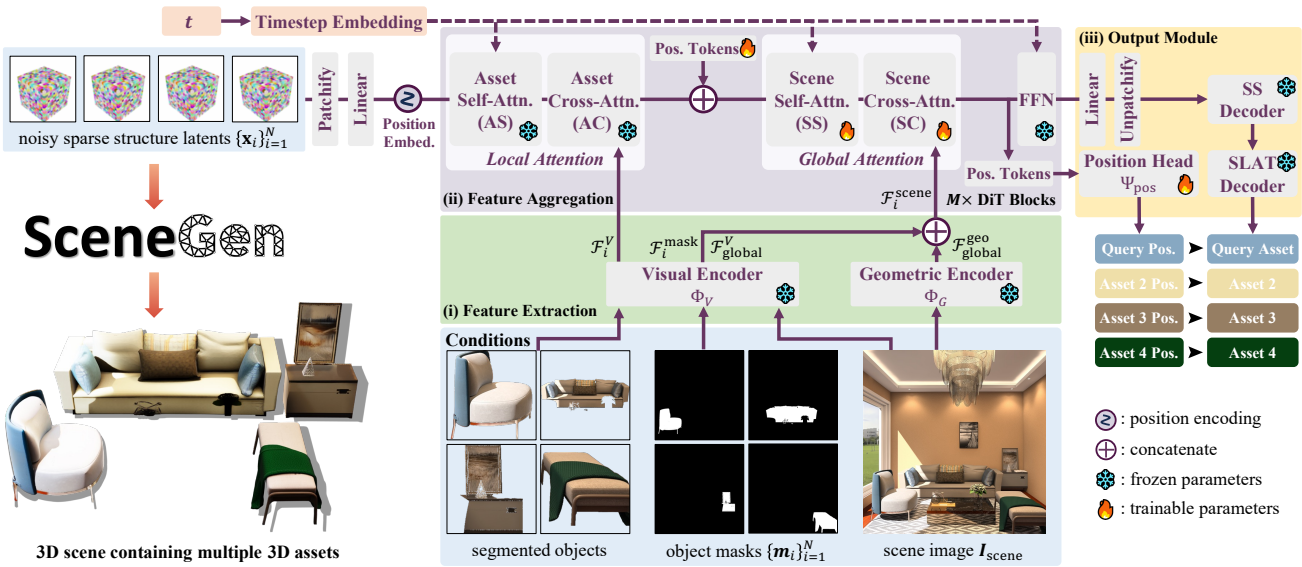

Figure 3. **Architecture Overview. SceneGen** takes a single scene image with multiple objects and corresponding segmentation masks as input. A pre-trained *local attention* block first refines the texture of each asset. Then, our introduced *global attention* block integrates asset-level and scene-level features extracted by dedicated visual and geometric encoders. Finally, two off-the-shelf structure decoders and our *position head* decode these latent features into multiple 3D assets with geometry, texture, and relative spatial positions.

Feature aggregation. To integrate the extracted scene context features ($\mathcal{F}_i^{\text{scene}}$), SceneGen employs a feature aggregation module, that enables the simultaneous generation of multiple 3D assets. This module comprises a *local attention* block that refines details of individual assets, a *global attention* block that incorporates scene context to facilitate inter-asset interactions, and a feedforward network. Concretely, the local attention blocks and feedforward networks are initialised from pre-trained weights from TRELLIS [58], a flow-matching [36] model designed to synthesize 3D content from noisy sparse structure latents. For clarity and conciseness, given the sparse structure latents ($\{\mathbf{x}_i\}_{i=1}^N$, where each $\mathbf{x}_i \in \mathbb{R}^{T\times C}$) of N objects in a scene, we denote the standard attention mechanism as $\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, and elaborate on a single DiT block as follows.

The *local attention* block aims to enhance details of individual objects through asset-level self-attention (AS) and cross-attention (AC). To be specific, it focuses on fusing the latent features of each object ($\mathbf{x}_i$) with their corresponding visual features ($\mathcal{F}_i^V$) to yield refined representations of each object ($\mathbf{x}_i^{\text{AC}}$), which can be formulated as:

$$\mathbf{x}_i^{\text{AS}} = \text{Attention}(\mathbf{x}_i, \mathbf{x}_i, \mathbf{x}_i)$$
$$\mathbf{x}_i^{\text{AC}} = \text{Attention}(\mathbf{x}_i^{\text{AS}}, \mathcal{F}_i^V, \mathcal{F}_i^V)$$

To establish inter-dependencies among 3D assets, we propose a *global attention* block, comprising scene-level self-attention (SS) and cross-attention (SC), which capture inter-object relationships and integrate scene geometry, respectively. Consequently, this design ensures physically plausible spatial arrangements of generated assets.

