Supplementary Materials to "Progressive Rendering Distillation: Adapting Stable Diffusion for Instant Text-to-Mesh Generation without 3D Data"

The contents of this supplementary file include:

- Progressive Rendering Distillation pseudo code (referring to Sec. 3.2 in the main paper).
- More implementation details (referring to Sec. 4.1 in the main paper).
- Additional qualitative comparisons (referring to Sec. 4.2 in the main paper).
- Additional results with expanded training corpus (referring to Sec. 4.2 in the main paper).
- Additional ablation experiments (referring to Sec. 4.3 in the main paper).

1. Pesudo Code

The pseudo-code of our Progressive Rendering Distllation (PRD) training scheme is appended in Algorithm 1.

2. More Implementation Details

Dual rendering. We integrate DiffMC [15] for mesh rasterization and NeuS [14] for volume rendering to supervise the generation of 3D outputs. Such a dual rendering approach can ensure the training stability: when SDF values are all positive or all negative throughout the 3D space and thus the mesh extraction fails, volume rendering can still guide the training process to optimize the 3D space. Due to memory constraints, volume rendering is limited to low resolution (128×128). We complement this with high-resolution (512×512) mesh rasterization. To handle mesh extraction failures caused by the uniformly distributed SDF signs, we implement the method proposed in [18] to artificially enforce the position of zero-level set in the 3D space. We manually control gradient magnitudes during backpropagation. The gradient of volume rendered multi-views with respect to the texture decoding MLP starts at 1.0 and linearly decreases to 0.01 at the end of training, preventing from blurry textures caused by low-resolution volume rendering supervision. The gradient of mesh rasterized multi-views with respect to both the SDF decoding MLP and deformation decoding MLP is fixed at 0.001 throughout training, which stabilizes training and improves generation performance.

Training objective. With the multi-view teacher [9–11], we decode the multi-views $\boldsymbol{x}_\pi$ to latent $\boldsymbol{z}_\pi$, which are diffused by adding Gaussian noise at timestep t [3], denoted by $\boldsymbol{z}_{\pi,t}$. We write the diffusion module of the multi-view teacher as $\boldsymbol{z}_{\phi^{2D}}(\boldsymbol{z}_{\pi,t}; t, \pi, y)$ to represent the process of noise prediction and latent denoising, where y is the text prompt. With ASD [6], the derivative of the objective with respect to the 3D generator $\phi^{3\text{D}}$ is:

$$\nabla_{\phi^{3\text{D}}} \mathcal{L}_{\phi^{2\text{D}}}(\boldsymbol{x}_\pi; \pi, y) = \mathbb{E}_{t,\boldsymbol{\epsilon},\Delta t}\left[\omega(t)\left(\boldsymbol{z}^{Cls}_{\phi^{2D}}(\boldsymbol{z}_{\pi,t}; t, \pi, y) - \boldsymbol{z}_{\phi^{2D}}(\boldsymbol{z}_{\pi,t+\Delta t}; t+\Delta t, \pi, y)\right)\frac{\partial \boldsymbol{z}_\pi}{\partial \phi^{3\text{D}}}\right], \tag{1}$$

where ϕ^{2D} denotes the teacher model parameters, t is sampled from $\mathcal{U}[T_{\text{Min}}, T_{\text{Max}}]$ with $0 < T_{\text{Min}} < T_{\text{Max}} < T = 1000$, and Cls indicates classifier-free guidance (CFG) [2]. By introducing a timestep shift Δt [6] sampled from a uniform distribution $\mathcal{U}[0, \eta(t - T_{\text{Min}})]$, ASD achieves more effective training of the native 3D generator. We utilize the timestep-dependent weighting factor from DMD [20], as implemented in [13, 16]. We let

$$\omega(t) = \frac{1}{\text{NoGrad}(\text{Mean}(\boldsymbol{z}_\pi - \boldsymbol{z}^{Cls}_{\phi^{2D}}(\boldsymbol{z}_{\pi,t}; t, \pi, y))) + \delta}, \tag{2}$$

where NoGrad detaches gradients for loss back-propagation, and Mean applies L_1-norm across height, width, channel dimensions and all rendered views. Unlike [13, 16, 20], we add constant $\delta = 0.1$ to the denominator, which stabilizes

Algorithm 1 Progressive Rendering Distillation (PRD)

