DirectTriGS: Triplane-based Gaussian Splatting Field Representation for 3D Generation

Supplementary Material

7. Dataset Information.

Objaverse [1] is the main dataset for our experiment, which contains over 800K 3D objects. As the rendering process on such a massive dataset is very time-consuming, we adopt the pre-processed version sourced from the repository of [8], which pre-filters over 260K samples. In this processed dataset, every object is normalized to the voxel range of $[\pm 0.5, \pm 0.5, \pm 0.5]$, and rendered to RGBA images in a resolution of $512 * 512 * 4$, with 40 views in total. Our training data only comprises multi-view images and their corresponding camera poses, without any kind of original 3D data.

8. Implementation Details.

Triplane. The triplane resolution is configured as $3 \times 128 \times 128 \times 16$, where 16 represents the channels within each grid. The first half of the channels is designated for encoding geometry information, while the remaining half is allocated for encoding GS appearance details. Each triplane is initialized to random Gaussian noise with a standard deviation of 0.01. This random initialization allows the triplane to be decoded into random SDF values, subsequently leading to the generation of diverse fragmented mesh faces. Upon rasterization of these faces onto the screen, the geometry loss facilitates swift removal of undesired faces. During our experiments, we observed that this initialization method enables faster convergence compared to zero initialization.

As for loss configuration, we configure $w_1 = 5.0$, $w_2 = 1.0$, $w_3 = 1.2$, $\beta = 0.2$, $\gamma_1 = 0.2$, $\gamma_2 = 0.1$, $\gamma_3 = 0.01$, $\gamma_4 = 1.0$ by experiments, corresponding to the loss function described in Eq. 3, Eq. 4 and Eq. 5.

TriRenderer. As for the TriRenderer instroduced in Fig.1, both the geometry decoder and the GS attribute decoder inside it are composed of linear blocks. In the GS attribute decoder, there are 3 headers for GS splats scaling, opacity and SH prediction, and the rotation is fixed by the mesh face normal as introduced in Section 4.2. All the GS attribute headers are linear layers. We set SH degree to 1 in all experiments, which is enough to obtain satisfying results on Objaverse.

9. Simple Check of Triplane and VAE.

To better investigate whether it is reasonable to encode triplane using convolution-based methods, we simply scale the channel value of trained triplanes to pixel range and visualize them as shown in Fig. 8, where clear shapes from 3 different views can be observed. As for the VAE reconstruction, a slight blur in the reconstructed pictures are observed as Fig.9, which is inevitable but acceptable.