Similar to [52], we initialize one learnable position token ($\boldsymbol{p}_i$) and four register tokens ($\boldsymbol{r}_i$) [7] for refined features of each object ($\mathbf{x}_i^{\text{AC}}$), denoted as: $\hat{\mathbf{x}}_i = [\boldsymbol{p}_i; \boldsymbol{r}_i; \mathbf{x}_i^{\text{AC}}]$, where $[\cdot;\cdot]$ refers to concatenation along the token length dimension. Notably, we assign a unique position token ($\boldsymbol{p}_{\text{query}}$) and register tokens ($\boldsymbol{r}_{\text{query}}$) to the query asset, while adopting shared position token ($\boldsymbol{p}_i$) and register tokens ($\boldsymbol{r}_i$) for other assets. For each asset feature ($\hat{\mathbf{x}}_i \in \mathbb{R}^{T\times C}$), we concatenate them along the token sequence dimension to form a unified scene representation ($\mathbf{X} \in \mathbb{R}^{(N\cdot T)\times C}$), which is processed by our scene-level self-attention layer, resulting in updated tokens of each asset ($\{\mathbf{x}_i^{\text{SS}}\}_{i=1}^N$), formulated as:

$$\{\mathbf{x}_i^{\text{SS}}\}_{i=1}^N = \text{Attention}(\mathbf{X}, \mathbf{X}, \mathbf{X})$$

Through this process, intra-asset and inter-asset information aggregation establishes essential shape and position awareness for coherent multi-asset generation. We then employ scene-level cross-attention to integrate multiple pre-extracted scene-aware features, thus incorporating 3D geometric context. The features of each asset are updated into geometry-aware representations ($\{\mathbf{x}_i^{\text{SC}}\}_{i=1}^N$), denoted as:

$$\mathbf{x}_i^{\text{SC}} = \text{Attention}(\mathbf{x}_i^{\text{SS}}, \mathcal{F}_i^{\text{scene}}, \mathcal{F}_i^{\text{scene}})$$

This preserves object-specific details while integrating global geometric constraints, which effectively addresses occlusion challenges and enables geometric refinement.

Output module. After passing through M DiT blocks, we obtain the updated position tokens ($\{\hat{\boldsymbol{p}}\}_{i=1}^N$) and latent features ($\{\tilde{\mathbf{x}}\}_{i=1}^N$) of each generated asset, which are subsequently decoded into their relative spatial positions and

detailed 3D representations (structure and texture), respectively. For relative positions, we concatenate the position tokens of all non-query assets, which are then decoded into corresponding 8D position vectors ($\{\hat{\boldsymbol{P}}_i\}_{i=2}^N$) by our proposed position head (Ψ_{pos}), comprising four self-attention layers and a linear layer, denoted as:

$$\{\hat{\boldsymbol{P}}_i\}_{i=2}^N = \{[\hat{\boldsymbol{t}}_i, \hat{\boldsymbol{q}}_i, \hat{\boldsymbol{s}}_i]\}_{i=2}^N = \Psi_{\text{pos}}(\{\hat{\boldsymbol{p}}_i\}_{i=2}^N)$$

Here, each vector ($\hat{\boldsymbol{P}}_i$) represents an asset's spatial position (translation, rotation, and scale) relative to the preselected query asset ($i = 1$). Furthermore, the latent features can be directly decoded into the geometry and texture of each asset ($\{\hat{\boldsymbol{S}}\}_{i=1}^N$) using off-the-shelf sparse structure generator ($\boldsymbol{\mathcal{G}}_{\text{S}}$) and structured latents generator ($\boldsymbol{\mathcal{G}}_{\text{L}}$) from TRELLIS [58], formulated as:

$$\{\hat{\boldsymbol{S}}\}_{i=1}^N = \boldsymbol{\mathcal{G}}_{\text{L}}(\boldsymbol{\mathcal{G}}_{\text{S}}(\{\tilde{\mathbf{x}}\}_{i=1}^N))$$

3.3. Training

During training, only the global attention blocks, learnable position tokens, and position head are optimized, with all other parameters frozen to facilitate efficient training, as depicted in Figure 3. The technical details regarding training data and loss function designs are presented below.

Training data. Our SceneGen model is trained on the 3D-FUTURE [14] dataset, containing photorealistic scene renderings with instance masks and asset annotations. This dataset comprises 12K training scenes and 4.8K test scenes, each featuring a scene image with one or multiple objects. To better capture inter-object spatial relationships, we augment the training set by iteratively designating each asset as the query asset while randomly permuting the remaining assets, which expands the effective training samples to 30K.

