Appendix

A Implementation Details

In the training process of the NeRF [Mildenhall *et al.*, 2021] setting, we set rendering resolution to 128×128, and batch size to 1. We apply our random multi-view render system to capture a combined image with four sub-images with rotation angle α set to $90°$. We use AdamW optimizer [Kingma and Ba, 2014] with learning rate 1×10^{-2} and 1×10^{-3} for geometry and background modeling. The background is replaced with random colors with 80% of chance. In the DMTet [Shen *et al.*, 2021] setting, most of the parameters stay the same, but in the self-boost stage, we increase the resolution to 512×512 for a better result. The initialization stage of 3D Gaussian Splatting [Kerbl *et al.*, 2023] is somehow different from the other two methods as they use hash-grid while 3D Gaussian Splatting is able to initialize from point cloud representation directly. The rendering resolution is also 512×512.

We apply the CFG trick and negative prompts following the example from MVDream [Shi *et al.*, 2023], further append prompt ", 3d asset" or ", multi-view of the 3d asset" to get a more consistent result.

B Simply Combination Ablation Study

Our BoostDream method does not just simply combine the feed-forward approach with the SDS-based method. To further test the benefits of applying our multi-view based strategy, we also design an ablation study using DreamFusion [Poole *et al.*, 2022] with the same initialization stage as our method. We use the results from Shap-E [Jun and Nichol, 2023] in the initialization stage and use the same prompt text as input to optimize the NeRF representation with DeepFloyd [StabilityAI, 2023]. The results of the original DreamFusion, the DreamFusion with initialization stage, and our BoostDream-NeRF are shown in Figure 1. We can see in the first row even with the proper initialization, DreamFusion still suffers from the Janus problem and has coarse results compared to our BoostDream results.

C Control Condition Ablation Study

We also test our method with different multi-view control conditions replacing the normal map. We choose canny edge [Canny, 1986] and depth map [Ranftl *et al.*, 2020] as guidance obtained through the same multi-view render system as normal map. The results are shown in the Figure 2. Canny edge just contains the edge information of the 3D asset. Intuitively, it is unsuitable as a control condition when generating high-quality 3D assets. The results also illustrate this point: when using canny edge as the control condition, the 3D asset suffers from incomplete generation. Especially in the second row, the bear turns out to be unnatural and has strange colors. Instead of canny edge using edge information to guide the refinement process, the depth map utilizes depth information, leading to complete generation results. However, we find that the generated results are less detailed when the control condition is depth map. This can be explained by the fact that minor details information is not prominent in depth map but salient in normal map [Zhang *et al.*, 2023]. We can further validate this idea with the last column, the generated 3D assets are high-quality and with more details when under the guidance of normal map.

D Result on Different 3D Representations

This section supplements the comparison experiment in Section 4.3. We implement our BoostDream on other differentiable representations, including DMTet [Shen *et al.*, 2021] and 3D Gaussian Splatting [Kerbl *et al.*, 2023]. The results are shown in Figure 3, illustrating the generality of our method in generating high-quality assets using different differential 3D representations.

References

[Canny, 1986] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[Jun and Nichol, 2023] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

[Kerbl *et al.*, 2023] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Mildenhall *et al.*, 2021] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[Poole *et al.*, 2022] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[Ranftl *et al.*, 2020] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

[Shen *et al.*, 2021] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[Shi *et al.*, 2023] 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.

[StabilityAI, 2023] StabilityAI. Deepfloyd. https://huggingface.co/DeepFloyd, 2023.

