Points-to-3D: Structure-Aware 3D Generation with Point Cloud Priors

Jiatong Xia[*], Zicheng Duan[*], Anton van den Hengel, Lingqiao Liu[†]
Australian Institute for Machine Learning, University of Adelaide, Australia
Project page:

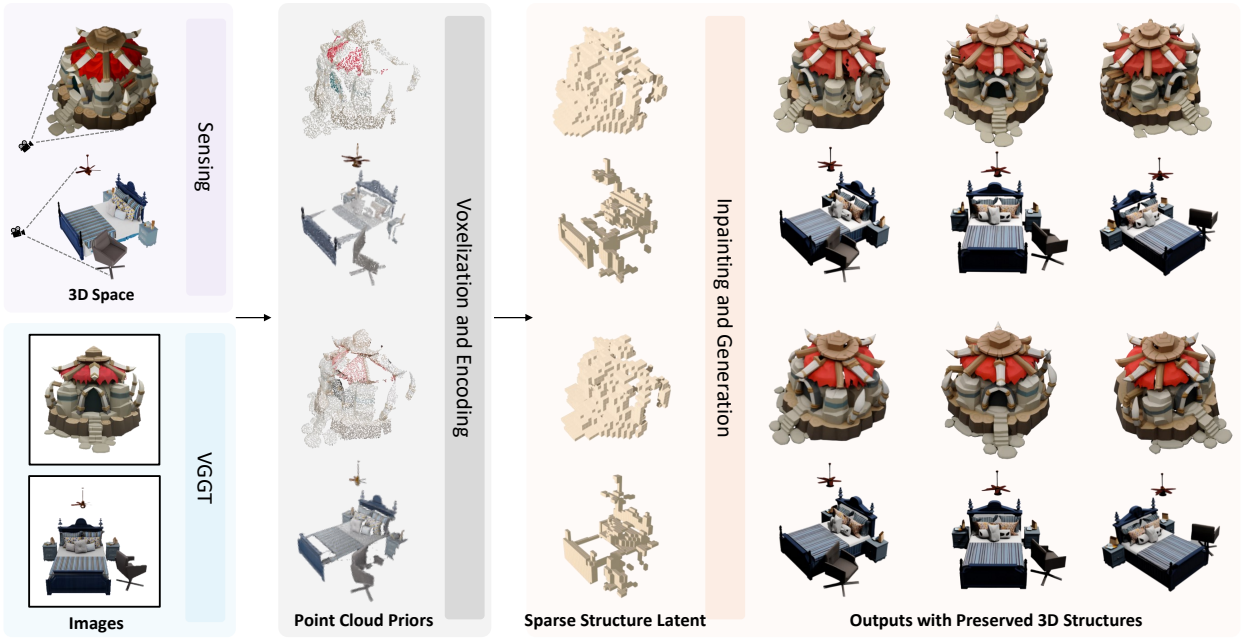

Figure 1. We introduce explicit 3D point cloud priors into 3D generation framework, given a pre-existing point cloud or a feed-forward point cloud prediction from image input, our model generates high-quality 3D assets that faithfully preserve the observed structure while plausibly completing unobserved regions with coherent geometry.

Abstract

Recent progress in 3D generation has been driven largely by models conditioned on images or text, while readily available 3D priors are still underused. In many real-world scenarios, the visible-region point cloud are easy to obtain—from active sensors such as LiDAR or from feed-forward predictors like VGGT—offering explicit geometric constraints that current methods fail to exploit. In this work, we introduce Points-to-3D, a diffusion-based framework that leverages point cloud priors for geometry-controllable 3D asset and scene generation. Built on a latent 3D diffusion model TRELLIS, Points-to-3D first replaces pure-noise sparse structure latent initialization with a point cloud priors tailored input formulation. A structure inpainting network, trained within the TRELLIS framework on task-specific data designed to learn global structural inpainting, is then used for inference with a staged sampling strategy (structural inpainting followed by boundary refinement), completing the global geometry while preserving the visible regions of the input priors. In practice, Points-to-3D can take either accurate point-cloud priors or VGGT-estimated point clouds from single images as input. Experiments on both single-object and multi-object scenarios consistently demonstrate superior performance over state-of-the-art baselines in terms of rendering quality and geometric fidelity, highlighting the effectiveness of explicitly embedding point-cloud priors for achieving more accurate and structurally controllable 3D generation.

1. Introduction

Advances in 3D generation now allow models to synthesize realistic and diverse 3D assets from single-view

[*]Jiatong Xia and Zicheng Duan equally contributed to this work.
[†]Corresponding author, e-mail: lingqiao.liu@adelaide.edu.au

images or text prompts. These "foundation" 3D models [19, 28, 62, 65, 73] can produce 3D assets across broad categories, supporting applications in content creation and virtual environments. However, conditioning solely on 2D images or text provides limited geometric controllability: while the output may appear plausible, the model lacks any mechanism to respect real 3D measurements. In practice, partial point clouds from sensors or image-based predictors provide reliable geometry for visible regions, yet current 3D generative pipelines make little use of this readily available structural information.

In this work, we address this gap by enabling geometry-controllable 3D generation driven by point cloud priors. We focus on the setting where a visible-region point cloud—captured or predicted—is treated as a hard structural constraint, requiring the generated asset to align with observed geometry while plausibly completing unobserved parts. Achieving this cannot be done by simply injecting the point cloud as an additional condition; it requires to integrates the structural prior into the latent space itself.

Recent latent 3D diffusion models [20, 28, 54], represented by TRELLIS [62], factorize 3D generation into two stages: a coarse structural stage operating on a sparse occupancy representation, followed by a semantic and appearance refinement stage. This paradigm offers an explicit structure latent that could be guided by 3D priors. Yet in their default formulation, these structure latents are initialized purely from Gaussian noise and rely only on text or image embeddings, making them unable to anchor structural generation to actual 3D observations.

To overcome this limitation, we introduce Points-to-3D, a point-cloud–guided 3D generation framework that re-defines how the structural latent is initialized and completed. Instead of starting from pure noise, we voxelize the visible point cloud and encode it with the TRELLIS sparse structure VAE to obtain a partially observed latent that directly reflects the measured geometry. Regions supported by observations are preserved as fixed constraints, while unobserved regions remain free to be synthesized. A mask-aware inpainting network completes this mixed latent, enabling the model to generate coherent structures that respect real 3D measurements while plausibly filling missing areas.

To support this formulation, we construct a visibility-aware training pipeline that first produces realistic partial–complete structure pairs from ground-truth assets, these pairs supervise the inpainting model to generate geometric cues from visible regions to invisible ones while maintaining consistency with the input point cloud. During inference, Points-to-3D adopts a lightweight two-stage procedure: it first establishes a globally consistent structure under visibility constraints, and then performs a brief refinement step to enhance boundary quality without disturbing anchored geometry. This design enables controllable and structurally faithful 3D generation from both sensor-captured and image-predicted point clouds.

We evaluate Points-to-3D on object-level (Toys4K [51]) and scene-level (3D-FRONT [13]) benchmarks. Across all settings, our method consistently outperforms TRELLIS [62] and other baselines in rendered-view quality and geometric fidelity. Gains are especially significant in regions covered by point-cloud priors, where Points-to-3D achieves near-perfect alignment while maintaining realistic completions in unseen areas. Furthermore, combining our point-cloud–anchored structure generation with text conditioning enables controllable text-to-3D generation guided by concrete 3D measurements.

2. Related Work

2.1. 3D Modeling

Recovering the 3D model of specific scenes or objects is a fundamental problem in computer vision and graphics. Classical 3D reconstruction leverages multi-images to recover geometry, including Structure-from-Motion (SfM) [44], Multi-View Stereo (MVS) [66, 67, 69], SDF-based approaches [39, 49, 71], e.t.c. Radiance-field models like Neural Radiance Fields (NeRF) [1, 2, 5, 25, 27, 36, 37] and 3D Gaussian Splatting (3DGS) [18, 21, 24, 30, 70] further enable high-fidelity reconstruction and novel-view synthesis after scene-specific optimization, and recent feed-forward variants [4, 7, 8] reducing the per-scene cost. DUSt3R-related feed-forward reconstruction methods [29, 55, 59, 63, 72], exemplified by VGGT [58], predict per-pixel point cloud and implicitly handle camera poses, achieving strong performance even with a single image. However, recovering 3D assets from only one image is still beyond the capabilities of reconstruction methods. 3D generative models [20, 28, 54, 62] effectively address this scenario: conditioned on a single reference image or even text prompt, they can synthesize plausible 3D assets aligned with the reference content.