Training objectives. Our SceneGen model is trained end-to-end using a composite loss function ($\mathcal{L}$) comprising three key components: (i) the average conditional flow matching [36] loss ($\mathcal{L}_{\text{cfm}}$), applied to each generated asset for supervising asset generation; (ii) the position loss ($\mathcal{L}_{\text{pos}}$) for maintaining accurate relative spatial arrangements among assets; and (iii) the voxel-space collision loss ($\mathcal{L}_{\text{coll}}$) for enforcing physically plausible object placement. The overall objective function ($\mathcal{L}$) combines these components with a weighting factor (λ), which can be formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{cfm}} + \lambda(\mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{coll}})$$

Concretely, the flow matching loss establishes straight probability paths between distributions via linear interpolation: $\mathbf{x}_i(t) = (1-t)\mathbf{x}_i^0 + t\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, $t \in [0, 1]$, and $\mathbf{x}_i^0$ denotes the noise-free sparse structure latents for each of the N assets. The conditional flow matching objective ($\mathcal{L}_{\text{cfm}}$) learns a parameterized function $\boldsymbol{v}_\theta$ to approximate the velocity field ($\boldsymbol{v}(\mathbf{x}_i(t), t) = \nabla_t \mathbf{x}_i(t)$), represented as:

$$\mathcal{L}_{\text{cfm}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{t,\boldsymbol{\epsilon}} \|\boldsymbol{v}_\theta(\mathbf{x}_i(t), t) - (\boldsymbol{\epsilon} - \mathbf{x}_i^0)\|_2^2$$

The position loss ($\mathcal{L}_{\text{pos}}$) adopts a μ-weighted Huber loss ($\|\cdot\|_{\delta_P}$) between the predicted positions ($\hat{\boldsymbol{P}}_i = [\hat{\boldsymbol{t}}_i, \hat{\boldsymbol{q}}_i, \hat{\boldsymbol{s}}_i]$) for all non-query assets ($i \in [2, \ldots, N]$) and their ground truth ($\boldsymbol{P}_i = [\boldsymbol{t}_i, \boldsymbol{q}_i, \boldsymbol{s}_i]$), denoted as:

$$\begin{aligned}\mathcal{L}_{\text{pos}} = \sum_{i=2}^{N} (&\mu_t \|(\hat{\boldsymbol{t}}_i - \boldsymbol{t}_i)/d_{\text{scene}}\|_{\delta_P} \\ &+ \mu_q \|\hat{\boldsymbol{q}}_i - \boldsymbol{q}_i\|_{\delta_P} + \mu_s \|\hat{\boldsymbol{s}}_i - \boldsymbol{s}_i\|_{\delta_P})\end{aligned}$$

Here, the translation error component is normalized by the scene scale (d_{scene}) of each sample to mitigate numerical instability caused by varying query asset selections. This stabilizes translation loss during training while improving generalization across distinct query asset configurations.

The collision loss ($\mathcal{L}_{\text{coll}}$) quantifies surface collision in a $64 \times 64 \times 64$ voxel grid ($\boldsymbol{V}$). Specifically, the predicted sparse structure latents ($\tilde{\mathbf{x}}_i$) are decoded into point clouds ($\{\boldsymbol{p}_i\}_{i=1}^L$) via a pre-trained sparse structure decoder from TRELLIS [58], then transformed using predicted pose parameters ($\hat{\boldsymbol{P}}_i$) and voxelized into $\boldsymbol{V}$. The collision loss is defined as the ratio of overlapping surface voxels to all surface voxels, using the Huber loss ($\|\cdot\|_{\delta_C}$), denoted as:

$$\mathcal{L}_{\text{coll}} = \|\text{IoU}_{\text{scene}}\|_{\delta_C} = \|\frac{\sum_i \mathbb{I}[\boldsymbol{V}_i > 1]}{\sum_i \mathbb{I}[\boldsymbol{V}_i > 0]}\|_{\delta_C}$$

Ideally, $\text{IoU}_{\text{scene}} = 0$ indicates there are no asset collisions.

3.4. Extension to Multi-view Inputs

Despite being trained exclusively on single-image samples, our model exhibits inherent multi-view compatibility, enabled by its flexible feature extraction and conditioning strategy. Given a scene with K input views ($\{\boldsymbol{I}_{\text{scene}}^k\}_{k=1}^K$), the visual features ($\mathcal{F}_V^k$) for each view are extracted independently via the visual encoder (Φ_V), while the geometric features are derived from a unified scene representations encoded by aggregating information across all views using the geometric encoder (Φ_G), denoted as:

$$\mathcal{F}_{\text{geo}}^k = \Phi_G(\{\boldsymbol{I}_{\text{scene}}^j\}_{j=1}^K)[k]$$

The final asset positions are determined by averaging the predictions across all views. Experimental results (detailed in Sec. 4.3) indicate that this multi-view inference scheme improves generation quality by leveraging better geometric understanding, despite the model having never been explicitly fine-tuned on such multi-view inputs.