[Zhang *et al.*, 2023] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

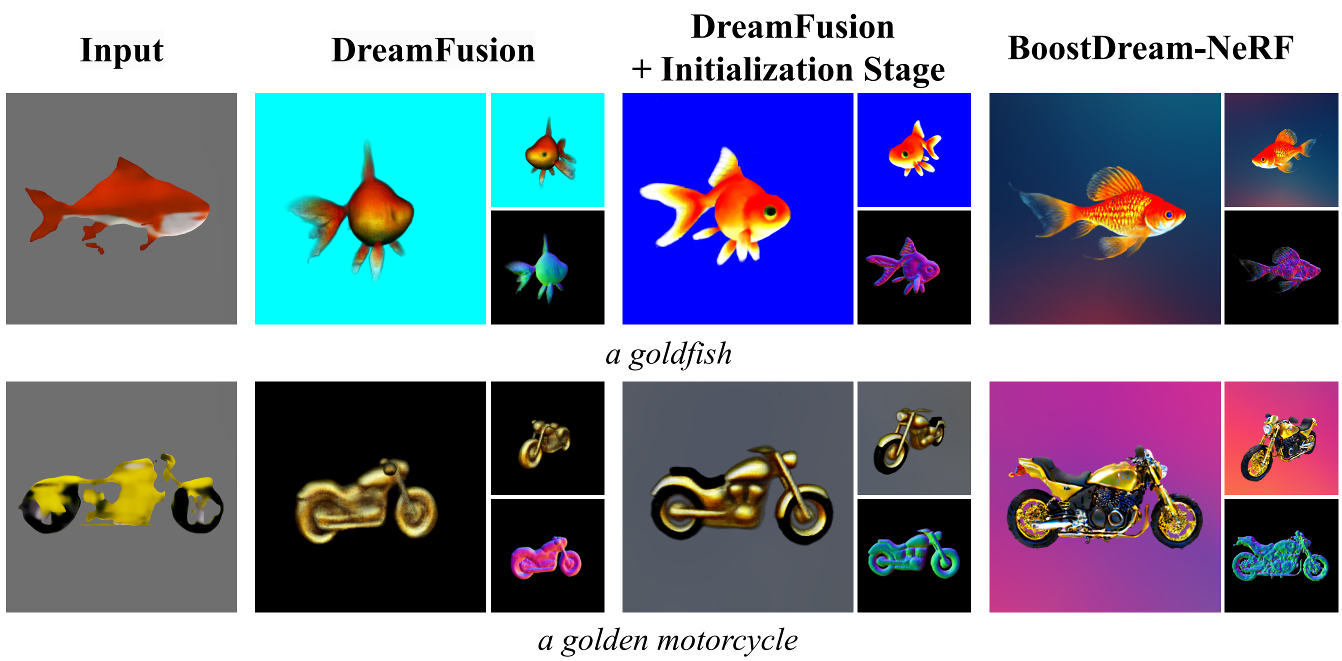

Figure 1: Simply Combination Ablation Study. The first column is the input coarse model generated by Shap-E [Jun and Nichol, 2023], while the next three columns are the results for the original DreamFusion [Poole *et al.*, 2022], the DreamFusion with initialization stage, and our BoostDream-NeRF, respectively.

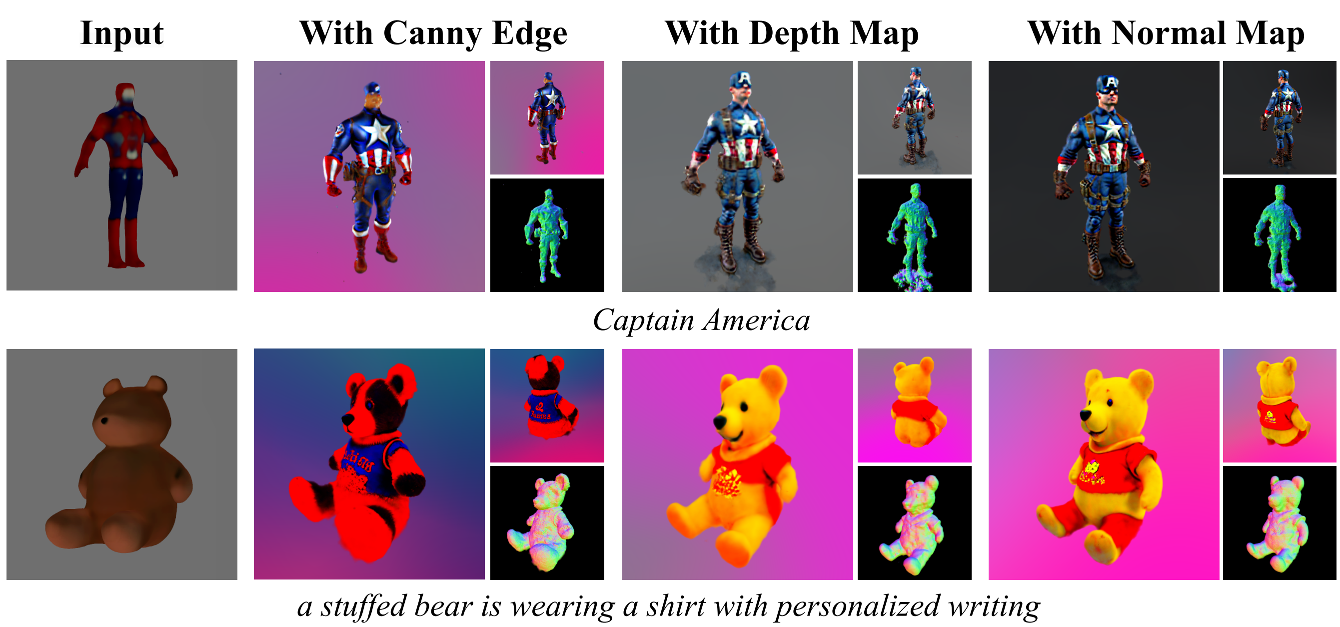

Figure 2: Control Condition Ablation Study. The first column is the input coarse model generated by Shap-E [Jun and Nichol, 2023], while all other columns are the output of our BoostDream method with different control conditions.