2.2. 3D Generative Models

Prior 3D generation relies on GANs [3, 16, 45] produce convincing results, yet their instability restricts scalability and output diversity. models [17, 32, 50], starting with 2D generation [40, 43, 57], diffusion-based methods rapidly expanding across a spectrum of 3D representations [19, 20, 28, 33, 46, 48, 53, 54, 65, 73]. Recently, TRELLIS [62] introduces a novel latent representation that enables decoding into versatile 3D output formats, demonstrating strong quality, versatility, and editability, and offering a superior paradigm and framework for 3D generation. Subsequent works [31, 35, 60, 64] have leveraged TRELLIS to implement a wide range of practical applications. Nevertheless, most existing improvements focus on

enhancing performance at the reference-conditioning level, while directly embedding 3D priors into latent initialization to enable more reliable generation still remains largely unexplored in 3D generation.

2.3. Point Cloud Priors

Incorporating 3D priors to assist downstream tasks has proven to be a effective strategy. Where point cloud stand out as one of the most practical and informative representations. Leveraging point cloud priors has advanced a wide range of 3D perception tasks [47, 61, 75] as well as reconstruction tasks [12, 42]. In particular, visible-region point clouds are easy to obtain from diverse sources, including active sensors such as LiDAR and structured-light depth cameras—now widely accessible even on mobile devices—or from reconstruction approaches such as VGGT [58]. Integrating these easily obtainable point cloud into 3D generative models offers promising potential for explicit geometry control and accurate modeling of complex multi-object scenes. This work seeks to establish a simple yet effective paradigm for incorporating point cloud priors into a diffusion-based 3D generation framework.

2.4. Inpainting

Inpainting is a common paradigm for completing missing content while preserving observed structures. In 2D vision, diffusion-based inpainting methods [6, 22, 34, 68] use spatial masks to guide the synthesis of occluded regions, achieving coherent and controllable image completion. Similar ideas appear in 3D completion [9, 10, 15, 23] where partial scans are used to infer full geometry, but such approaches usually operate as separate completion modules rather than within a generative framework. TRELLIS, however, has the potential to perform inpainting directly within its sparse structured latent spaces, enabling improved global coherence without the need for external modules. Building on this perspective, we formulate point-cloud–conditioned 3D generation as a latent inpainting problem, allowing observed geometry to be embedded and completed naturally within the generative process without relying on auxiliary completion components.

3. Method

We seek to achieve geometry-controllable 3D generation by conditioning on point clouds, whether captured by real-world sensors or inferred from a single image via feed-forward prediction. This section first outlines TRELLIS, the baseline that underpins our work, then formalizes the problem and introduces our method, detailing the model architecture, training-data construction, and sampling strategy.

3.1. Preliminaries: TRELLIS

TRELLIS [62] is a recently proposed 3D generation model that produces diverse and high-fidelity 3D assets from image or text prompts. Unlike conventional diffusion models that operate directly in voxel or implicit-function space, TRELLIS performs diffusion in a compact latent space specifically designed to encode 3D structure and appearance. This latent space is learned via a pair of variational autoencoders (VAEs) trained to compress and reconstruct 3D assets. The first VAE encodes voxelized 3D features derived from the original asset into a latent representation called the *structured latent* (SLAT), denoted as $\mathbf{z} = \{(\mathbf{z}_i, \mathbf{p}_i)\}_{i=1}^{L}$, where $\mathbf{z}_i \in \mathbb{R}^{c_{\text{slat}}}$ is a local feature attached to voxel position $\mathbf{p}_i \in [0, N-1]^3$ with $N = 64$. The SLAT $\mathbf{z}$ can be decoded into multiple 3D output formats—Gaussian splats, radiance fields, or meshes—through corresponding decoders, enabling flexible rendering backends. The second VAE $(\mathcal{E}_s, \mathcal{D}_s)$ learns a compact representation of geometry by encoding a binary voxel occupancy grid $\mathbf{M} \in \{0,1\}^{N\times N\times N}$—whose occupied positions correspond to $\{\mathbf{p}_i\}_{i=1}^{L}$—into a *sparse structure* (SS) latent $\mathbf{q} \in \mathbb{R}^{r\times r\times r\times c_s}$ (with $r = 16$), which can be decoded back into $\mathbf{M}$.

The generation process in TRELLIS proceeds in two stages following a coarse-to-fine paradigm. In the **Structure Generation** stage, a Flow Transformer $\mathcal{G}_s$ takes Gaussian noise $\boldsymbol{\epsilon}_{\text{s}} \sim \mathcal{N}(0, \mathbf{I})$ and a condition embedding $\mathbf{c}$ (from image or text) to sample the SS latent $\mathbf{q}$, which is then decoded by $\mathcal{D}_s$ into a binary voxel grid $\mathbf{M}$, defining the asset's geometric scaffold. In the subsequent **Structured Latent Generation** stage, a Sparse Flow Transformer $\mathcal{G}_l$ takes noise $\boldsymbol{\epsilon}_{\text{slat}} \sim \mathcal{N}(0, \mathbf{I})$, the voxel positions $\{\mathbf{p}_i\}_{i=1}^{L}$, and the same condition $\mathbf{c}$ to generate the SLAT $\mathbf{z}$, which is decoded into the final 3D asset with texture and semantics. Overall, TRELLIS establishes a two-level generative hierarchy that first synthesizes a sparse geometric structure and then enriches it with detailed appearance.

While the VAEs in TRELLIS possess the intrinsic ability to encode meaningful 3D geometry, the generative process itself is not conditioned on external 3D information. Our approach, **Points-to-3D**, leverages this encoded structural capability by directly injecting point-cloud priors into the VAE latent space, thereby grounding the diffusion process to explicit 3D observations.

3.2. Problem Formulation

In many real-world settings, we aim to generate 3D assets conditioned on point cloud priors—obtained either via active sensing (e.g., LiDAR) or model prediction (e.g., VGGT). These point cloud typically cover only the visible portion of the scene. In such a case, the goal is to use the visible-region point cloud $\mathbf{P}$ as a prior for geometry-controllable 3D asset generation: preserving the observed foreground structure while completing unobserved regions guided by foreground cues. To this end, we cast the task as inpainting conditioned on $\mathbf{P}$, inferring missing geometry