Method	Instance Specific	Geometric Metrics					Image Category	Visual Metrics						Inference Time (s)
		CD-S↓	CD-O↓	F-Score-S↑	F-Score-O↑	IoU-B↑		PSNR↑	SSIM↑	LPIPS↓	FID↓	CLIP-S↑	DINO-S↑	
PartCrafter [34]	✗	0.2027	—	40.43	—	—	Scene	—	—	—	—	—	—	**7.2**
							GT-Render	—	—	—	—	—	—	
DepR [71]	✓	0.0518	0.0862	63.02	47.66	0.2989	Scene	—	—	—	—	—	—	11.6
							GT-Render	—	—	—	—	—	—	
Gen3DSR [10]	✓	0.0521	0.0935	61.26	41.26	0.2978	Scene	15.92	0.8885	0.1730	63.95	0.8059	0.4334	179.0
							GT-Render	15.43	0.8899	0.1660	78.26	0.7950	0.4416	
MIDI* [23]	✓	0.0501	0.0602	68.74	61.04	0.2493	Scene	**16.93**	0.8814	0.1778	22.75	0.8711	0.6892	42.5
							GT-Render	15.45	0.8814	0.1711	28.26	0.8706	0.7034	
SceneGen	✓	**0.0118**	**0.0138**	**90.60**	**89.73**	**0.5818**	Scene	16.76	**0.8903**	**0.1417**	**19.59**	**0.9152**	**0.8322**	26.0
							GT-Render	**17.59**	**0.8991**	**0.1234**	**12.34**	**0.9236**	**0.8702**	

Table 1. **Quantitative Comparisons on the 3D-FUTURE Test Set.** We evaluate the geometric structure using scene-level Chamfer Distance (CD-S) and F-Score (F-Score-S), object-level Chamfer Distance (CD-O) and F-Score (F-Score-O), and volumetric IoU of object bounding boxes (IoU-B). For visual quality, CLIP-S and DINO-S represent CLIP and DINOv2 image-to-image similarity, respectively. We report the time cost for generating a single asset on a single A100 GPU, and * indicates adopting MV-Adapter [24] for texture rendering.

4. Experiments

This section starts with the experimental settings in Sec. 4.1, followed by comprehensive quantitative and qualitative evaluations in Sec. 4.2 and Sec. 4.3, respectively. Finally, we conduct ablation studies in Sec. 4.4.

4.1. Experimental Settings

Implementation details. All experiments are conducted on 8× NVIDIA A100 GPUs, where we train SceneGen for 240 epochs using the AdamW [39] optimizer with a learning rate of 5×10^{-5} and a batch size of 8. The weighting factor λ decays dynamically within $[0.2, 1]$ using a decay factor of 0.99, and the thresholds of Huber loss δ_P and δ_C are set to 0.02 and 0.05, respectively. To handle varying numbers of assets across training scenes, each training step dynamically samples scenes containing identical asset counts. During inference, we adopt 25 sampling steps with the classifier-free guidance (CFG) weight set to $w = 5.0$.

Evaluation metrics. We assess the generated 3D scenes from both geometric and visual perspectives. For geometry, we reconstruct point clouds from the synthesized asset surfaces and align them with the ground truth using FilterReg [17] for faster and more accurate registration than traditional Iterative Closest Point (ICP [2]). We then compute commonly used point cloud metrics, Chamfer Distance (CD) and F-Score, at both scene and object levels, as well as the volumetric IoU of asset bounding boxes.

For visual quality, we focus on the scene texture rendering. Specifically, after alignment with the ground truth point cloud, we render the predicted scenes with *Blender* from the original input camera viewpoint. We consider two types of ground truth: (i) instance-masked scene images extracted using corresponding object masks, and (ii) images rendered from ground truth assets at the same viewpoint (excluding ambient lighting). We compare our rendered results with both types of ground truth using PSNR, SSIM, LPIPS [69], FID [21], CLIP [44] similarity, and DINOv2 [42] similarity to assess the texture quality of generated assets. Regarding efficiency, we report the inference time cost for synthesizing a single 3D asset on a single A100 GPU. More details will be included in the **Supplementary Material**.

Baselines. We compare SceneGen with representative 3D scene generation methods, including PartCrafter [34], DepR [71], Gen3DSR [10], and MIDI [23], using their pre-trained models. Specifically, we adopt object masks to specify generation targets for all baselines except for PartCrafter, which does not support mask-based control. Instead, we directly provide PartCrafter with extracted objects and the number of assets as input. Moreover, as PartCrafter and DepR do not offer code for texture rendering, our evaluation of these methods focuses on geometric quality, while visual quality is compared with Gen3DSR and MIDI (relying on MV-Adapter [24] for texture synthesis).

Benchmarks. All evaluations are conducted on the 3D-FUTURE [14] test set, comprising 4.8K scenes. Each scene contains a photorealistic rendered image with one or more objects and corresponding segmentation masks as input.