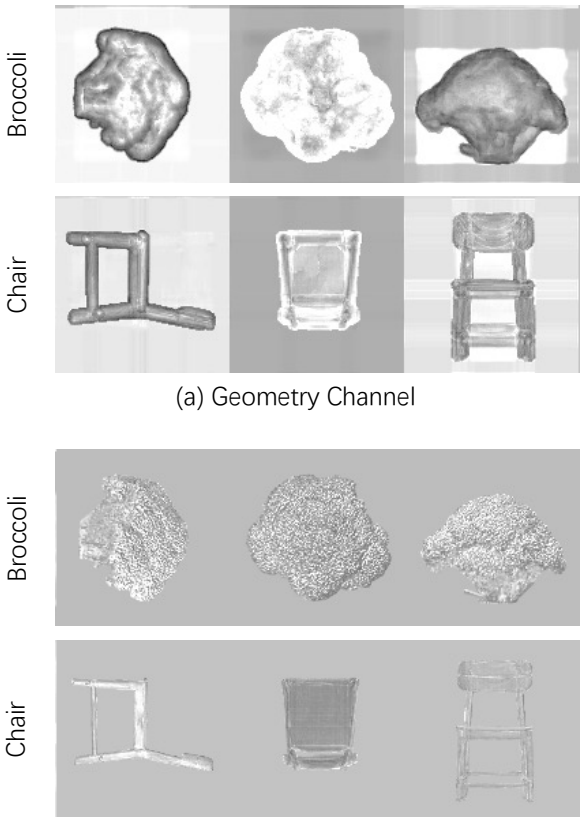
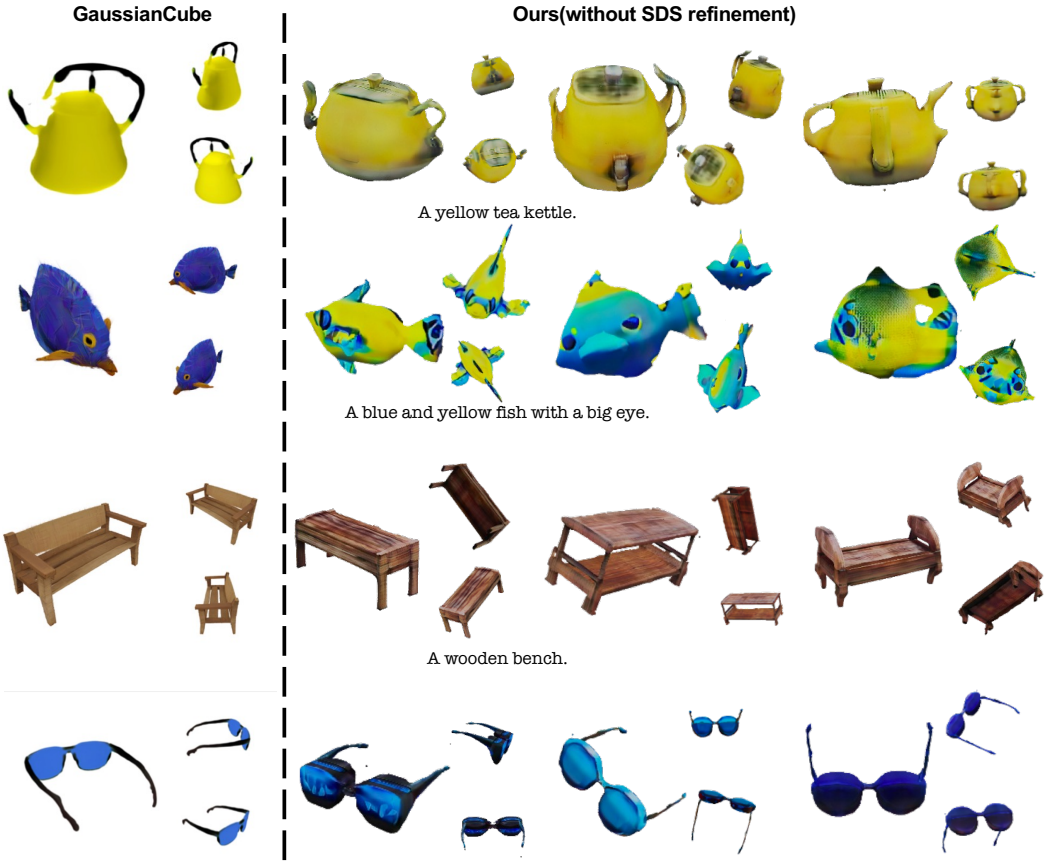


Figure 8. Channel visualization of sample triplanes.

10. Comparison with GaussianCube.

GaussianCube [9] is the most recent paper aiming to solve a similar task of ours, which can generate 3D GS directly without SDS or reconstruction from images. As for now, the authors of GaussianCube have not release their pre-trained models and runnable code for text-to-3D task on Objaverse dataset. Therefore, we just use the images provided in their paper for a qualitative comparison. The generated samples are shown in Fig. 10. Our method produces more diverse and detailing generation results.

Figure 9. Triplane reconstructed by VAE. Left: ground truth. Right: reconstruction.



(a) Generation with same prompts.

(b) Generation with slightly different prompt, for we do not include "Sonic the Hedgehog" in our training data.

Figure 10. Comparison with GaussianCube.

11. More Generation Results and Comparisons with LN3Diff and 3DTopia (without SDS Refinement).

More generated samples compared with LN3Diff [4], 3DTopia[2] and BrightDreamer[3] are rendered as Fig. 11∼ Fig. 14, where LN3Diff and 3DTopia are both direct generation methods exploiting Triplane as intermediate representation. The main differences between our work and these two alternatives are discussed as follows.

First, while we select 3D GS as our 3D representation, LN3Diff and 3DTopia use NeRF as their target 3D representations, which unavoidably involves some disadvantages of neural field such as artifacts, implicitness for further editing, and slower rendering speed. As shown in these figures, our work enables a generation with explicit 3D GS, and fast rendering with a minimum of artifacts, especially compared with 3DTopia.