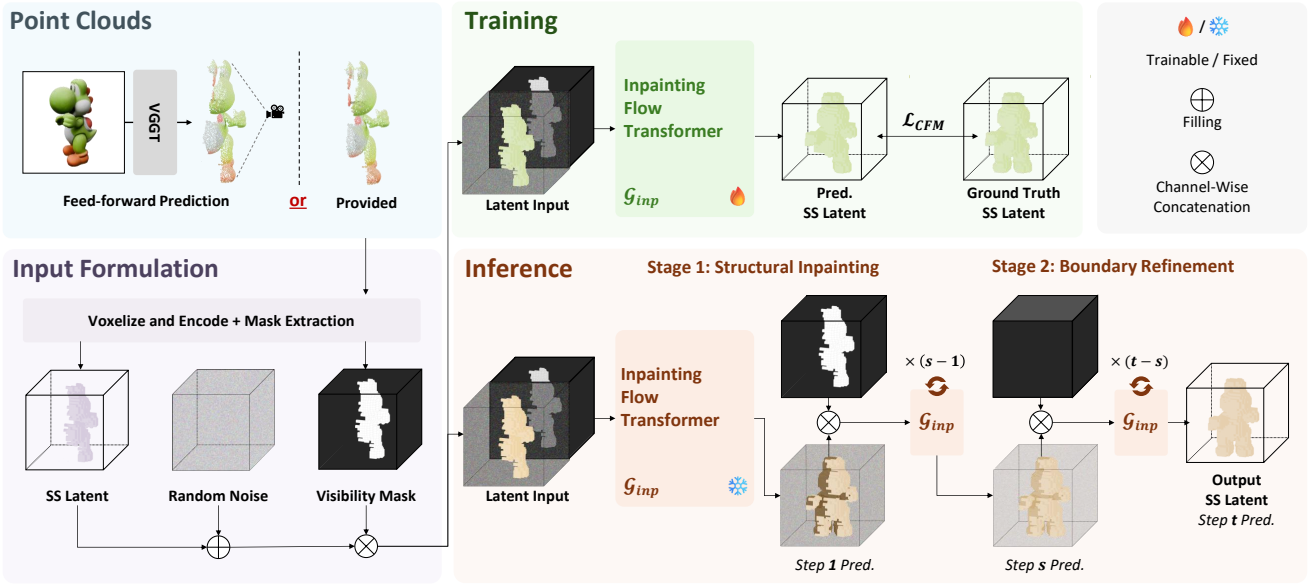

Figure 2. **Overall framework.** Given point cloud priors—either pre-existing or predicted by VGGT from input image—we first voxelize and VAE-encode it to obtain an SS latent, where the empty regions are filled with random noise and concatenated with an extracted mask to form the input paradigm for our model. During training, the input training data is fed into our inpainting flow transformer $\mathcal{G}_{inp}$, which is optimized via a conditional flow matching loss. During inference, the input test data is processed by the trained $\mathcal{G}_{inp}$ through a two-stage sampling procedure: (1) a structural inpainting stage with s sampling steps to inpaint the global structure. And (2) a boundary refinement stage with remaining $(t-s)$ steps to refine the inpainting boundaries, yielding the final output SS latent.

from the surrounding latent context.

Specifically, unlike TRELLIS—which initializes generation from pure noise $\boldsymbol{\epsilon}_{\text{s}}$—our structural inpainting stage begins by voxelizing the visible point-cloud priors $\mathbf{P}$ into a binary 3D occupancy grid $\mathbf{M}' \in \{0,1\}^{N\times N\times N}$. This voxelized structure is then encoded with the VAE encoder $\mathcal{E}_s$ to obtain the initial SS latent $\mathbf{q}_{\text{vis}} \in \mathbb{R}^{r\times r\times r\times c_{\text{s}}}$ (with $r=16$), which serves as the generation starting point. Formally:

$$\mathbf{q}_{\text{vis}} = \mathcal{E}_s(\mathbf{M}'). \tag{1}$$

To indicate which SS latent regions should be preserved, we derive an occupancy mask $\mathbf{m}_{\text{s}} \in \mathbb{R}^{r\times r\times r\times c_{\text{m}}}$ by downsampling $\mathbf{M}'$ to the latent resolution. Then, we preserve the visible region SS latent with $\mathbf{m}_{\text{s}}$ and fill the remaining with noise to obtain the inpainting input SS latent $\mathbf{q}_{\text{comb}}$:

$$\mathbf{q}_{\text{comb}} = \mathbf{m}_{\text{s}} \odot \mathbf{q}_{\text{vis}} + (1-\mathbf{m}_{\text{s}}) \odot \boldsymbol{\epsilon}_{\text{s}}. \tag{2}$$

Ultimately, we aim to build an inpainting model $\mathcal{G}_{inp}$ based on the structure generation model $\mathcal{G}_s$ to take $\mathbf{q}_{\text{comb}}$ as input, and inpaint the final SS latent $\mathbf{q}$, facilitating visible regions geometry-controllable generation.

3.3. Point Clouds Priors Driven Generative Model

Model design. As shown in the purple box of Fig. 2, to enforce the inpainting model, $\mathcal{G}_{inp}$, on distinguishing the regions to preserve and generate, we further concatenate the mask $\mathbf{m}_{\text{s}}$ to the $\mathbf{q}_{\text{comb}}$ along the channel dimension. This turn $\mathbf{q}_{\text{comb}}$ to $\mathbf{x}_{\text{inp}}$:

$$\mathbf{x}_{\text{inp}} = \text{Concat}[\mathbf{q}_{\text{comb}}, \mathbf{m}_{\text{s}}], \mathbf{x}_{\text{inp}} \in \mathbb{R}^{r\times r\times r\times (c_{\text{s}}+c_{\text{m}})} \tag{3}$$

To adapt $\mathcal{G}_{inp}$ with more input channels, we simply replace its input layer inherited from $\mathcal{G}_s$ by a newly registered projection layer with channel dimension $(c_{\text{s}}+c_{\text{m}})$, and maintain all other network structures unchanged. Then, we fully finetune $\mathcal{G}_{inp}$ to learn to inpaint a completed sparse structure latent $\mathbf{q}_{\text{pred}} \in \mathbb{R}^{r\times r\times r\times c_{\text{s}}}$ using Conditional Flow Matching loss (CFM) and regard the ground-truth sparse latent $\mathbf{q}_{\text{gt}}$ as supervision. This can be formulated as:

$$\mathcal{L}_{CFM} = \mathbb{E}_{t,\mathbf{q}_{\text{gt}},\epsilon} \left\| \mathcal{G}_{inp}(\mathbf{x}_{\text{inp}}, t) - (\epsilon - \mathbf{q}_{\text{gt}}) \right\|_2^2 \tag{4}$$

Note that the condition c and time-dependent noise scheduling for $\mathbf{x}_{\text{inp}}$ are omitted for simplicity.

Training data from visible point clouds. We construct diverse pairs of training data from visible point clouds together with their corresponding ground-truth sparse structure latent to train our model as illustrated in Fig. 3. The main challenge lies in accurately obtaining the visible-region point clouds corresponding to the input condition images of each 3D asset. To achieve this, we render the depth map $\mathbf{D}_t$ with height and width as H and W from T viewpoints with the condition images $\mathbf{I}_t$ for each ground-truth 3D asset. For each object, we first uniformly sample S surface points $\hat{\mathbf{P}} = \{\hat{\mathbf{p}}_i = (u_i, v_i, w_i)\}_{i=1}^{S}$, and given the

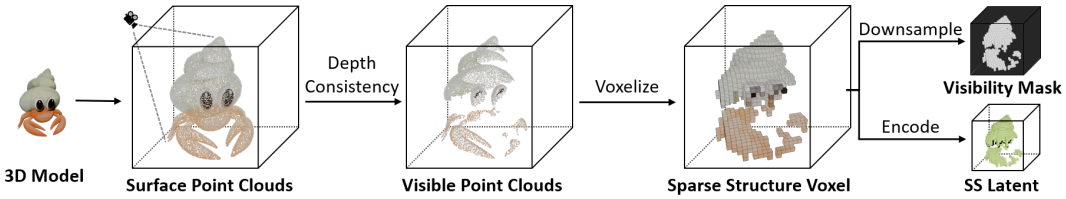

Figure 3. **Training data processing.** We preserve the visible portion of the complete point cloud and convert it into training inputs.

world-to-camera transformation $\mathbf{T}_t = [\mathbf{R}_t \mid \mathbf{t}_t]$ for the t-th view, each point is transformed to the camera as:

$$\hat{\mathbf{p}}_i^t = \mathbf{R}_t \left(\hat{\mathbf{p}}_i - \mathbf{t}_t\right) = \left(u_i^t, v_i^t, w_i^t\right)^\top \tag{5}$$

The corresponding image-plane projection $\mathbf{u}_t \in [1, H] \times [1, W]$ is computed using the intrinsic matrix $\mathbf{K}$. We apply an observation mask $\mathbf{O^t}$ to indicate which point is considered visible in view t if its projected depth w_i^t is consistent with the rendered depth within a tolerance threshold τ:

$$\mathbf{O}_i^t = \begin{cases} 1, & \text{if } |\mathbf{D}_t\left(\mathbf{u}_i\right) - w_i^t| < \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The visible point cloud $\mathbf{P}_t$ for view t is thus obtained as $\mathbf{P}_t = \{\hat{\mathbf{p}}_i^t \mid \mathbf{O}_i^t = 1\}$. Each $\mathbf{P}_t$ is then voxelized into the sparse structure voxel, which is then encoded and calculated to the SS latent $\mathbf{q}_{\text{comb}}^t$. Simultaneously, the downsampled occupancy mask $\mathbf{m}_{\text{s}}^t$ is obtained to indicate the visible region of the obtained SS latent. The ground-truth SS latent $\mathbf{q}_{\text{gt}}$ is derived from the complete 3D structure of the object. Consequently, the samples $(\mathbf{q}_{\text{comb}}^t, \mathbf{m}_{\text{s}}^t, \mathbf{I}_t, \mathbf{q}_{\text{gt}})$ are used to supervise the model $\mathcal{G}_s$ (green box in Fig 2) to learn structure completion from visible-region priors.