4.2. Quantitative Results

As presented in Table 1, we draw the following key observations: (i) **geometric quality**: SceneGen consistently outperforms existing methods across all scene-level and asset-level metrics. This stems from its joint integration of local asset features and global scene context during generation. The interactions among multiple assets facilitate the model in producing physically plausible geometric structures, while the position head further improves the structural realism by explicitly predicting spatial arrangements. (ii) **visual quality**: SceneGen can render high-quality textures for generated 3D assets without relying on any external texture generation models. Moreover, whether using masked scene images or ground-truth renderings as references, our

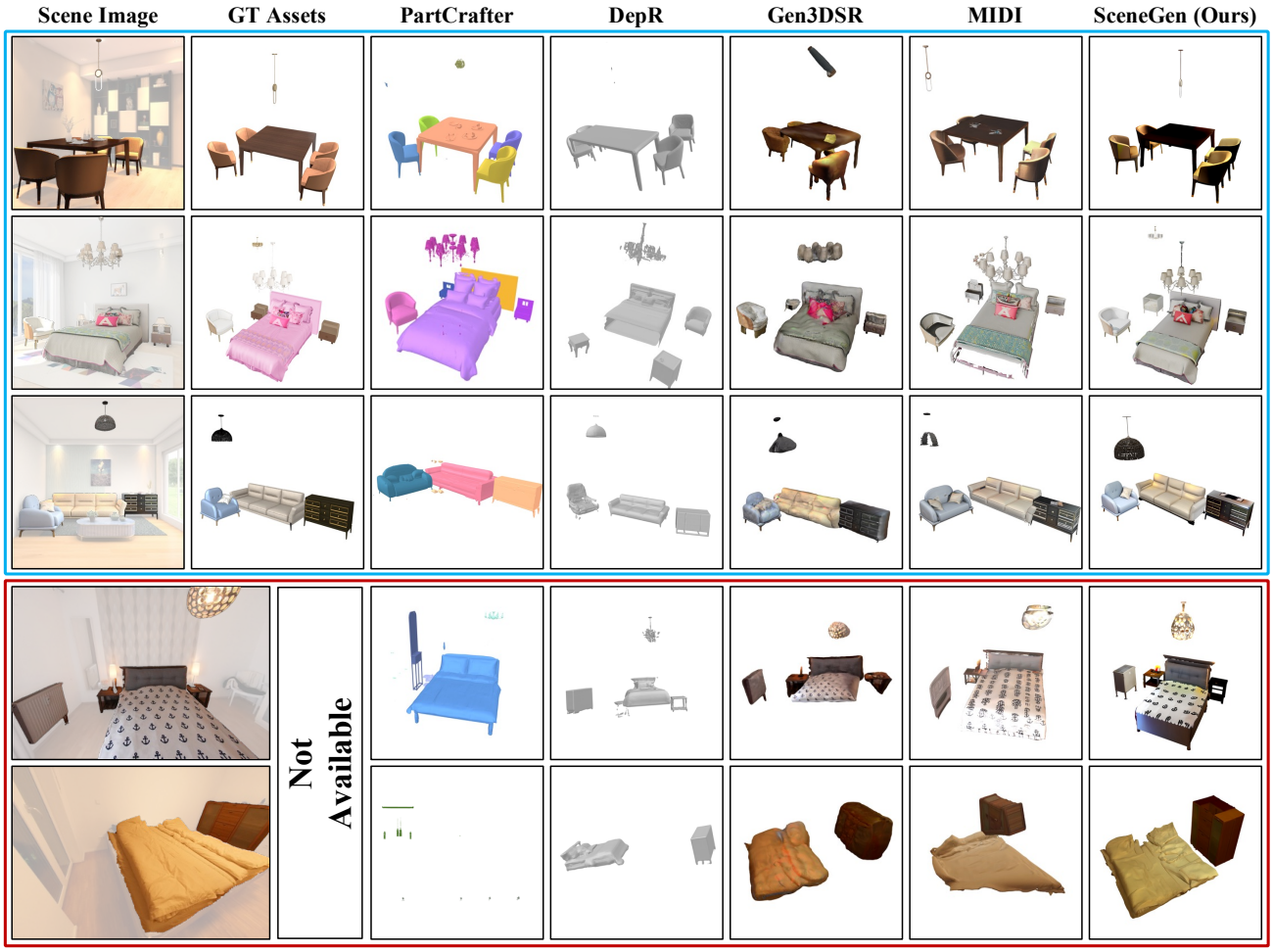

Figure 4. **Qualitative Comparisons on the 3D FUTURE Test Set and ScanNet++.** Our proposed SceneGen is capable of generating physically plausible 3D scenes featuring **complete structures**, **detailed textures**, and **precise spatial relationships**, demonstrating superior performance over prior methods in terms of both geometric accuracy and visual quality on both the synthetic and real-world datasets.

method consistently achieves the best performance across all metrics. This indicates that our synthesized assets are spatially closer to the ground truth while maintaining superior texture fidelity. and (iii) **efficiency**: While PartCrafter demonstrates a clear advantage in inference speed, it suffers from limited generation quality and controllability. In contrast, SceneGen achieves both superior quality and a strong balance between quality and efficiency, synthesizing a 3D scene containing four assets with geometry and textures within 2 minutes on a single A100 GPU.

In addition, while the baseline methods, *e.g.*, PartCrafter, DepR, and MIDI have been trained on 3D-FRONT [13], which may overlap with our test data, our SceneGen still consistently outperforms them across all metrics, further demonstrating its effectiveness and superiority.