Second, we use explicit mesh surface for GS point binding, which enables more accurate texture projection. Moreover, different geometry losses help to constrain the mesh to be clear and smooth. Therefore, our method produces more detailing appearances without blurs compared with LN3Diff and 3DTopia, which can be further examined by zooming in the figures.

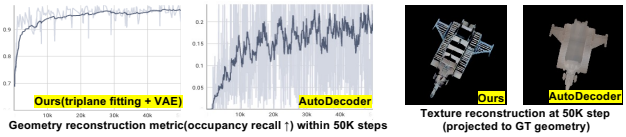
Moreover, there are several technical points are worth a discussion. For example, different from our end-to-end triplane VAE, LN3Diff adopts a image-to-triplane encoding. It is possible that the decoders cannot capture enough features from sparse input images for retrieving a high quality 3D object. Also, in such framework, different images sets of an unique object may corresponds to different latent codes, which is also a potential risk for following learning on latent space. As for 3DTopia, we guess that the absence of supervision for appearance detailing may the key reason for its blur output. In our method, the perceptual losses are incorporated in both the triplane encoding procedure and the VAE training.

12. Comparison with BrightDreamer.

BrightDreamer [3] is another methods that claim to generation 3D GS directly. However, it is trained on single class datasets created by Instant3D [5], we cannot compare it with ours using various prompts as before. Therefore, we just pick several cases for demonstration. As shown in Fig.15, while BrightDreamer also enables fast and direct GS generation, its output are more over-saturated, blurrier, and with more artifacts.

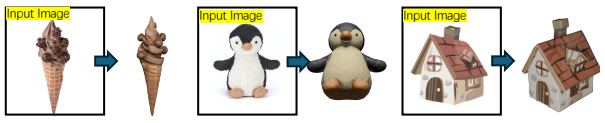
13. Additional Ablation Studies and Discussions.

Attempts to Simplify the Pipeline. 1) Unification of the geometry and appearance branches. We found that the losses of 2 branches interfere with each other, which leads to worse reconstruction in both triplane fitting and VAE training process. 2) We attempted to use an autodecoder as [7] to simplify the pipeline, which omits the triplane fitting and directly learn the latents instead. The experiments demonstrate a very slow convergence compared with our current design.



Some visual results of the mesh. Due to the direct regression of 3D shape using only 2D images, the result of stage 1) in Fig.1 a) is slightly worse than the methods using 3D supervision. Therefore, some unsmoothness in the generated 3D shapes can still be observed.

Preliminary Experiments on Image-Conditioned Generation. Although image-to-3D is a very popular route, since it relies heavily on image generation, and our initial target was to validate the effectiveness of triplane representation and TriRenderer, we chose text-to-3D as our primary task. We also conducted some preliminary training on image-to-3D, following the approach of LN3Diff by using CLIP features of images as conditions for generation. In practice, we found that since images provide richer guidance than text, image-to-3D is actually an easier task compared to text-to-3D.



Classifier Guidance. We found that the classifier guidance is an important factor for geometry completeness in LDM, and we use CFG between [5.0, 15.0] to avoid fragmental shapes.

Discussion on Limitations.1) While our framework does not directly bootstrap on pre-trained 2D models, it may leverages their capabilities through image conditioning and SDS postprocessing. 2) As for the generation quality, the

training data is the main bottleneck. There are many low-poly even strange objects in Objaverse and it is hard to screen them. Minor artifacts may appear in output, but our explicit 3D representation allows for straightforward post-processing (e.g., erosion/dilation) to address this issue. 3) As for SDS, we use it as an optional postprocess. We will explore alternative variants in future work to address its own limitations, e.g. ISM proposed in LucidDreamer [6].

References

[1] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 1

[2] Fangzhou Hong, Jiaxiang Tang, Ziang Cao, Min Shi, Tong Wu, Zhaoxi Chen, Shuai Yang, Tengfei Wang, Liang Pan, Dahua Lin, et al. 3dtopia: Large text-to-3d generation model with hybrid diffusion priors. *arXiv preprint arXiv:2403.02234*, 2024. 3

[3] Lutao Jiang and Lin Wang. Brightdreamer: Generic 3d gaussian generative framework for fast text-to-3d synthesis. *arXiv preprint arXiv:2403.11273*, 2024. 3

[4] Yushi Lan, Fangzhou Hong, Shuai Yang, Shangchen Zhou, Xuyi Meng, Bo Dai, Xingang Pan, and Chen Change Loy. Ln3diff: Scalable latent neural fields diffusion for speedy 3d generation. In *European Conference on Computer Vision*, pages 112–130. Springer, 2025. 3