3.4. Staged Sampling from Point Cloud Priors

During inference, we split the t step generation into two separate sampling stages, namely the structural inpainting stage and the boundary refinement stage. As illustrated in Fig. 2 orange box, the first stage produces a coarse but globally consistent skeleton structure guided by the visible point clouds using inpainting, while the second stage refines the boundary regions that connect newly generated content to the predefined visible areas. Specifically, in each sampling step of the structural inpainting stage, the trained model outputs $\mathbf{q}_{\text{pred}}$ and reconstructs the inpainting input $\mathbf{x}_{\text{inp}}$ for the next iteration by concatenating $\mathbf{q}_{\text{pred}}$ with the visibility mask $\mathbf{m}_{\text{s}}$ following Eq. (3). We repeat this process for s steps to obtain a draft skeleton 3D structure that is mostly coherent with the visible point cloud. However, slight inconsistencies and missing details may appear around the boundary regions between generated and predefined visible areas, mainly due to information loss introduced during downsampling. To address this, we define the latter $(t - s)$ steps as the boundary refinement stage. Here, we replace the visibility mask $\mathbf{m}_{\text{s}}$ with an all-ones mask $\mathbf{m}_1$, effectively converting inpainting into standard denoising. This allows the model to refine details on either side of the masked or unmasked regions without drastically modifying the existing global geometry, resulting in a fully completed and high-quality sparse structure.

4. Results

4.1. Experiments Setup

Datasets. We train our model on a combination of three datasets: 3D-FUTURE [14] dataset, HSSD [26] dataset and ABO [11] dataset. During training, we render $T = 24$ input views for each object and sample $S = 50,000$ point clouds from each object mesh. For each views, we compute the corresponding visible point clouds, which is then process to an initial SS latent used for training. We evaluate our model on two types of test datasets: the single-object dataset Toys4K [51] and the multi-object dataset 3D-FRONT [13]. And our method is tested under two settings to cover a broader range of application scenarios. In the first setting, we use the sampled visible point cloud from each view as available input. And in the second setting, where no preprocessed point cloud priors are provided, the test view condition image is fed into VGGT to obtain an estimated point cloud for our model as the initial point cloud priors. We also evaluate our method on several real-world images from the Pix3D [52] dataset.

Evaluation metrics. We evaluate the final generation results in two aspects. For the rendered images of the generated 3D assets, we assess the image quality by comparing with those rendered from the ground-truth 3D asset, and using PSNR, SSIM, LPIPS [74], and DINO [38] feature similarity as evaluation metrics. For the geometric quality, we employ Chamfer Distance and F-score, as well as the rendered normal maps PSNR and LPIPS as evaluation metrics. For text-to-3D generation evaluation, we use the CLIP [41] score to measure the consistency between the generated results and the input text prompts.

Implementation. We train our model for 20k iterations with a batch size of 8 on 4 Nvidia A100 GPUs, and following the other TRELLIS [62]'s sparse structure flow transformer's training setting. During inference, we set the $t = 50$ sampling steps for the trained Sparse Structure Flow Transformer, allocating $s = 25$ steps for structural inpainting and the remaining steps for refinement. For the other comparison methods, we use their official code and settings

Method	Rendering				Geometry			
	PSNR ↑	SSIM(%) ↑	LPIPS ↓	DINO(%) ↓	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
GaussianAnything [28]	20.08	89.31	0.183	26.74	0.084	0.513	20.99	0.199
Real3D [20]	19.55	90.65	0.169	27.65	0.065	0.574	21.31	0.178
LGM [54]	20.55	89.98	0.181	23.45	0.075	0.487	20.04	0.202
VoxHammer [31] (3D Inversion)	20.51	90.01	0.123	15.10	0.046	0.724	20.28	0.158
TRELLIS [62]	21.94	91.46	0.105	7.82	0.034	0.832	23.81	0.105
Points-to-3D (Ours-VGGT Esti.)	22.55	92.09	0.088	7.37	0.024	0.881	24.53	0.085
Points-to-3D (Ours-P.C.Priors)	**22.91**	**92.83**	**0.070**	**7.29**	**0.013**	**0.964**	**27.10**	**0.053**

Table 1. **Comparison on single-object generation on Toy4K dataset.** We showcase the performance of our method in two scenarios: one where explicit point cloud priors are provided, and another where point cloud are inferred from condition images using VGGT [58].

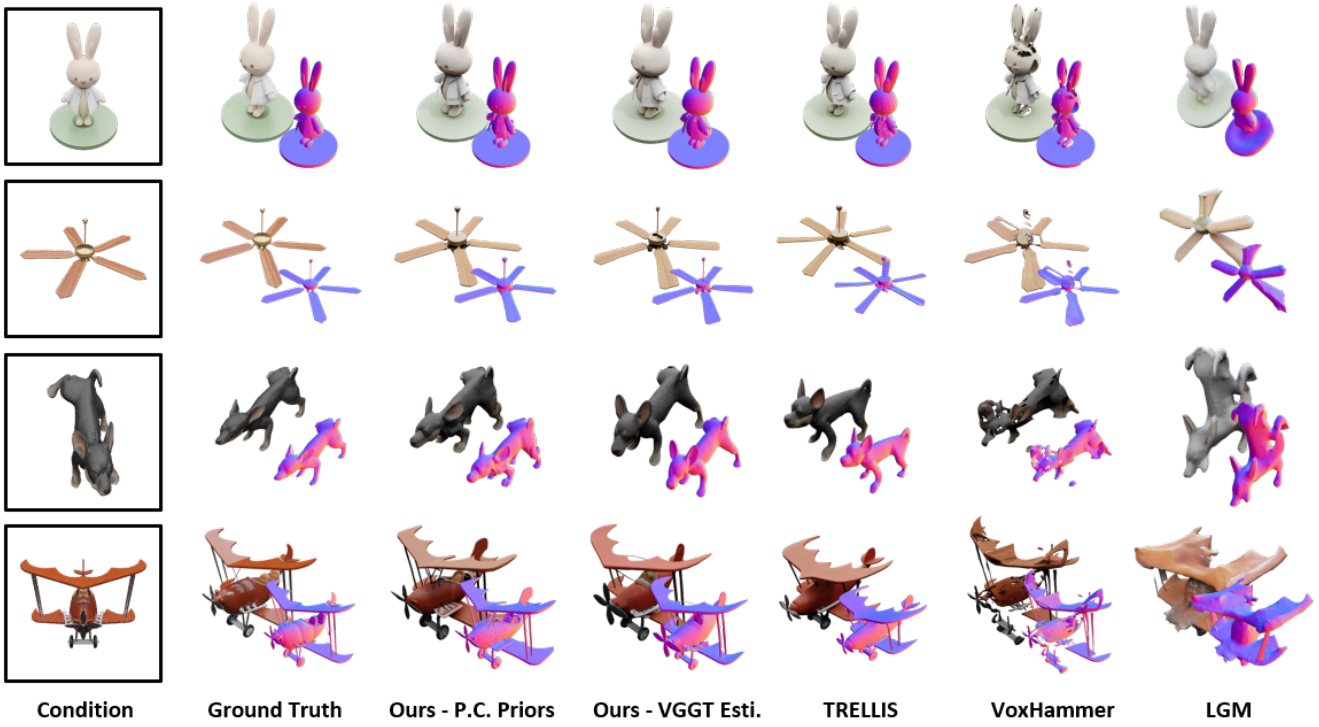

Figure 4. **Single-object generation on Toys4K.** For the explicit point cloud priors results, we use point cloud extracted strictly from the visible region of input images, whereas the "VGGT-estimated" results use point clouds inferred from the condition images by VGGT.

to reproduce their results. We reproduce the results of the 3D editing method VoxHammer [31] to represent the 3D inversion's results. Specifically, we use the structural voxels obtained from the same initial point clouds as in our method to define the "Unedited Region" in VoxHammer, and then apply their pipeline to obtain the final generated results.