4.3. Qualitative Results

Comparisons with baselines. As illustrated in Figure 4, we qualitatively compare SceneGen with existing baselines on both the 3D FUTURE [14] test set and in-the-wild ScnaNet++ [62], where they still struggle with 3D scene generation: PartCrafter lacks controllability over the generated targets and often mistakenly merges distinct assets, while both PartCrafter and DepR are limited to geometry generation and cannot render textures. More critically, all these methods exhibit difficulties in accurately understanding the spatial relationships among assets. In contrast, our proposed SceneGen precisely predicts the spatial relationships among assets and synthesizes multiple 3D assets with accurate geometry and high-quality textures, without relying on any additional tools or optimizations.

Extension to multi-image inputs. Benefiting from our architecture design, SceneGen can seamlessly handle multi-image inputs after being trained exclusively on single-image samples. Given the lack of suitable datasets for quantitative evaluation, we qualitatively assess the impact of multi-image inputs by randomly sampling several scenes from ScanNet++ [62] and employing SAM2 [45] to obtain

$\mathcal{F}^{\text{geo}}_{\text{global}}$	$\mathcal{F}^{V}_{\text{global}}$	$\mathcal{F}^{\text{mask}}_{i}$	$\mathcal{A}_{\text{SS}}$	Geometric Metrics					Image Category	Visual Metrics					
				CD-S↓	CD-O↓	F-Score-S↑	F-Score-O↑	IoU-B↑		PSNR↑	SSIM↑	LPIPS↓	FID↓	CLIP-S↑	DINO-S↑
✓	✓	✓	✓	**0.0118**	**0.0138**	**90.60**	**89.73**	**0.5818**	Scene	**16.76**	**0.8903**	**0.1417**	**19.59**	**0.9152**	**0.8322**
									GT-Render	**17.59**	**0.8991**	**0.1234**	**12.34**	**0.9236**	**0.8702**
✗	✓	✓	✓	0.0183	0.0266	83.33	74.71	0.4805	Scene	15.89	0.8845	0.1574	20.21	0.9049	0.8063
									GT-Render	16.27	0.8918	0.1421	15.36	0.9125	0.8420
✗	✗	✓	✓	0.0250	0.0286	79.08	73.46	0.4253	Scene	15.56	0.8806	0.1655	20.68	0.8980	0.7850
									GT-Render	15.86	0.8873	0.1511	16.62	0.9046	0.8187
✗	✗	✗	✓	0.0310	0.0290	75.20	73.17	0.3825	Scene	15.30	0.8773	0.1730	21.12	0.8932	0.7737
									GT-Render	15.55	0.8837	0.1591	17.45	0.9000	0.8076
✗	✗	✗	✗	0.0764	0.0352	54.21	70.55	0.1705	Scene	13.32	0.8418	0.2329	27.56	0.8399	0.6059
									GT-Render	13.39	0.8464	0.2217	28.61	0.8440	0.6362

Table 2. **Ablations on SceneGen Variants.** We progressively remove global geometric features ($\mathcal{F}^{\text{geo}}_{\text{global}}$), global visual features ($\mathcal{F}^{V}_{\text{global}}$), mask visual features ($\mathcal{F}^{\text{mask}}_{i}$), and substitute the scene-level self-attention ($\mathcal{A}_{\text{SS}}$) to validate each component's contribution to SceneGen.

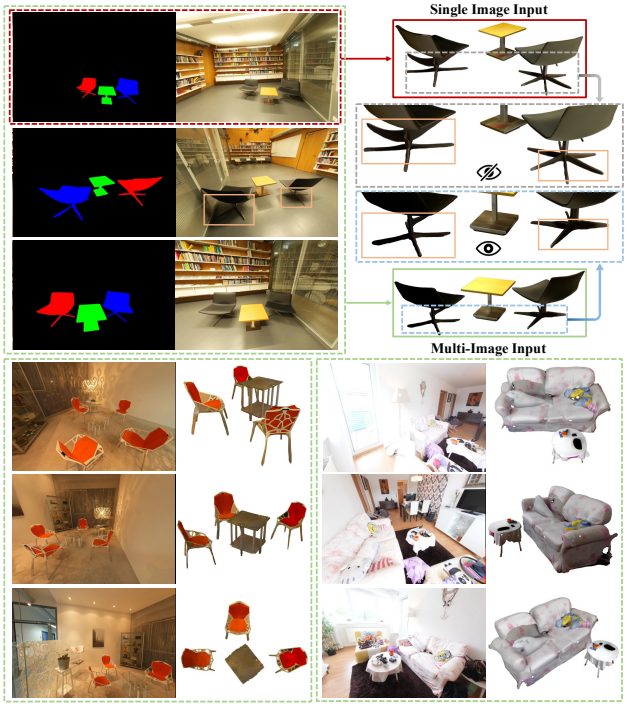

Figure 5. **Qualitative Results with Multi-view Inputs.** SceneGen can directly handle multi-view inputs in ScanNet++ and even achieves better generation quality, especially accurate structure.

segmentation masks of corresponding objects. As depicted in Figure 5, compared to single-image inputs, incorporating multi-view images leads to 3D assets with more complete geometry and finer texture details. This illustrates that SceneGen can adaptively integrate complementary information from multiple views to produce higher-quality 3D scenes, further validating its practicality and scalability. More qualitative results are included in the **Supplementary Material**.

