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 [3] (9,472 objects), HSSD [4] (6,670 objects), and ABO [1] (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 τ = 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 [5] (500 objects) dataset and 3D-FRONT [2] (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 [9], 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 [5] dataset. Specifically, we first feed the multi-view reference images into VGGT [7] to obtain a more complete predicted point cloud. As shown in Tab. 1, while multi-view input naturally improves the baseline TRELLIS [8] 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. 1 further shows the visualization comparisons. These results demonstrate the robustness and effectiveness of our method across different numbers of input images.

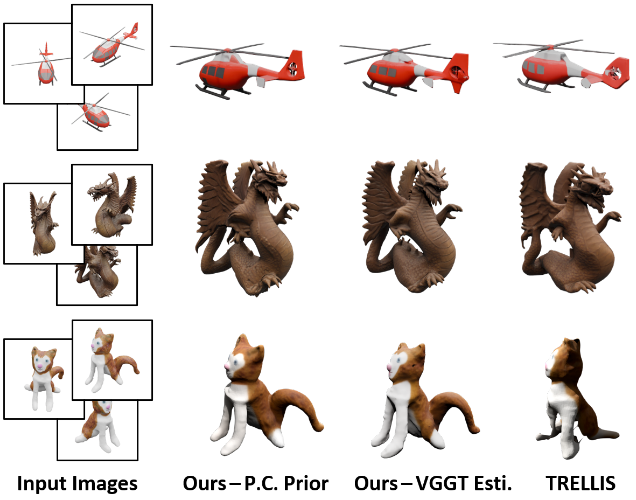

Figure 1. **Generation results with 3 input views on Toys4K.** The first column of our results uses sampled point-cloud priors extracted from the visible regions of the three input images, whereas the "VGGT-estimated" results rely on point clouds inferred from the input images by VGGT.

B.2. Comparison with SAM3D

We additionally compare our approach with the latest state-of-the-art method SAM3D [6], which also builds on TRELLIS [8]. 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. 1, with pointmap inputs as well, SAM3D exhibits limited ability to enforce precise geometric control compared to our approach. This is further illustrated in Tab. 2, where SAM3D

	Views	Rendering				Geometry			
Method	Num.	PSNR ↑	SSIM(%) ↑	LPIPS ↓	DINO(%) ↓	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
SAM3D [6]	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 [8]	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 1. **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.

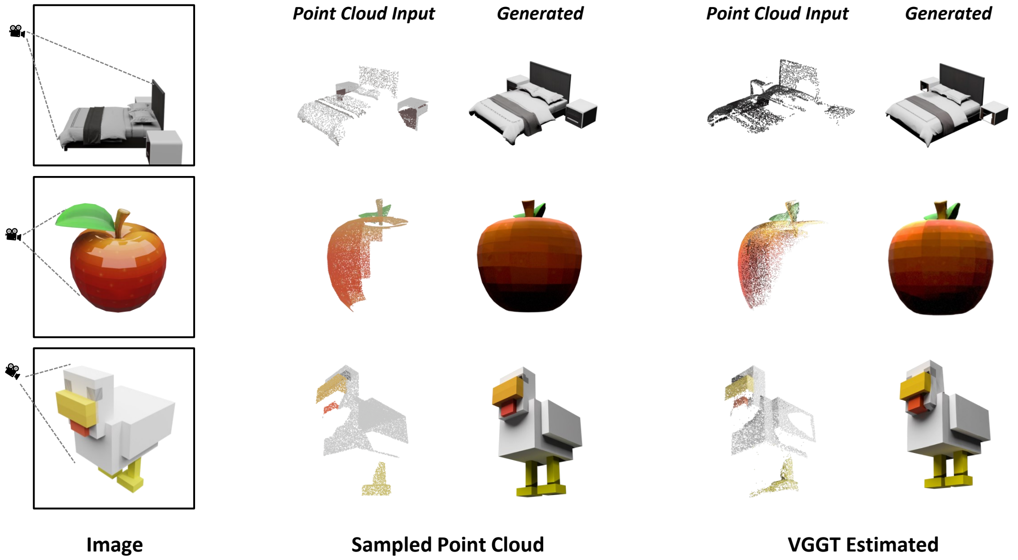

Figure 2. **Input point cloud priors examples.** We show the observable point cloud priors examples for the two input modes with single-view input in this paper, along with their corresponding generation results.