4.2. Main Results

Single-object generation. We first present the results of single-object generation on Toys4K [51]. As shown in Tab. 1, our method consistently outperforms existing approaches across all evaluation metrics, whether using existing point cloud priors or VGGT [58]-predicted point cloud. Notably, in terms of geometric metrics, the results with point cloud priors achieve an F-score of 0.963, demonstrates our approach produces geometry closely approximates the ground-truth structure. The significant improvement in geometry enhances the visual fidelity of the results. As illustrated in Fig. 4, our results better match the overall appearance of the ground-truth compared to other methods, and normal maps further highlight the superior geometric quality achieved by our approach. Notably, while VoxHammer adopts the same 3D priors as ours, the image condition fails to provide cues for the missing parts of the 3D priors, making 3D inversion process struggles to complete the unknown regions. In contrast, our method leverages the trained model's inpainting capability to fully exploit the existing 3D priors and effectively infer the missing geometry.

Multi-object generation. We evaluate our method on the multi-object generation dataset 3D-FRONT [13]. As

Method	Rendering				Geometry			
	PSNR ↑	SSIM(%) ↑	LPIPS ↓	DINO(%) ↓	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
TRELLIS [62]	18.21	83.12	0.239	12.33	0.094	0.478	18.76	0.258
VoxHammer [31] (3D Inversion)	19.29	84.70	0.179	18.41	0.051	0.686	20.43	0.181
SceneGen [35]	18.32	83.35	0.231	14.43	0.086	0.485	19.08	0.229
MIDI [19]	19.23	85.59	0.166	14.25	0.075	0.513	20.82	0.164
Points-to-3D (Ours-VGGT Esti.)	20.52	86.51	0.152	8.90	0.040	0.743	20.97	0.160
Points-to-3D (Ours-P.C.Priors)	**21.63**	**87.73**	**0.124**	**8.29**	**0.025**	**0.886**	**22.38**	**0.124**

Table 2. **Comparison on multi-object generation on 3D-FRONT dataset.** Points-to-3D consistently outperforms state-of-the-art multi-object generation methods across all evaluation metrics.

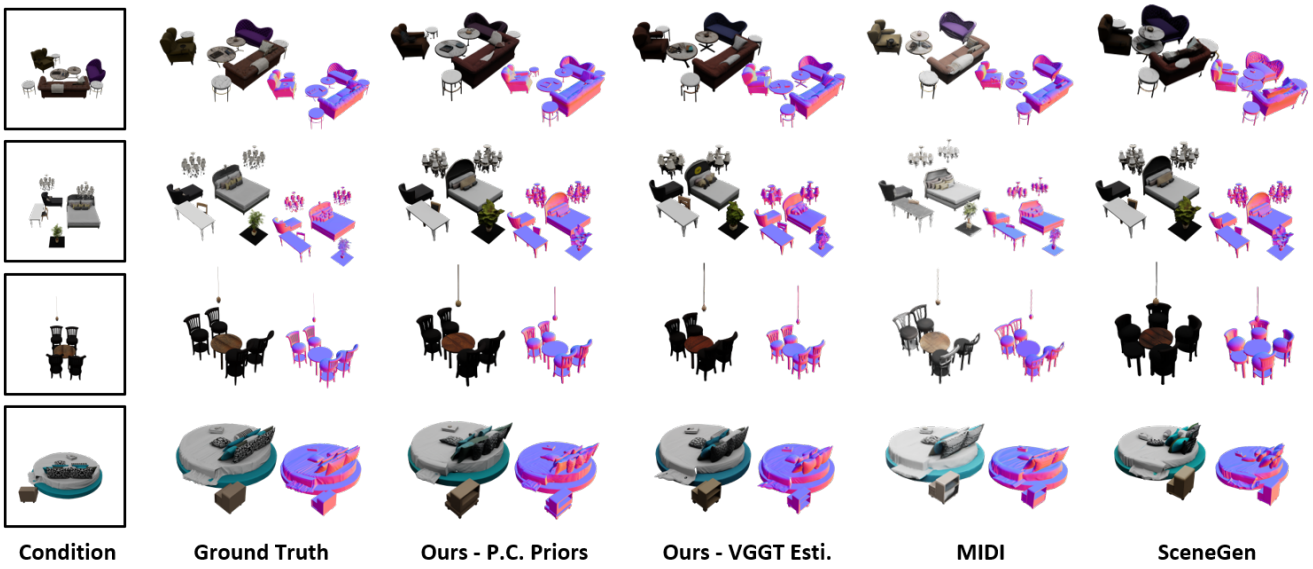

Figure 5. **Multi-object generation on 3D-FRONT.** The input point cloud priors setting is the same as in Fig. 4.

Methods	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
TRELLIS [62]-O.	0.034	0.832	23.81	0.105
TRELLIS [62]-V.	0.032	0.854	24.77	0.093
Points-to-3D-O.	**0.013**	**0.964**	**27.10**	**0.053**
Points-to-3D-V.	**0.007**	**0.998**	**29.00**	**0.036**

Table 3. **Comparison on visible and overall geometry results on Toys4K.** We present the comparison between our method and TRELLIS. For each method, the upper row (O.) shows the overall results, while the lower row (V.) shows the visible region results.

shown in Tab. 2, incorporating point cloud priors provides substantial guidance for reconstructing overall geometry in complex multi-object scenarios, which support our method achieves significant improvements across all evaluation metrics compared to other methods. The rendered images and normal maps in Fig. 5 further demonstrate that our results better align with the ground-truth scene geometry and visual appearance. Unlike MIDI [19] or SceneGen [35], which implicitly utilize spatial information, our framework explicitly incorporates geometric priors within the architecture, enabling more direct and effective control over 3D geometry, offering a promising solution for generating complex 3D scenes.

Visible region performance. We also highlight the generation results for the visible regions—i.e., the areas covered by our point cloud priors. As shown in Tab. 3, within these visible regions, our generated results achieve an F-Score of 0.998 and a chamfer distance (CD) of 0.007, indicating a strong alignment with the ground truth structure. This demonstrates our structure generation pipeline effectively preserves the information provided by the point cloud priors while producing high-quality overall geometry. Compared to our baseline method, both in the visible regions and across the entire structure, our approach achieves substantial improvements in geometric fidelity, fulfilling the primary objectives of our work.

4.3. Ablation Studies

VGGT point clouds estimation. When point cloud are not available as input, our method can also leverage the condition image to predict an initial point cloud using feed-forward methods like VGGT [58]. We evaluate the generation results based on VGGT-estimated point cloud, as shown in Tab. 1 and Tab. 2. Although the generation results

Inp.	Ref.	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
50	0	0.014	0.960	25.88	0.065
40	10	0.013	0.962	26.49	0.059
30	20	0.013	0.963	26.89	0.056
25	25	**0.013**	**0.963**	**27.10**	**0.053**
20	30	0.013	0.962	27.03	0.055
10	40	0.014	0.961	26.72	0.061

Table 4. **Ablation study.** We evaluate the number of inpainting steps (Inp.) and refinement steps (Ref.) in our sampling strategy.

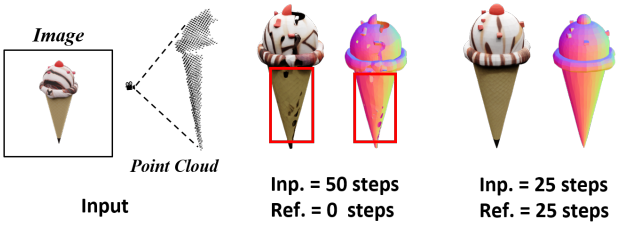

Figure 6. **Ablation study.** Allocating the full sampling to inpainting (Inp.) results in geometric "holes" along the inpainting edge.

with VGGT point cloud exhibit some gap compared to using accurate point cloud priors—this is largely due to the inherent prediction errors of VGGT. Nevertheless, compared to other existing approaches, it consistently achieves substantial improvements in both geometric accuracy and visual fidelity. These highlight the strong robustness and flexibility of our pipeline, with the absence of high-precision priors, our framework can still effectively utilize predicted point cloud form image-only inputs to achieve high-quality geometry generation.

