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

Supplementary Material

Contents

A. Preliminaries on 3D Foundation Models

Given the inherent challenges of directly generating a 3D scene with multiple 3D assets from a single image, SceneGen aims to fully leverage the visual and geometric priors embedded in state-of-the-art 3D foundation models. Therefore, we build our model based on TRELLIS [12], and adopt DINOv2 [8] and VGGT [11] as our visual and geometric encoders, respectively. In the following, we provide a detailed introduction to TRELLIS and VGGT to better illustrate their roles.

TRELLIS. For a 3D asset ($\mathcal{O}$), TRELLIS encodes its geometry and appearance into a unified representation ($\boldsymbol{z}$), denoted as: $\boldsymbol{z} = \{(\boldsymbol{z}_i, \boldsymbol{p}_i)\}_{i=1}^{L}$. Here, $\boldsymbol{p}_i \in \{0, 1, \ldots, D-1\}^3$ denotes the positional index of an active voxel in the 3D grid intersecting the surface of $\mathcal{O}$, and $\boldsymbol{z}_i \in \mathbb{R}^C$ represents the local latent feature attached to the corresponding voxel, with D and L representing the 3D grid resolution and the total number of active voxels, respectively.

The generation process adopts two cascaded rectified flow models: the **sparse structure generator** ($\boldsymbol{\mathcal{G}}_\mathrm{S}$) synthesizes the sparse voxel structure $\{\boldsymbol{p}_i\}_{i=1}^{L}$, encoding geometric priors by predicting its low-resolution feature gird ($\boldsymbol{S}$); while the **structured latents generator** ($\boldsymbol{\mathcal{G}}_\mathrm{L}$) generates texture and appearance features $\{\boldsymbol{z}_i\}_{i=1}^{L}$ conditioned on $\{\boldsymbol{p}_i\}$. Both models are optimized via the conditional flow matching (CFM) [7] objective, which establishes straight probability paths between distributions through linear interpolation: $\boldsymbol{x}(t) = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}$, where $\boldsymbol{x}_0$ denotes data samples, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, and $t \in [0, 1]$. The velocity field ($\boldsymbol{v}(\boldsymbol{x}, t) = \nabla_t \boldsymbol{x}$) governs the reverse process, with the CFM objective formulated as:

$$\mathcal{L}_\mathrm{cfm}(\theta) = \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{\epsilon}} \|\boldsymbol{v}_\theta(\boldsymbol{x}, t) - (\boldsymbol{\epsilon} - \boldsymbol{x}_0)\|_2^2$$

Notably, the sparse structured generator ($\boldsymbol{\mathcal{G}}_\mathrm{S}$) learns rich geometric priors from large-scale 3D data, effectively capturing both object geometries and spatial relationships, thus delivering essential asset-level understanding capabilities. Our SceneGen is compatible with both the sparse structured generator and structured latents generator, thus can sequentially employ them to decode synthesized latent features into the geometry and texture of 3D assets.

VGGT. Trained on large-scale 3D annotated data, VGGT can extract 3D scene features through a purely feedforward network without explicit 3D inductive biases. For single or multi-view RGB inputs ($\{\boldsymbol{I}_i\}_{i=1}^{s}$), its aggregator derives scene geometric features ($\{\mathcal{F}_i^\mathrm{geo}\}_{i=1}^{s}$), represented as:

$$\{\mathcal{F}_i^\mathrm{geo}\}_{i=1}^{s} = \{[\mathcal{F}_G^\mathrm{geo}, \mathcal{F}_I^\mathrm{geo}]\}_{i=1}^{s} = \mathrm{VGGT}(\{\boldsymbol{I}_i\}_{i=1}^{s})$$

Here, $\mathcal{F}_G^\mathrm{geo}$ and $\mathcal{F}_I^\mathrm{geo}$ denote features extracted by *global self-attention* and *local self-attention* layers, respectively. These features are efficiently decoded by lightweight DPT layers [9] into depth maps, point maps, and tracks, validating their rich scene geometric representation capacity.