4.4. Ablation Studies

To validate the efficacy of our modules, we conduct comprehensive evaluations on several variants of SceneGen, assessing both the geometric and visual quality of synthesized scenes. Concretely, we investigate the impact of gradually removing global geometric features ($\mathcal{F}^{\text{geo}}_{\text{global}}$), global visual features ($\mathcal{F}^{V}_{\text{global}}$), mask visual features ($\mathcal{F}^{\text{mask}}_{i}$), as well as substituting the scene-level self-attention block ($\mathcal{A}_{\text{SS}}$) with a simple asset-level self-attention block ($\mathcal{A}_{\text{AS}}$). As depicted in Table 2, we have the following observations: (i) Removing any of the aforementioned components degrades the overall performance, confirming their necessity in SceneGen; (ii) The geometric features primarily affect the structure of synthesized scenes, while the visual features further impact the visual quality; and (iii) The absence of scene-level self-attention blocks eliminates inter-asset interactions during generation, leading to notable performance declines across all metrics. These results strongly demonstrate the necessity and effectiveness of our proposed feature extraction and aggregation modules for SceneGen.

5. Conclusion

In this paper, we present **SceneGen**, a novel framework that takes a single scene image and target object masks as input to simultaneously synthesize multiple 3D assets with structure, texture, and relative spatial positions in a single feedforward pass. Specifically, we incorporate dedicated visual and geometric encoders to extract both asset-level and scene-level features, which are effectively fused with our proposed feature aggregation module. Notably, through our meticulous design, SceneGen can naturally generalize to multi-image inputs and achieve even better generation fidelity. Quantitative and qualitative evaluations demonstrate that SceneGen produces physically plausible and mutually consistent 3D assets, significantly outperforming previous methods in terms of generation quality and efficiency.

Acknowledgments

Weidi would like to acknowledge the funding from Scientific Research Innovation Capability Support Project for Young Faculty (ZY-GXQNJSKYCXNLZCXM-I22).

References

[1] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2

[2] PJ Besl and Neil D McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992. 6

[3] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3r: Estimating disentangled motion from dust3r without training. In *Proceedings of the International Conference on Computer Vision*, 2025. 2

[4] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2

[5] Tao Chu, Pan Zhang, Qiong Liu, and Jiaqi Wang. Buol: A bottom-up framework with occupancy-aware lifting for panoptic 3d scene reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2, 3

[6] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[7] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *Proceedings of the International Conference on Learning Representations*, 2024. 4

[8] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Conference on Neural Information Processing Systems*, 2023. 2

[9] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2

[10] Andreea Dogaru, Mert Özer, and Bernhard Egger. Gen3dsr: Generalizable 3d scene reconstruction via divide and conquer from a single view. In *International Conference on 3D Vision*, 2025. 3, 6

[11] Wenqi Dong, Bangbang Yang, Zesong Yang, Yuan Li, Tao Hu, Hujun Bao, Yuewen Ma, and Zhaopeng Cui. Hiscene: creating hierarchical 3d scenes with isometric view generation. In *ACM Multimedia*, 2025. 2, 3

[12] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. In *Conference on Neural Information Processing Systems*, 2023. 2, 3

[13] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the International Conference on Computer Vision*, 2021. 2, 7

[14] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 2021. 2, 5, 6, 7

[15] Daoyi Gao, Dávid Rozenberszki, Stefan Leutenegger, and Angela Dai. Diffcad: Weakly-supervised probabilistic cad model retrieval and alignment from an rgb image. *ACM Transactions On Graphics*, 2024. 2, 3

[16] Gege Gao, Weiyang Liu, Anpei Chen, Andreas Geiger, and Bernhard Schölkopf. Graphdreamer: Compositional 3d scene synthesis from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3

[17] Wei Gao and Russ Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 6

[18] Zeqi Gu, Yin Cui, Zhaoshuo Li, Fangyin Wei, Yunhao Ge, Jinwei Gu, Ming-Yu Liu, Abe Davis, and Yifan Ding. Artiscene: Language-driven artistic 3d scene generation through image intermediary. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 2, 3

[19] Can Gümeli, Angela Dai, and Matthias Nießner. Roca: Robust cad model retrieval and alignment from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3

[20] Xianglong He, Junyi Chen, Sida Peng, Di Huang, Yangguang Li, Xiaoshui Huang, Chun Yuan, Wanli Ouyang, and Tong He. Gvgen: Text-to-3d generation with volumetric representation. In *Proceedings of the European Conference on Computer Vision*, 2024. 2

[21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Conference on Neural Information Processing Systems*, 2017. 6

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Conference on Neural Information Processing Systems*, 2020. 2