Input
BoostDream-NeRF
BoostDream-DMTet
BoostDream-3D Gaussian Splatting
a Gundam with a weapon
a pagoda in Japanese style on the cloud
a Chinese garden tree
a brown and white horse is wearing a blue saddle

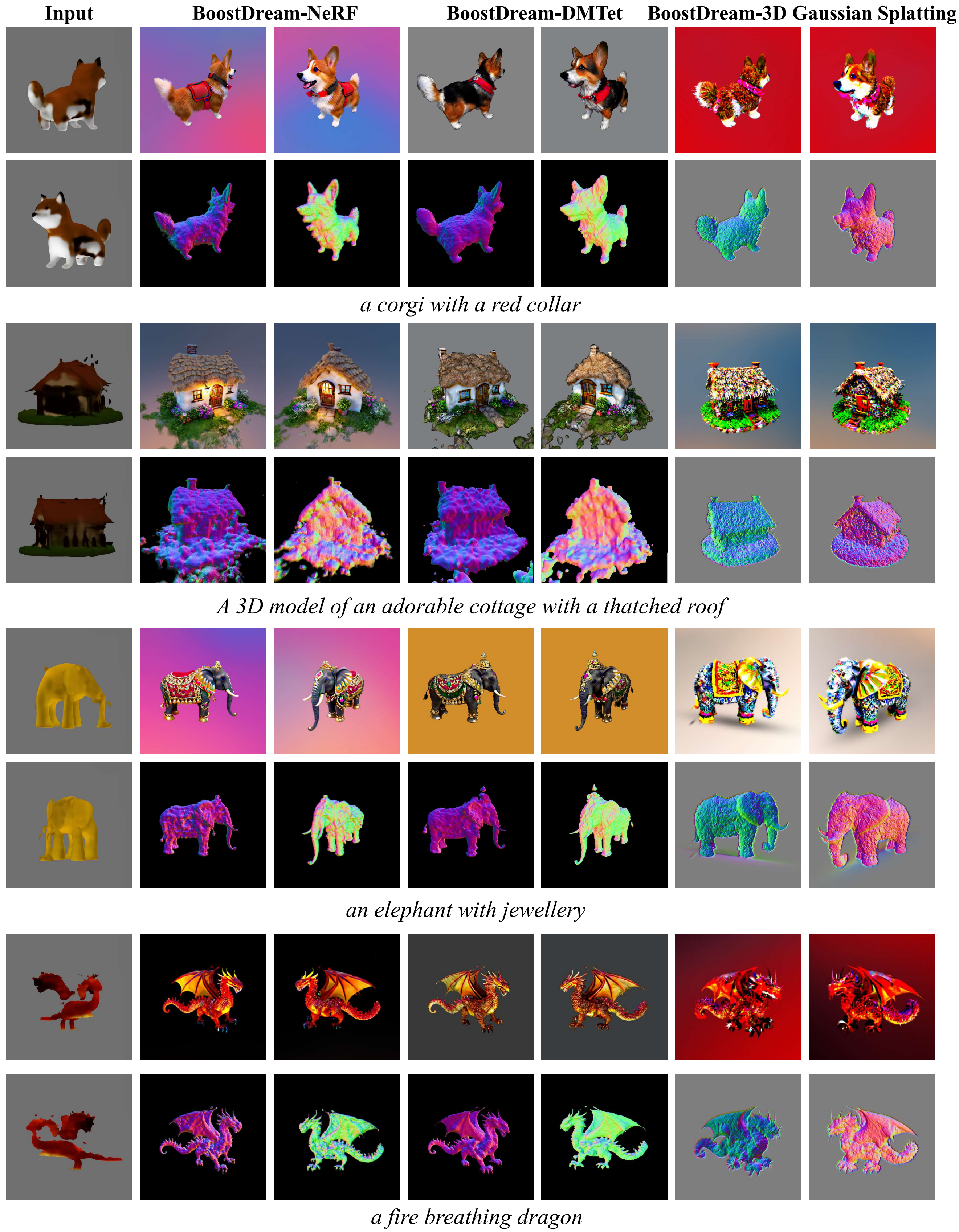

Figure 3: Result on Different 3D Representations.The first column is the input coarse model generated by Shap-E [Jun and Nichol, 2023], while the next three columns are the results of our BoostDream method implemented with NeRF [Mildenhall *et al.*, 2021], DMTet [Shen *et al.*, 2021] and 3D Gaussian Splatting [Kerbl *et al.*, 2023], respectively.