By integrating these complementary strengths, our **SceneGen** effectively captures both local asset-level and global scene-level features from the input image, achieving robust performance on the challenging 3D scene generation task.

B. More Details about Training Data

We train SceneGen on the 3D-FUTURE [3] dataset, leveraging its rich textures and diverse lighting conditions to simulate real-world environments and thereby enhance the model's generalization ability. Additionally, to ensure the model robustly learns the relative spatial relationships among multiple assets, we further scale up training data through data augmentation. Specifically, for a scene with N objects, we iteratively select each asset as the query asset and randomly shuffle the remaining ones during training. Considering GPU memory constraints, we set the maximum number of assets per scene to $N' = 7$ on a single A100 GPU. For samples containing more than N' assets, we randomly select a subset of N' assets for training. Furthermore, following TRELLIS [12], we apply its aesthetic score filtering criterion to exclude assets with aesthetic scores below 4.5, thereby ensuring high data quality. The distribution of asset counts across training scenes is illustrated in Figure 1.

C. More Implementation Details

This section provides a holistic explanation of implementation details discussed in the paper. Concretely, Sec. C.1 describes the specific strategies applied to extend SceneGen to multi-image inputs; and Sec. C.2 elaborates on our evaluation protocols.

C.1. Extension to Multi-image Inputs

While SceneGen is primarily designed for 3D scene generation based on a single scene image and trained exclusively on single-view images, it can be seamlessly adapted to multi-view inputs during inference with no need for additional training or fine-tuning. Concretely, during inference, our model can take images of the same scene from multiple viewpoints, along