[23] Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 2, 3, 6

[24] Zehuan Huang, Yuan-Chen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. In *Proceedings of the International Conference on Computer Vision*, 2025. 2, 6

[25] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural

wavelet-domain diffusion for 3d shape generation. In *ACM SIGGRAPH Asia Conference*, 2022. 2

[26] Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, et al. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025. 2

[27] Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Geo4d: Leveraging video generators for geometric 4d scene reconstruction. In *Proceedings of the International Conference on Computer Vision*, 2025. 2

[28] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 2

[29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations*, 2014. 2

[30] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *Proceedings of the European Conference on Computer Vision*, 2024. 2

[31] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *Proceedings of the International Conference on Learning Representations*, 2024. 2

[32] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2

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

[34] Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. In *Conference on Neural Information Processing Systems*, 2025. 2, 3, 6

[35] Lu Ling, Chen-Hsuan Lin, Tsung-Yi Lin, Yifan Ding, Yu Zeng, Yichen Sheng, Yunhao Ge, Ming-Yu Liu, Aniket Bera, and Zhaoshuo Li. Scenethesis: A language and vision agentic framework for 3d scene generation. *arXiv preprint arXiv:2505.02836*, 2025. 2, 3

[36] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proceedings of the International Conference on Learning Representations*, 2023. 4, 5

[37] Chang Liu, Haoning Wu, Yujie Zhong, Xiaoyun Zhang, Yanfeng Wang, and Weidi Xie. Intelligent grimm - open-ended visual storytelling via latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 2

[38] Xinhang Liu, Yu-Wing Tai, and Chi-Keung Tang. Agentic 3d scene generation with spatially contextualized vlms. *arXiv preprint arXiv:2505.20129*, 2025. 2, 3

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, 2019. 6

[40] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[41] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 2, 3, 6

[43] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the International Conference on Computer Vision*, 2023. 2, 3

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, 2021. 6

[45] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, et al. Sam 2: Segment anything in images and videos. In *Proceedings of the International Conference on Learning Representations*, 2025. 7

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[47] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[48] Fan-Yun Sun, Weiyu Liu, Siyi Gu, Dylan Lim, Goutam Bhat, Federico Tombari, Manling Li, Nick Haber, and Jiajun Wu. Layoutvlm: Differentiable optimization of 3d layout via vision-language models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 2, 3

[49] Xiang Tang, Ruotong Li, and Xiaopeng Fan. Towards geometric and textural consistency 3d scene generation via single image-guided model generation and layout optimization. *arXiv preprint arXiv:2507.14841*, 2025. 3

[50] Aether Team, Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, et al. Aether: Geometric-aware unified world modeling. In *Proceedings of the International Conference on Computer Vision*, 2025. 2

[51] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep

structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 2

[52] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 2, 3, 4

[53] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

[54] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 2

[55] Haoning Wu, Shaocheng Shen, Qiang Hu, Xiaoyun Zhang, Ya Zhang, and Yanfeng Wang. Megafusion: Extend diffusion models towards higher-resolution image generation without further tuning. In *Winter Conference on Applications of Computer Vision*, 2025. 2

[56] Haoning Wu, Ziheng Zhao, Ya Zhang, Weidi Xie, and Yanfeng Wang. Mrgen: Diffusion-based controllable data engine for mri segmentation towards unannotated modalities. In *Proceedings of the International Conference on Computer Vision*, 2025. 2

[57] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. In *Conference on Neural Information Processing Systems*, 2024. 2

[58] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 2, 3, 4, 5

[59] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. In *Proceedings of the International Conference on Learning Representations*, 2024. 2

[60] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3

[61] Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Wei Yang, Lan Xu, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image. In *ACM SIGGRAPH Conference*, 2025. 2, 3

[62] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision*, 2023. 7

[63] Huangyue Yu, Baoxiong Jia, Yixin Chen, Yandan Yang, Puhao Li, Rongpeng Su, Jiaxin Li, Qing Li, Wei Liang, Song-Chun Zhu, et al. Metascenes: Towards automated replica creation for real-world 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 2, 3

[64] Guangyao Zhai, Evin Pınar Örnek, Shun-Cheng Wu, Yan Di, Federico Tombari, Nassir Navab, and Benjamin Busam. Commonscenes: Generating commonsense 3d indoor scenes with scene graph diffusion. In *Conference on Neural Information Processing Systems*, 2023. 2, 3

[65] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics*, 2023. 2

[66] Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: A structured and explicit radiance representation for 3d generative modeling. In *Conference on Neural Information Processing Systems*, 2024. 2

[67] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *Proceedings of the International Conference on Learning Representations*, 2025. 2

[68] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions On Graphics*, 2024. 2

[69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 6

[70] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 2

[71] Qingcheng Zhao, Xiang Zhang, Haiyang Xu, Zeyuan Chen, Jianwen Xie, Yuan Gao, and Zhuowen Tu. Depr: Depth guided single-view scene reconstruction with instance-level diffusion. In *Proceedings of the International Conference on Computer Vision*, 2025. 6