[5] Ming Li, Pan Zhou, Jia-Wei Liu, Jussi Keppo, Min Lin, Shuicheng Yan, and Xiangyu Xu. Instant3d: Instant text-to-3d generation. *International Journal of Computer Vision*, pages 1–17, 2024. 3

[6] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6517–6526, 2024. 4

[7] Evangelos Ntavelis, Aliaksandr Siarohin, Kyle Olszewski, Chaoyang Wang, Luc V Gool, and Sergey Tulyakov. Autodecoding latent 3d diffusion models. *Advances in Neural Information Processing Systems*, 36:67021–67047, 2023. 3

[8] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. *arXiv preprint arXiv:2311.16918*, 2023. 1

[9] Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: Structuring gaussian splatting using optimal transport for 3d generative modeling. *arXiv preprint arXiv:2403.19655*, 2024. 1

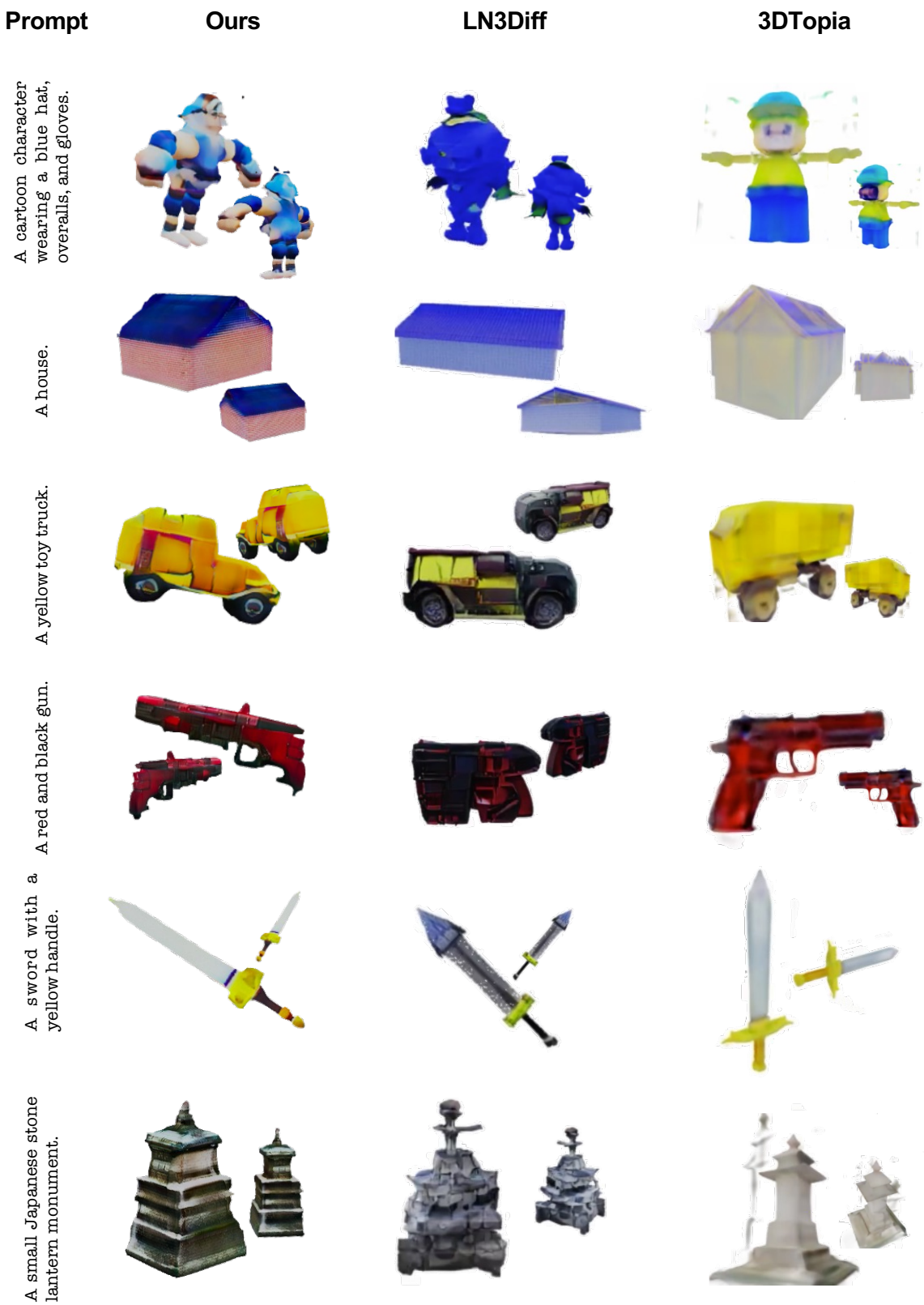


Figure 11. More generated samples (without SDS Refinement).