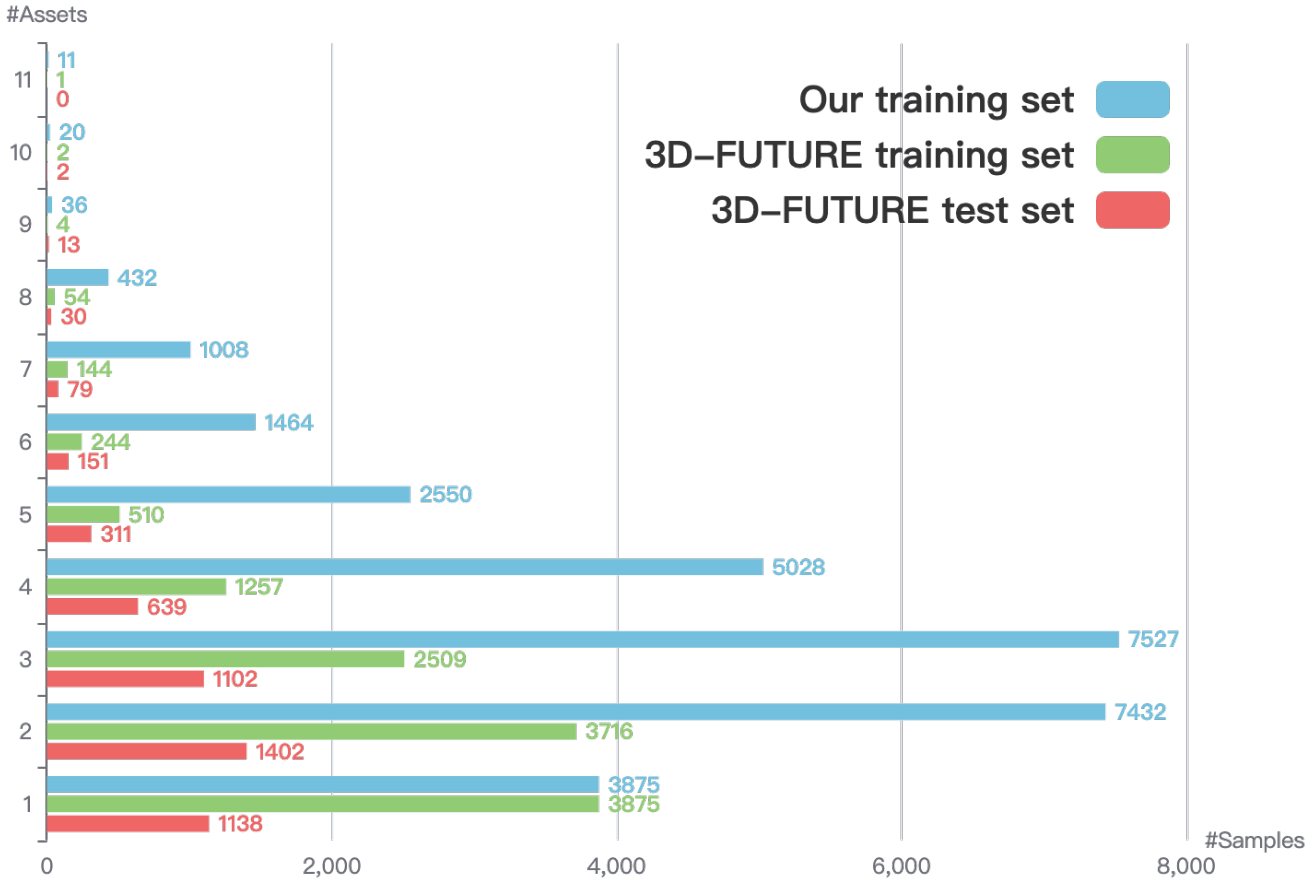

Figure 1. **Distribution of Asset Counts in our Training Data and Original 3D-FUTURE.**

with their corresponding objects and instance masks, as input. Within our SceneGen, the geometric encoder (Φ_G) (an off-the-shelf VGGT [11] aggregator) integrates geometric information across different viewpoints to produce better geometric representations for each perspective, thereby enabling SceneGen to synthesize more accurate geometric structures. Finally, we predict the relative positions among different assets from each viewpoint and use the mean of these predictions across all views as the final spatial position output. It is important to note that, to ensure correctness throughout the inference process, the input order of assets and their segmentation masks must remain consistent across all viewpoints.

Given the current lack of training and quantitative evaluation data for multi-view 3D scene generation, this work presents qualitative results on scenes sampled from ScanNet++ [13] to demonstrate the scalability of SceneGen, and leaves the construction of suitable multi-view datasets and evaluation methods for future work.

Method	Alignment	CD-S↓	CD-S 1↓	CD-S 2↓	F-Score-S↑	IoU-B↑
MIDI [5]	ICP	0.1697	0.0653	0.1044	41.64	0.1232
	FilterReg	0.0501	0.0278	0.0223	68.74	0.2493
SceneGen (Ours)	ICP	0.0310	0.0121	0.0189	83.74	0.5103
	FilterReg	**0.0118**	**0.0052**	**0.0066**	**90.60**	**0.5818**

Table 1. **Geometric Metrics Comparisons on Different Point Cloud Alignment Methods.**

C.2. Evaluation Protocols

Geometric metrics. Following previous work [5], we conduct geometry evaluation in normalized 3D space (also referred to as canonical space, *i.e.*, $x, y, z \in [-1, 1]$), where the ground truth and the synthesized query asset are first rigidly aligned using point cloud registration algorithms. Unlike MIDI [5], which relies on the traditional Iterative Closest Point (ICP [1]) method