Staged sampling strategy. We propose a staged sampling strategy in our pipeline, which leverages a limited number of last steps with noise to perform global optimization, effectively addressing the "holes" along inpainting boundaries that are otherwise difficult to avoid. We investigate the effect of refinement step allocation through an ablation study. In Tab.4, we present generation results under different allocations of inpainting and refinement steps with the same total sampling steps. When the entire sampling process is allocated to inpainting, the geometric reconstruction suffers from "holes" on inpainting edge, as further illustrated in Fig. 6. By setting the sampling schedule to 25 inpainting steps followed by 25 refinement steps, the geometric metrics reach their best performance, and the previously observed "holes" are effectively eliminated as in Fig. 6, yielding the overall best generation results.

4.4. Real-world Examples and Text-to-3D

We further evaluate the robustness of our method on real-world images from the Pix3D [52] dataset. As illustrated in Fig. 7, our approach maintains robust performance on real image inputs, producing geometry that aligns more

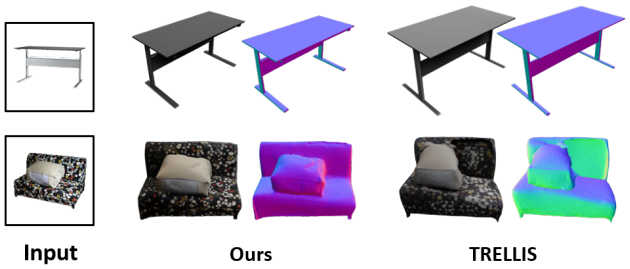

Figure 7. **Real-world examples on Pix3D.**

Figure 8. **Text-to-3D generation on Toys4K.**

Methods	CLIP ↑	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
LGM [54]	0.247	0.086	0.412	19.55	0.223
TRELLIS [62]	0.298	0.047	0.639	21.25	0.159
Points-to-3D	**0.299**	**0.022**	**0.892**	**24.75**	**0.094**

Table 5. **Comparison of text-to-3D generation on Toys4K.**

faithfully with the input images compared to the baseline method. Moreover, we also assess our model under text-to-3D settings on Toys4K [51], where text prompts and point cloud priors are provided as input. As shown in Tab. 5 and Fig. 8, our method successfully generates geometries that are semantically consistent with the input prompts and structurally well-controlled by the given point cloud priors.

5. Conclusion

We introduce Points-to-3D, a diffusion-based framework that first leverages explicit 3D point cloud priors as input to enable geometry-controllable 3D asset and scene generation. Built upon the latent 3D diffusion model TRELLIS [62], we investigate a natural way to embed point cloud as initialization within the framework. After training TRELLIS's structure generation network to acquire inpainting capabilities, we employ a staged sampling strategy—structural inpainting followed by boundary refinement—that reconstructs the global geometry while preserving the input visible regions. Experiments demonstrate the benefits of explicitly embedding 3D priors, highlighting a promising direction for controllable and reliable 3D generation in real-world applications.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, pages 5855–5864, 2021. 2

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 2

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022. 2

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, pages 14124–14133, 2021. 2

[5] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, pages 333–350, 2022. 2

[6] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6593–6602, 2024. 3

[7] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, pages 370–386, 2024. 2

[8] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. 2024. 2

[9] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 4456–4465, 2023. 3

[10] Ruihang Chu, Enze Xie, Shentong Mo, Zhenguo Li, Matthias Nießner, Chi-Wing Fu, and Jiaya Jia. Diffcomplete: Diffusion-based generative 3d shape completion. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[11] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *CVPR*, 2022. 5, 1

[12] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 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 *ICCV*, 2021. 2, 5, 6, 1

[14] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *IJCV*, 129:3313–3337, 2021. 5, 1

[15] Juan D Galvis, Xingxing Zuo, Simon Schaefer, and Stefan Leutengger. Sc-diff: 3d shape completion with latent diffusion models. *arXiv preprint arXiv:2403.12470*, 2024. 3

[16] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022. 2

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[18] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*, 2024. 2

[19] 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 Computer Vision and Pattern Recognition Conference*, pages 23646–23657, 2025. 2, 7

[20] Hanwen Jiang, Qixing Huang, and Georgios Pavlakos. Real3d: Scaling up large reconstruction models with real-world images. 2025. 2, 6

[21] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *CVPR*, pages 5322–5332, 2024. 2

[22] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, pages 150–168. Springer, 2024. 3

[23] Yoni Kasten, Ohad Rahamim, and Gal Chechik. Point cloud completion with pretrained text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 3

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

[25] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language embedded radiance fields. In *ICCV*, pages 19729–19739, 2023. 2

[26] 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 *CVPR*, 2024. 5, 1

[27] Andreas Kurz, Thomas Neff, Zhaoyang Lv, Michael Zollhöfer, and Markus Steinberger. AdaNeRF: Adaptive sampling for real-time rendering of neural radiance fields. In *ECCV*, pages 254–270. Springer, 2022. 2

[28] Yushi Lan, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy. Gaussiananything: Interactive point cloud latent diffusion for 3d generation. In *ICLR*, 2025. 2, 6

[29] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024. 2

[30] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *CVPR*, 2024. 2

[31] Lin Li, Zehuan Huang, Haoran Feng, Gengxiong Zhuang, Rui Chen, Chunchao Guo, and Lu Sheng. Voxhammer: Training-free precise and coherent 3d editing in native 3d space. *arXiv preprint arXiv:2508.19247*, 2025. 2, 6, 7

[32] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2

[33] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10072–10083, 2024. 2

[34] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 3

[35] Yanxu Meng, Haoning Wu, Ya Zhang, and Weidi Xie. Scenegen: Single-image 3d scene generation in one feedforward pass. *arXiv preprint arXiv:2508.15769*, 2025. 2, 7

[36] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[37] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H Mueller, Chakravarty R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. DONeRF: Towards real-time rendering of compact neural radiance fields using depth oracle networks. In *Comput. Graph. Forum*, pages 45–59, 2021. 2

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. 2024. 5

[39] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[40] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2

[41] 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. 2021. 5

[42] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, pages 12892–12901, 2022. 3

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[44] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 2

[45] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in neural information processing systems*, 33:20154–20166, 2020. 2

[46] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 2

[47] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[48] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2

[49] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 2

[50] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[51] Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. 2021. 2, 5, 6, 8, 1

[52] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*, 2018. 5, 8

[53] Stanislaw Szymanowicz, Jason Y. Zhang, Pratul Srinivasan, Ruiqi Gao, Arthur Brussee, Aleksander Holynski, Ricardo Martin-Brualla, Jonathan T. Barron, and Philipp Henzler. Bolt3d: Generating 3d scenes in seconds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 24846–24857, 2025. 2

[54] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024. 2, 6, 8

[55] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan.

Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5283–5293, 2025. 2

[56] SAM3D Team, Chen Xingyu, Chu Fu-Jen, Gleize Pierre, Kevin J. Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. Sam3d: 3dfy anything in images. https://ai.meta.com/research/publications/sam-3d-3dfy-anything-in-images/, 2025. 1, 2

[57] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 2

[58] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025. 2, 3, 6, 7, 1

[59] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 2

[60] Tianhao Wu, Chuanxia Zheng, Frank Guan, Andrea Vedaldi, and Tat-Jen Cham. Amodal3r: Amodal 3d reconstruction from occluded 2d images. *arXiv preprint arXiv:2503.13439*, 2025. 2

[61] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851, 2024. 3

[62] 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 Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 2, 3, 5, 6, 7, 8, 1

[63] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025. 2

[64] Yunhan Yang, Yufan Zhou, Yuan-Chen Guo, Zi-Xin Zou, Yukun Huang, Ying-Tian Liu, Hao Xu, Ding Liang, Yan-Pei Cao, and Xihui Liu. Omnipart: Part-aware 3d generation with semantic decoupling and structural cohesion. *arXiv preprint arXiv:2507.06165*, 2025. 2

[65] Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image. *ACM Transactions on Graphics (TOG)*, 44(4): 1–19, 2025. 2

[66] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 2

[67] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5525–5534, 2019. 2

[68] Xin Yu, Tianyu Wang, Soo Ye Kim, Paul Guerrero, Xi Chen, Qing Liu, Zhe Lin, and Xiaojuan Qi. Objectmover: Generative object movement with video prior. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17682–17691, 2025. 3

[69] Zehao Yu and Shenghua Gao. Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1949–1958, 2020. 2

[70] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *CVPR*, pages 19447–19456, 2024. 2

[71] Jingyang Zhang, Yao Yao, and Long Quan. Learning signed distance field for multi-view surface reconstruction. *International Conference on Computer Vision (ICCV)*, 2021. 2

[72] 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 *ICLR*, 2025. 2

[73] 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 (TOG)*, 43(4):1–20, 2024. 2

[74] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5, 1

[75] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 3

Points-to-3D: Structure-Aware 3D Generation with Point Cloud Priors

Supplementary Material

A. Experimental Details

Our training dataset consists of object collections from the 3D-FUTURE [14] (9,472 objects), HSSD [26] (6,670 objects), and ABO [11] (4,485 objects) datasets. For each object, we render the image of the $T = 24$ views, together with the corresponding depth map, and extract the visible point cloud for each view by enforcing depth consistency with a threshold $\tau = 0.05$ times the depth range (maximum minus minimum depth) in that view. The visible point cloud is then converted into an initial SS latent, which is paired with the original SS latent as ground truth to train the sparse structure flow transformer for inpainting.

For evaluation, we use randomly sampled subset of the Toys4K [51] (500 objects) dataset and 3D-FRONT [13] (500 scenes) dataset. For each test object or scene, we render 8 views using cameras with yaw angles $(0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°)$ and a fixed pitch angle of $30°$. The camera is positioned at a radius of 1.8 from the object center. For PSNR, SSIM, and LPIPS [74], we directly compare the rendered images of generated results with the rendered images of the ground-truth objects and report the average scores. For the DINO-based similarity metric, we report the average discrepancy between the rendered images of the generated and ground-truth assets, quantified as $(1 - S_{\text{DINO}})$, where S_{DINO} denotes the DINO similarity score. For the normal-based metric, we render normal maps from the 8 views and compute the average score between the normal maps of the generated and ground-truth assets. For Chamfer Distance (CD) and F-score, we normalize all the objects within the range (-0.5, 0.5) and set the F-score distance threshold to 0.05. During testing, for the point cloud priors input, we align the point cloud to the orientation of the corresponding ground-truth object to ensure that the generation conditioned on this point cloud can be directly evaluated.

B. More Results

We provide additional qualitative examples and experimental results to further demonstrate the performance of our method.

B.1. Multi-Views Input Generation

Because our flow-based model performs iterative denoising, it can directly incorporate multi-view reference images as conditioning inputs at different denoising steps. For VGGT-estimated point clouds, multi-view inputs produce more accurate predictions; moreover, across all point cloud priors, greater point cloud coverage consistently leads to better reconstruction. We further evaluate the case of using three input views on Toys4K [51] dataset. Specifically, we first feed the multi-view reference images into VGGT [58] to obtain a more complete predicted point cloud. As shown in Tab. 6, while multi-view input naturally improves the baseline TRELLIS [62] geometry, our method achieves substantially higher structural accuracy, consistently maintaining controllable geometry. For accurate point cloud priors, we extract the visible sampled surface point cloud from the three views using depth consistency and use it as the input prior. With these priors, our method produces reconstructions that are very close to the ground truth. Fig. 9 further shows the visualization comparisons. These results demonstrate the robustness and effectiveness of our method across different numbers of input images.

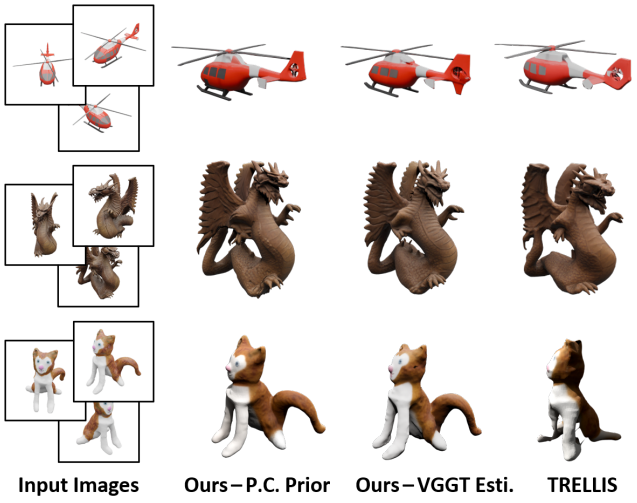

Figure 9. **Generation results with 3 input views on Toys4K.** The first column of our results uses sampled point-cloud priors extracted from the visible regions of the three input images, whereas the "VGGT-estimated" results rely on point clouds inferred from the input images by VGGT.

B.2. Comparison with SAM3D

We additionally compare our approach with the latest state-of-the-art method SAM3D [56], which also builds on TRELLIS [62]. Although SAM3D highlights the value of 3D priors and also leverages point maps, it integrates these priors indirectly through the attention mechanism of the flow transformer block, which—as also stated in their paper—does not support explicit geometric control. As shown in Tab. 6, with pointmap inputs as well, SAM3D exhibits limited ability to enforce precise geometric control compared to our approach. This is further illustrated in Tab. 7,

Method	Views Num.	Rendering PSNR ↑	SSIM(%) ↑	LPIPS ↓	DINO(%) ↓	Geometry CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
SAM3D [56]	1	22.42	91.45	0.111	8.01	0.033	0.835	23.85	0.101
Points-to-3D (Ours-VGGT Esti.)	1	22.55	92.09	0.088	7.37	0.024	0.881	24.53	0.085
Points-to-3D (Ours-P.C.Priors)	1	**22.91**	**92.83**	**0.070**	**7.29**	**0.013**	**0.964**	**27.10**	**0.053**
TRELLIS [62]	3	23.19	92.63	0.075	5.79	0.025	0.904	26.22	0.066
Points-to-3D (Ours-VGGT Esti.)	3	23.44	93.21	0.057	5.58	0.015	0.971	28.35	0.035
Points-to-3D (Ours-P.C.Priors)	3	**23.98**	**94.02**	**0.050**	**5.26**	**0.009**	**0.988**	**30.45**	**0.028**

Table 6. **Comparison on single-object generation with different views input on Toy4K dataset.** We indicate the number of input views on the left side of the table, and the table's upper section shows the single-view results, while the lower section shows three-view results.

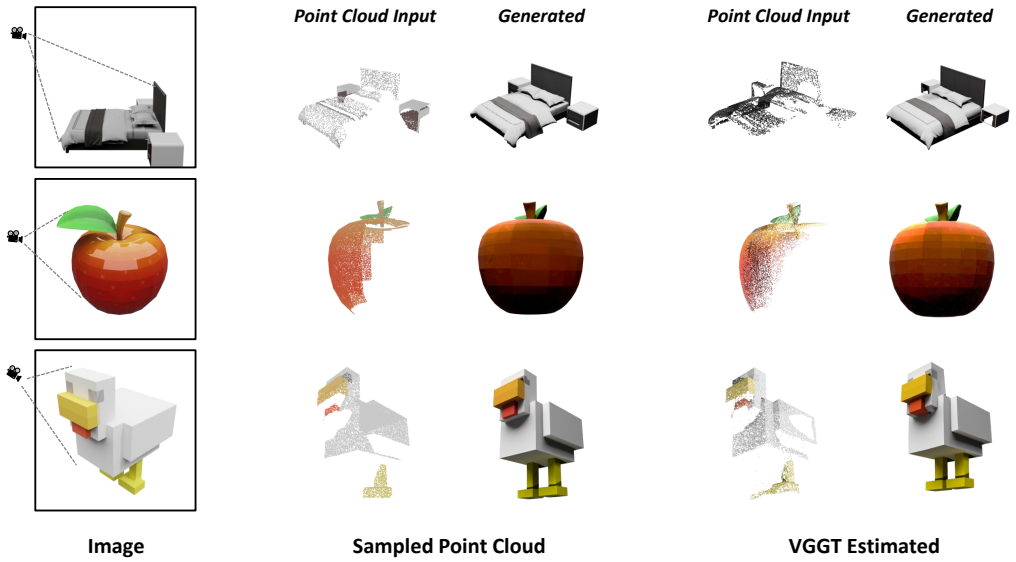

Figure 10. **Input point cloud priors examples.** We show the observable point cloud priors examples for the two input modes with single-view input in this paper, along with their corresponding generation results.

