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 [4] (9,472 objects), HSSD [5] (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 $\tau = 0.05$ 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 [6] (500 objects) dataset and 3D-FRONT [3] (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 [10], 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; and greater point cloud coverage consistently leads to better reconstruction. We further evaluate the case of using three input views on Toys4K [6] dataset. Specifically, we first feed the multi-view reference images into VGGT [8] to obtain a more complete predicted point cloud. As shown in Tab. 1, while multi-view input naturally improves the baseline TRELLIS [9] 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.

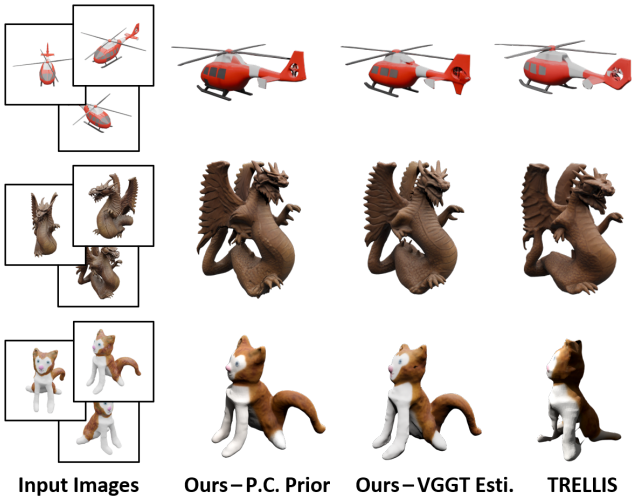

Figure 1. **Generation results with 3 input views on Toys4K.** The first column of our results uses sampled point-cloud priors extracted from the visible regions of the three input images, whereas the "VGGT-estimated" results rely on point clouds inferred from the input images by VGGT.

B.2. 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 [8]). 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

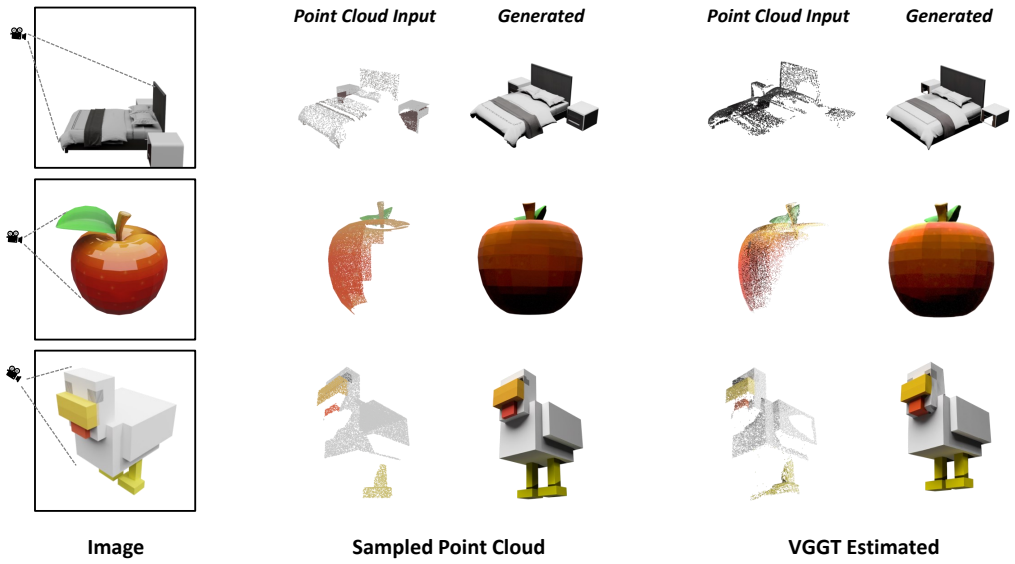

Figure 2. **Input point cloud priors examples.** We show the observable point cloud priors examples for the two input modes with single-view input in this paper, along with their corresponding generation results.