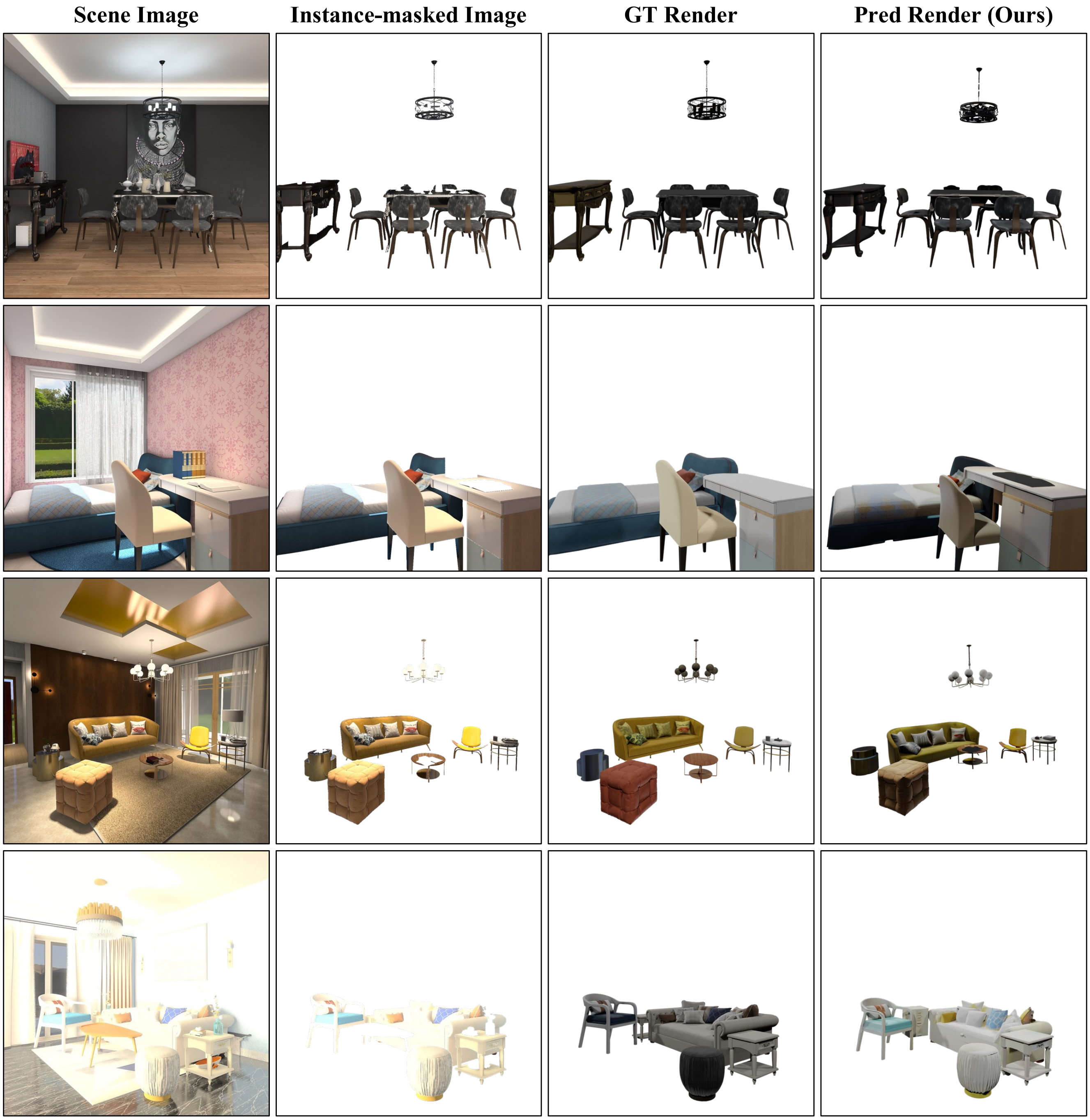

Figure 2. **Examples of Visual Metrics Evaluation Protocols.** Here, we present two complementary types of ground truth: instance-masked images may introduce slight differences due to potential occlusions, while GT-render images lack scene-level illumination.

prone to suboptimal alignment results, we employ FilterReg [4], a faster and more robust point cloud alignment approach. As presented in Table 1, both MIDI and SceneGen achieve better overall performance when aligned via FilterReg, demonstrating the reliability of this alignment method compared to traditional ICP. Moreover, under both alignment strategies, SceneGen consistently outperforms MIDI, indicating that explicitly predicting the spatial positions among assets enables SceneGen to more accurately model the relationships among distinct 3D assets within the scene.