Methods	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
SAM3D [6]-O.	0.033	0.835	23.85	0.101
SAM3D [6]-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 2. **Comparison on visible and overall geometry results of single-view input on Toys4K.** We present the comparison between our method and SAM3D [6]. For each method, the upper row (O.) shows the overall results, while the lower row (V.) shows the visible region results.

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. 2, 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 [7]). This experimental setup enables a comprehensive evaluation of our method over a broader spectrum of practical cases. As shown in Fig. 2, these visible-region priors impose reliable geometric constraints that steer our model toward controllable and faithful 3D generation. The accompanying quantitative results in Tab. 2 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 broader systems.

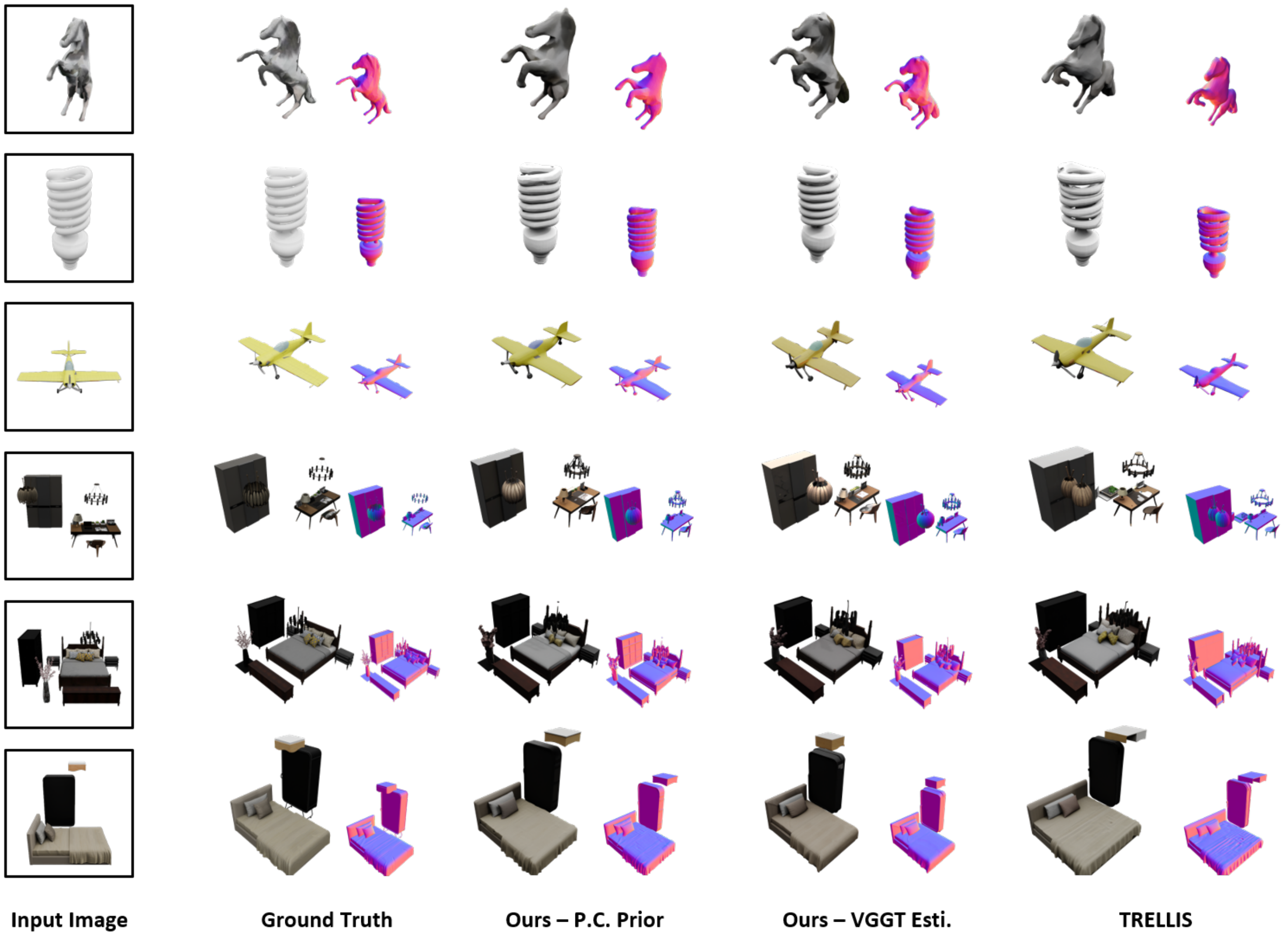

Figure 3. **More image-to-3D examples.** More single-image to 3D generation visualization results on Toy4K (row 1-3) and 3D-Front dataset (row 4-6).

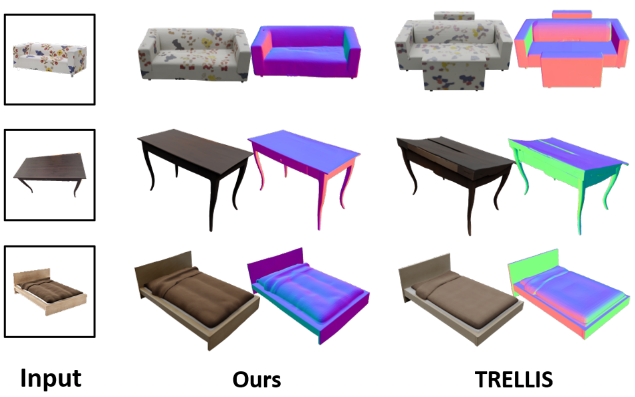

Figure 4. **More real-world image generation examples.**

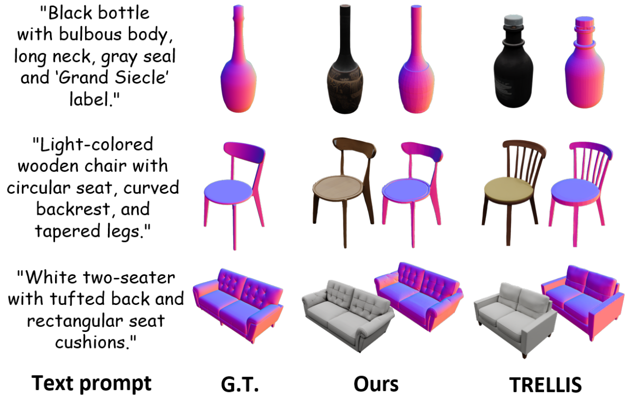
"Black bottle with bulbous body, long neck, gray seal and 'Grand Siecle' label."
"Light-colored wooden chair with circular seat, curved backrest, and tapered legs."
"White two-seater with tufted back and rectangular seat cushions."
Text prompt
G.T.
Ours
TRELLIS

Figure 5. **More text-to-3D generation examples.**

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

We provide additional visualization results for image-to-3D generation in Fig. 3, 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. 4, demonstrating the robustness of our method in practical scenarios. And we also provide more text-to-3D examples in Fig. 5, 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.

References

[1] 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. 1

[2] 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. 1

[3] 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. 1

[4] 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. 1

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

[6] 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

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

[8] 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. 1, 2

[9] 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. 1