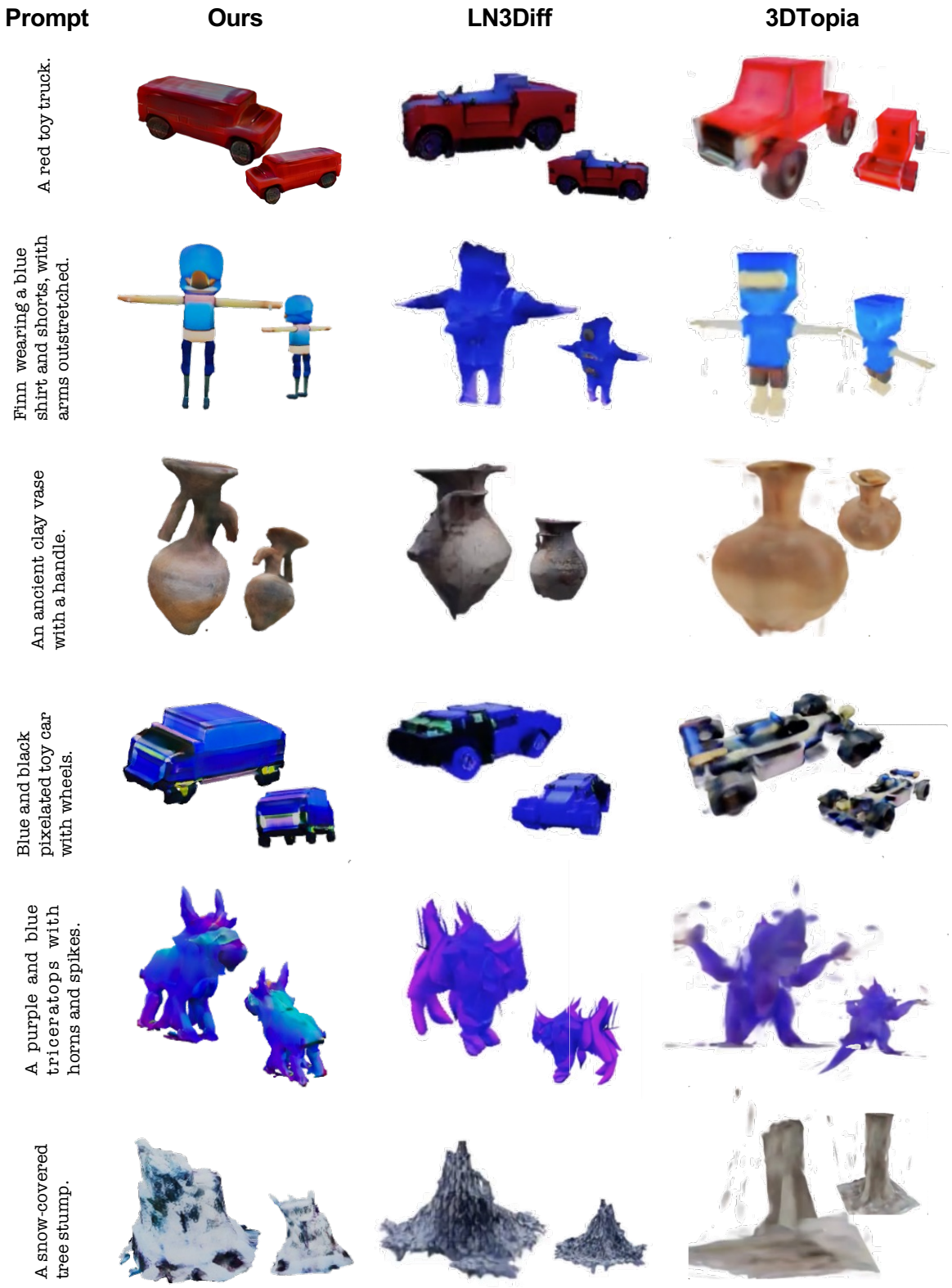


Figure 12. More generated samples (without SDS Refinement).