Visual metrics. Beyond the commonly used geometric evaluations described above, we also consider several visual metrics

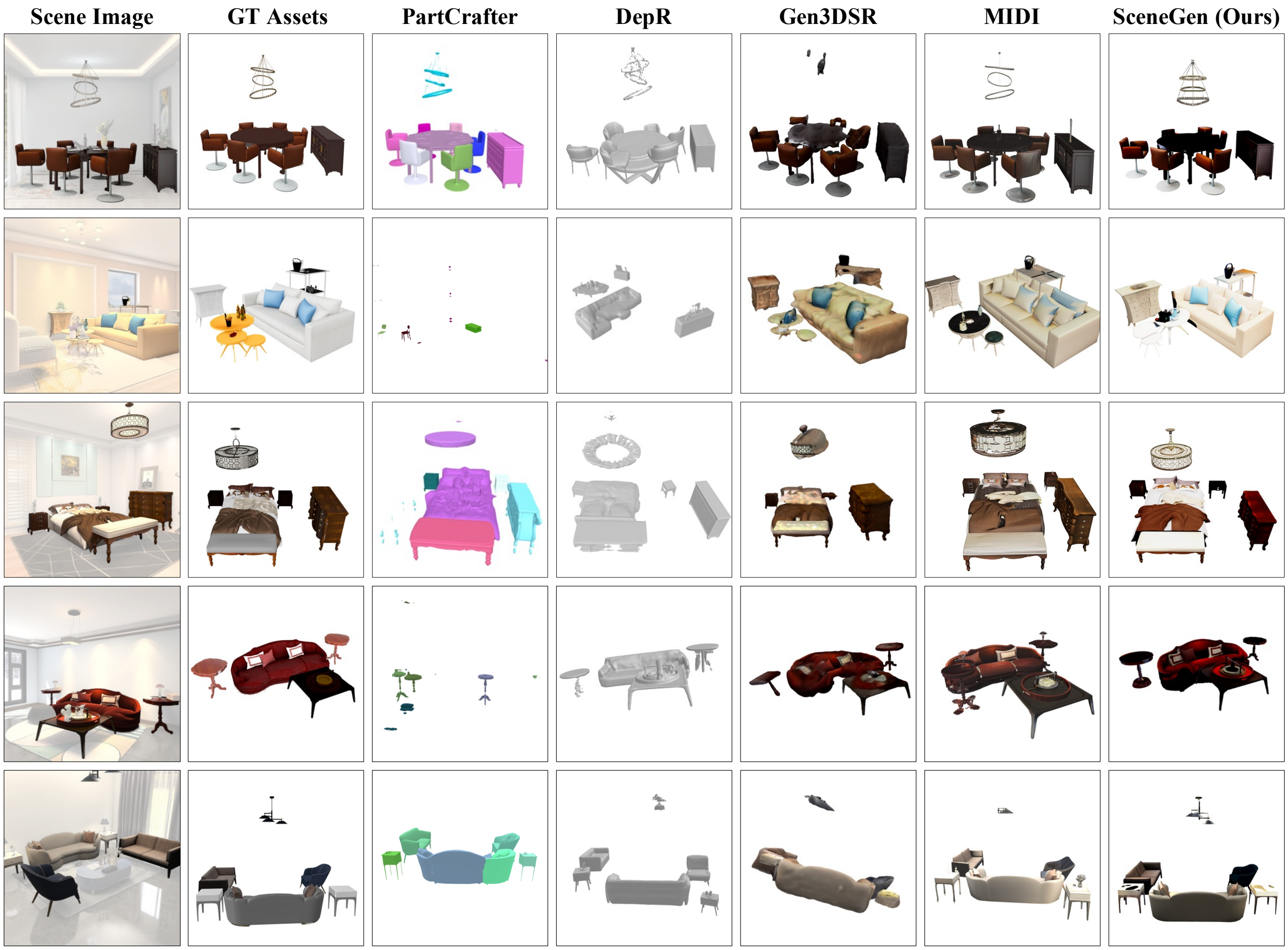

Figure 3. **More Qualitative Comparisons on the 3D FUTURE Test Set.**

to assess the visual quality of generated scenes. Concretely, after aligning the synthesized point clouds with the ground truth scenes, we use *Blender* to render them with the identical camera parameters. The rendered images are then compared with two types of ground truth to compute perceptual metrics that reflect the visual quality of synthesized scenes. As illustrated in Figure 2, these include: (i) instance-masked scene images, which are extracted using the corresponding object masks, where the occlusion relationships between assets introduce differences relative to predicted renderings; and (ii) GT-Render images, which are rendered from the ground truth assets at the same viewpoint using *Blender*, but lack scene-level illumination and complete textures, resulting in textural discrepancies compared to predicted scenes. Thus, by computing visual metrics against both types of ground truth, we provide a complementary evaluation of the visual quality of synthesized scenes.

Efficiency. To ensure a fair comparison across all methods, we report the average inference time over 500 trials of synthesizing scenes with a single asset on a single A100 GPU. Notably, our proposed SceneGen can directly generate 3D scenes containing 4 assets in a single feedforward pass within 2 minutes on the same hardware, eliminating the need for time-consuming sequential generation of individual 3D assets.

D. More Visualizations

This section presents additional qualitative results on the 3D-FUTURE [3] test set, offering a detailed comparison between our SceneGen and representative baselines. As depicted in Figure 3, we have the following observations: (i) PartCrafter [6] frequently suffers from missing or mixed-up assets due to its inability to control generation via object masks, despite already taking segmented objects and asset counts as input; (ii) Both PartCrafter and DepR [14] can only generate scene geometry without rendering texture details; and (iii) All baseline methods (PartCrafter [6], DepR [14], Gen3DSR [2], and MIDI [5]) share the common limitation of incorrect spatial relationships among synthesized assets. In contrast, our SceneGen fully integrates visual and geometric features within the scene to enable mutual influence among multiple assets during generation, producing 3D scenes with physically plausible geometry and high-quality texture details.

E. Limitations & Future Works

E.1. Limitations

While our SceneGen demonstrates superior performance in 3D scene generation, it is not without its limitations.