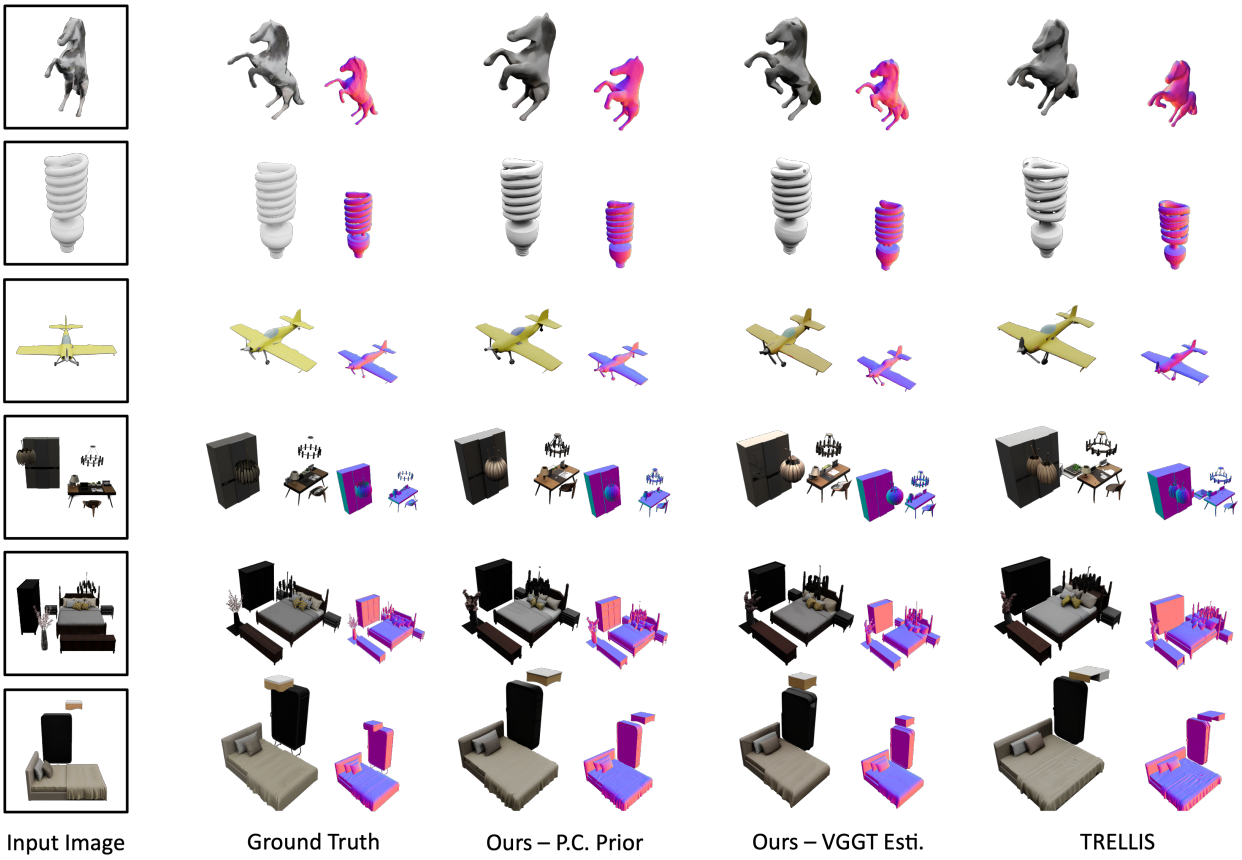

Figure 3. **More image-to-3D examples.** More single-image to 3D generation visualization results on Toy4K (row 1-3) and 3D-Front dataset (row 4-6).

Table 1. **Comparison on single-object generation with 3 views input on Toy4K dataset.**

Method	Rendering				Geometry			
	PSNR ↑	SSIM(%) ↑	LPIPS ↓	DINO(%) ↓	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
TRELLIS [9]	23.19	92.63	0.075	5.79	0.025	0.904	26.22	0.066
Points-to-3D (Ours-VGGT Esti.)	23.44	93.21	0.057	5.58	0.015	0.971	28.35	0.035
Points-to-3D (Ours-P.C.Priors)	**23.98**	**94.02**	**0.050**	**5.26**	**0.009**	**0.988**	**30.45**	**0.028**

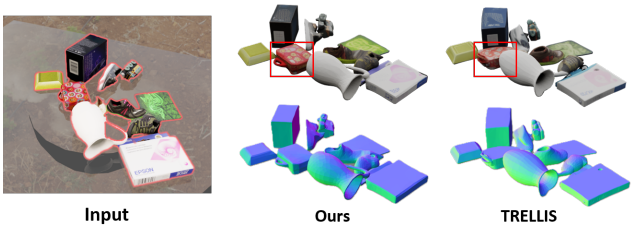

Figure 4. Example of cluttered table with depth sensor.