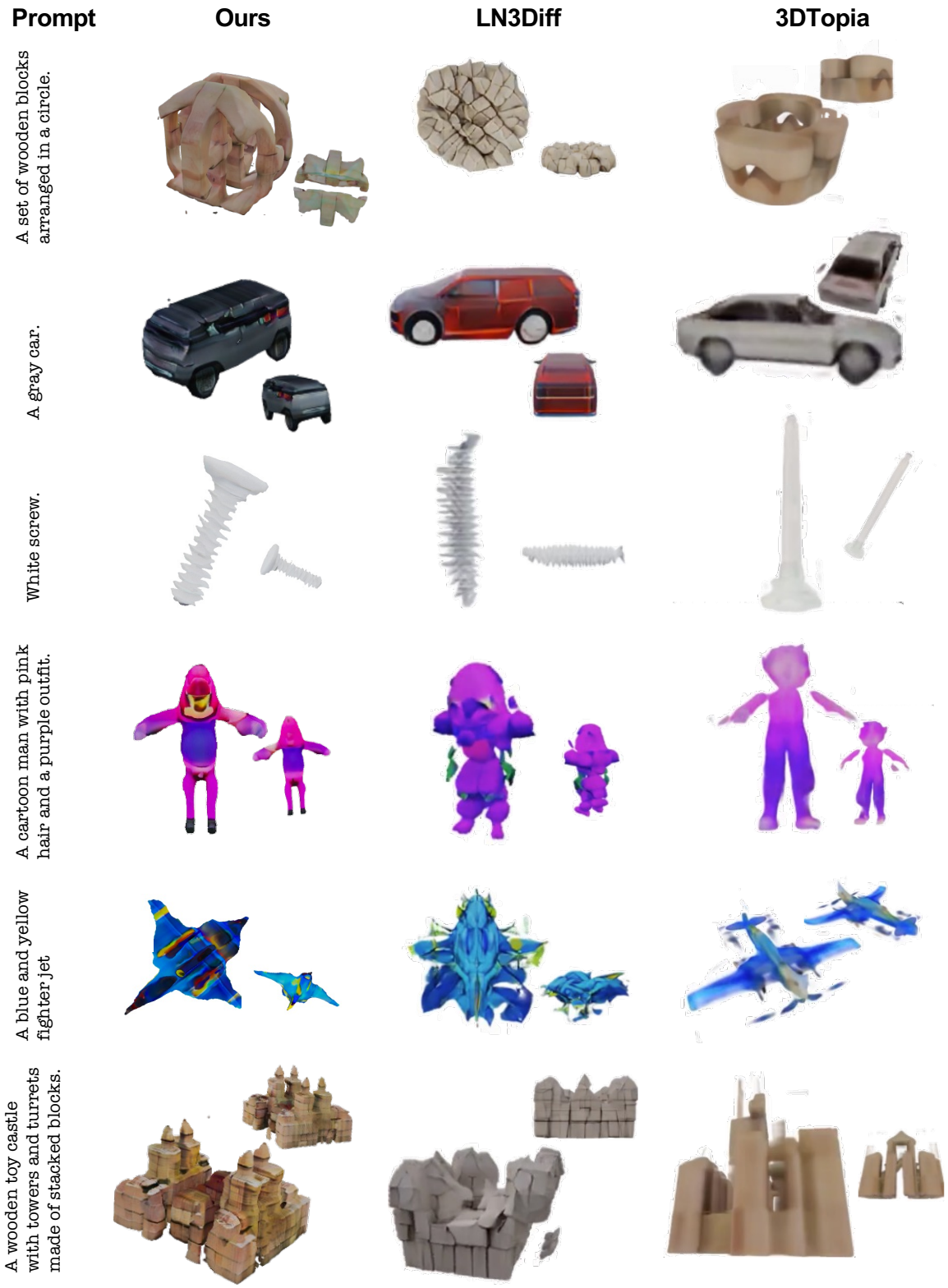


Figure 13. More generated samples compared with other methods (without SDS Refinement).

Figure 14. More generated samples compared with other methods(without SDS Refinement).

Figure 15. Case comparison with BrightDreamer.