Limited to Indoor Generation. While SceneGen demonstrates better texture generation and generalization capabilities compared to previous methods that rely on canonical representations, the narrow training data distribution limits its ability to generalize to non-indoor scenes, restricting its generalization to a broader range of environments.

Asset Collisions and Overlaps. Although SceneGen can generate multiple 3D assets and relative spatial positions in a single feedforward pass, without relying on complex post-processing, it does not always handle contact relationships among objects, occasionally leading to asset overlaps or geometric inconsistencies. This is mainly because our single-stage framework does not explicitly enforce strict spatial or physical constraints among objects.

Reliance on Segmentation Masks. SceneGen inherently requires segmentation masks of the target objects as input. In our current framework, we leverage either ground truth masks or masks pre-extracted by an off-the-shelf SAM 2 [10]. This reliance limits the flexibility of applying SceneGen directly to in-the-wild data to some extent and may potentially result in a lack of robustness against low-quality segmentation masks.

E.2. Future Works

To address the aforementioned limitations of SceneGen, we propose several directions for future improvement: (i) Constructing larger-scale 3D scene generation datasets that cover more diverse indoor and outdoor scenarios, to address biases in training data distribution and improve the generalization ability of models; (ii) Building suitable multi-view scene generation datasets to expand the application scope and practical potential of existing models; (iii) Incorporating explicit physical priors or constraints to facilitate the model to better learn complex interactions among objects; and (iv) Introducing an additional object segmentation module into the current framework or natively integrating the capability to segment assets from scenes.

References

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

[2] 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. 5

[3] 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

[4] 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. 4

[5] 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. 3, 5

[6] 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. 5

[7] 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. 2

[8] 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

[9] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[10] 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. 6

[11] 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

[12] 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

[13] 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. 3

[14] 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. 5