Figure 5. **More real-world image generation examples.**

Table 2. **Comparison of text-to-3D generation on Toys4K.**

Methods	CLIP ↑	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
LGM [7]	0.247	0.086	0.412	19.55	0.223
TRELLIS [9]	0.298	0.047	0.639	21.25	0.159
Points-to-3D	**0.299**	**0.022**	**0.892**	**24.75**	**0.094**

priors impose reliable geometric constraints that steer our model toward controllable and faithful 3D generation.

B.3. 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 scene-level generation. In Fig. 4 we also tested our model

Figure 6. **Text-to-3D generation examples.**

on cluttered table in RaySt3R [2] dataset with depth sensors and found it shows capability in handling such complex cases. Compared to TRELLIS, our method better preserves 3D geometry under this extreme out-of-distribution condition, highlighting the advantage and robustness of our method.

B.4. More Real-world and Text-to-3D Examples

We showcase more results in real-world image generation in Fig. 5, demonstrating the robustness of our method in practical scenarios. Moreover, we also assess our model under text-to-3D settings on Toys4K [6], where text prompts and point cloud priors are provided as input. As shown in Tab. 2 and Fig. 6, our method successfully generates geometries that are semantically consistent with the input prompts and structurally well-controlled by the given point cloud priors.

B.5. Noisy Point Clouds Input

Our method is primarily designed for settings where reliable 3D priors are available, where the goal is not to enhance existing point-cloud quality, but to faithfully pre-

Table 3. **Noisy point cloud priors.** We add different levels of perturbation to the accurate point-cloud priors to evaluate the impact of noisy 3D priors, and present the results of our simple repair process for noisy point cloud inputs below.

Methods	CD ↓	F-Score ↑	PSNR-N ↑	LPIPS-N ↓
P.C. priors	0.013	0.964	27.10	0.053
+ 5% perturbation	0.022	0.910	24.32	0.082
+ 10% perturbation	0.031	0.817	22.87	0.109
+ 15% perturbation	0.045	0.667	20.36	0.181
+ 10% perturbation & repair	0.027	0.855	23.47	0.098
+ 15% perturbation & repair	0.036	0.791	21.85	0.132

serve and explicitly integrate existing high-quality 3D priors—such as those obtained from LiDAR sensors (now reliable and widely available on smartphones)—into 3D generation frameworks, thereby enabling current 3D generation models to better leverage sensed 3D data and future advances in feed-forward point-map prediction methods. Nevertheless, we observe that low-quality point cloud inputs remain a key limitation of our current implementation, as they can negatively impact the overall performance.

We analyze the effect of noisy point clouds by adding random noise within a fixed error distance range, as shown in Tab. 3: our model remains robust to mild perturbations but degrades under heavy noise, which is expected since it explicitly relies on the visible prior. Although improving the quality of the priors is not the primary focus of this work, we propose a simple strategy that diffuses the initial noisy structure through a few sampling steps prior to the inpainting process. This design can be seamlessly integrated into our pipeline with no additional cost, leading to improved performance under noisy inputs, as shown in Tab. 3. However, in our current implementation, we do not recommend fully relying on unreliable 3D priors in the presence of noise. Addressing such cases requires further investigation in future work.

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] Bardienus P Duisterhof, Jan Oberst, Bowen Wen, Stan Birchfield, Deva Ramanan, and Jeffrey Ichnowski. Rayst3r: Predicting novel depth maps for zero-shot object completion. *arXiv preprint arXiv:2506.05285*, 2025. 3

[3] 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

[4] 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

[5] 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

[6] Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. 2021. 1, 3

[7] 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. 3

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

[9] 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, 3

[10] 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