Methods	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
SAM3D [56]-O.	0.033	0.835	23.85	0.101
SAM3D [56]-V.	0.031	0.841	24.81	0.090
Points-to-3D-O.	**0.013**	**0.964**	**27.10**	**0.053**
Points-to-3D-V.	**0.007**	**0.998**	**29.00**	**0.036**

Table 7. **Comparison on visible and overall geometry results of single-view input on Toys4K.** We present the comparison between our method and SAM3D [56]. For each method, the upper row (O.) shows the overall results, while the lower row (V.) shows the visible region results.

where SAM3D fails to achieve improved geometry even within the regions covered by the pointmap (i.e., the visible areas in the table). In contrast, our method injects 3D priors through a more direct and explicit mechanism, enabling effective and reliable geometric controllability, providing current 3D generation frameworks a stronger opportunity to benefit from sensed 3D priors as well as future improvements in feed-forward point-map prediction methods.

B.3. Point Cloud Priors Examples

In Fig. 10, we illustrate examples of the two types of point cloud priors considered in this work, which correspond to the two most common practical scenarios: (1) partial point clouds directly captured by hardware sensors (e.g., LiDAR on an iPhone), and (2) point cloud estimated from input images via feed-forward point-map prediction (e.g., VGGT [58]). This experimental setup enables a comprehensive evaluation of our method over a broader spectrum of practical cases. As shown in Fig. 10, these visible-region priors impose reliable geometric constraints that steer our model toward controllable and faithful 3D generation. The accompanying quantitative results in Tab. 7 further verify that our explicit prior-injection scheme effectively preserves the geometry in the input 3D priors. This formulation enables current 3D generation frameworks to integrate with

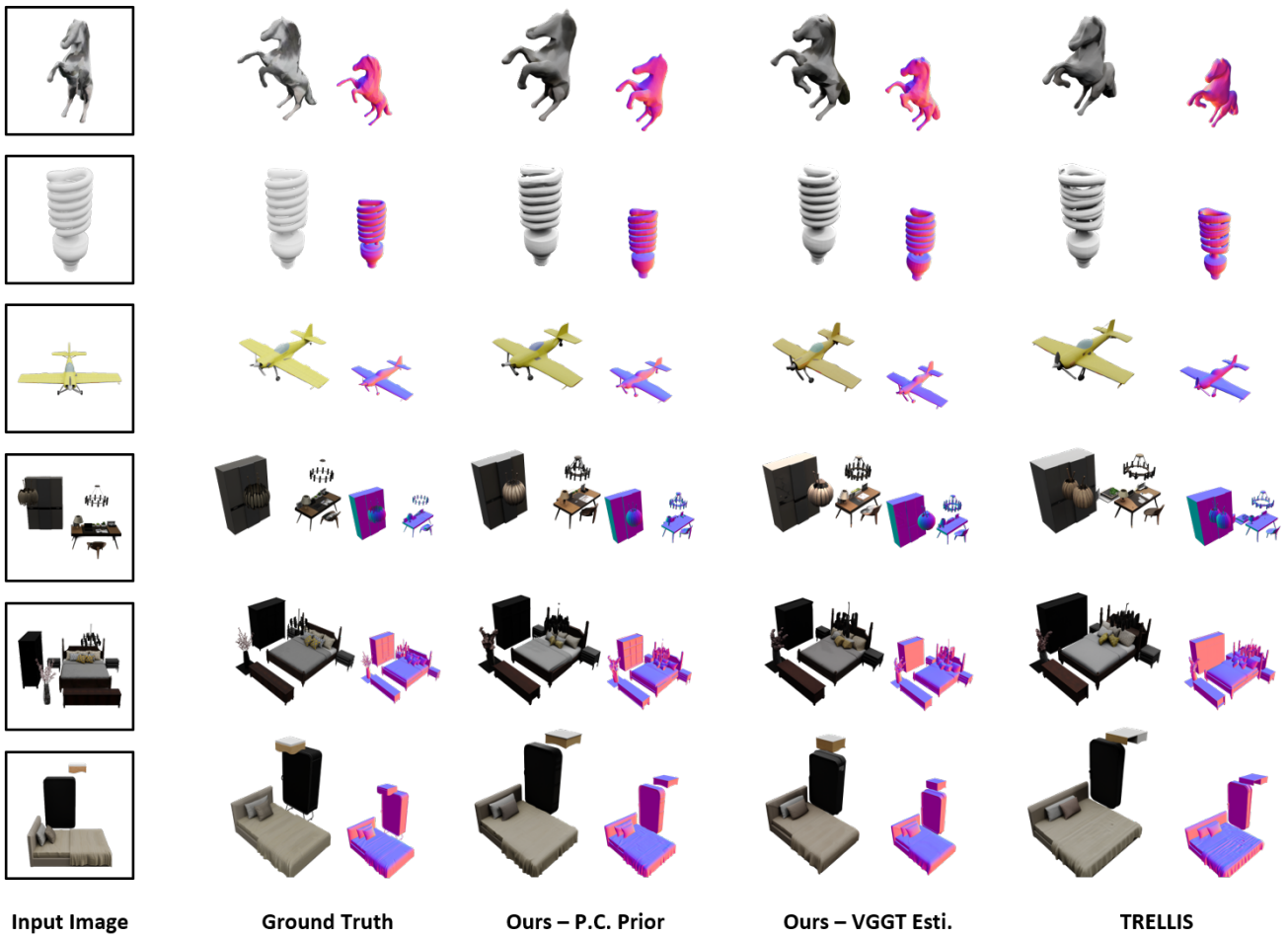

Figure 11. **More image-to-3D examples.** More single-image to 3D generation visualization results on Toy4K (row 1-3) and 3D-Front dataset (row 4-6).

Figure 12. **More real-world image generation examples.**

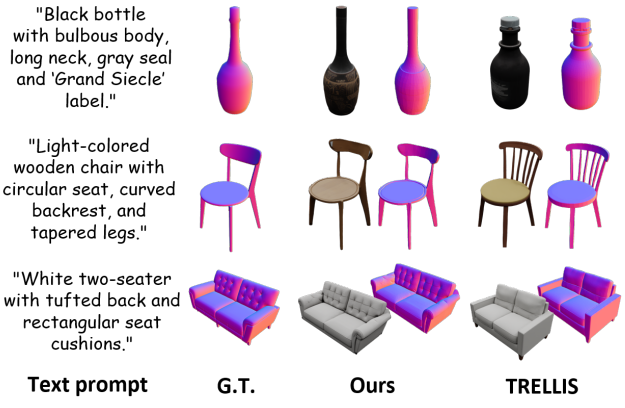

Figure 13. **More text-to-3D generation examples.**

broader systems.

B.4. More Image-to-3D Examples

We provide additional visualization results for image-to-3D generation in Fig. 11, demonstrating the effectiveness of our method. Experiments highlight that our method addresses a major limitation of existing 3D generation frameworks that struggle to fully incorporate available 3D information, and achieves substantial improvements in both single-object and multi-object generation.

B.5. More Real-world and Text-to-3D Examples

We showcase more results in real-world image generation in Fig. 12, demonstrating the robustness of our method in practical scenarios. And we also provide more text-to-3D examples in Fig. 13, illustrating that our method achieves more explicit geometric control when conditioned on text and partial point cloud priors, further validating the practical effectiveness of our approach.