Input: SD-based native 3D generator with $\boldsymbol{z}_{\phi^{3\mathrm{D}}}$ and $D_{\phi^{3\mathrm{D}}}$; score distillation objective $\mathcal{L}_{\phi^{2D}}$ parameterized by multi-view diffusion model ϕ^{2D}; prompt corpus $\mathbb{S}_y$; number of rendered views N; number of steps K

1 Initialize optimizer Opt for $\boldsymbol{z}_{\phi^{3\mathrm{D}}}$ and $D_{\phi^{3\mathrm{D}}}$
2 Define fixed timesteps $T = t_1 > t_2 > \cdots t_K > 0$
3 **while** *not converged* **do**
4   Sample text prompt $y \in \mathbb{S}_y$
5   Sample $\hat{z_0} \sim \mathcal{N}(0, \boldsymbol{I})$
6   **for** $t \leftarrow t_1$ **to** t_K **do**
7   Sample $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$
8         $\boldsymbol{z}_t \leftarrow \alpha_t \hat{\boldsymbol{z}}_0 + \sigma_t \epsilon$
9         $\hat{\boldsymbol{z}_0} \leftarrow \boldsymbol{z}_{\phi^{3\mathrm{D}}}(\boldsymbol{z}_t; t, y)$
10         $\hat{\theta} \leftarrow D_{\phi^{3\mathrm{D}}}(\hat{\boldsymbol{z}_0})$
11   Sample K camera poses $\pi_1, \ldots, \pi_N$
12   **for** $i \leftarrow 1$ **to** N **do**
13             $\boldsymbol{x}_{\pi_i} \leftarrow g(\hat{\theta}, \pi_i)$
14   **end**
15         $\boldsymbol{L} \leftarrow \mathcal{L}_{\phi^{2D}}(\boldsymbol{x}_{\pi_1}, \ldots, \boldsymbol{x}_{\pi_N}; \pi_1, \ldots, \pi_N, y)$
16   Save $\frac{1}{K} \nabla_{\phi^{3\mathrm{D}}} \boldsymbol{L}$ in Opt
17   **end**
18   Update $\boldsymbol{z}_{\phi^{3\mathrm{D}}}$ and $D_{\phi^{3\mathrm{D}}}$ with gradient saved in Opt
19 **end**
20 **return** $\boldsymbol{z}_{\phi^{3\mathrm{D}}}$ and $D_{\phi^{3\mathrm{D}}}$

Figure S1. Qualitative comparison with PI3D [5].

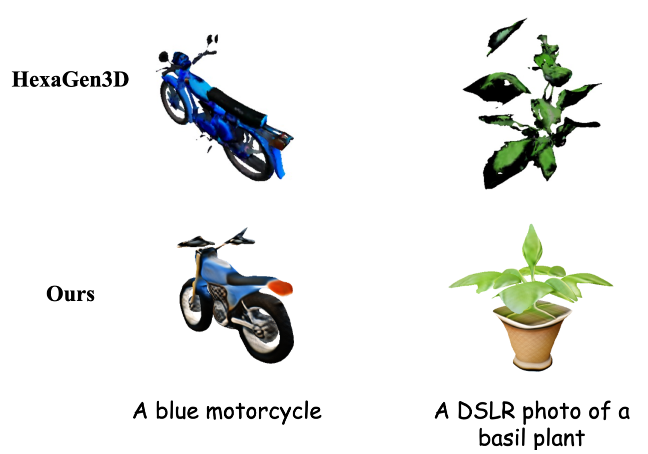

Figure S2. Qualitative comparison with HexaGen3D [7].

training and improves generation performance. We apply this objective function to supervise 3D outputs using three teacher models (SD, MV, RD) and two rendering pipelines (volume rendering and mesh rasterization). Regarding the sampling range of timestep t, $T_{\text{Max}} = 980$ throughout training, while T_{Min} starts at 500 and linearly decreases to 20. Teacher-specific hyperparameters vary: RD uses CFG=20 and $\eta = 0.1$; MV uses CFG decreasing from 20 to 10 and $\eta = 0$; SD uses CFG=5 and $\eta = 0$. Setting $\eta = 0$ for multi-view teachers that supervise RGB renderings aligns with the findings in PiSA-SR [13]. Additionally, we incorporate regularization terms during training, such as sparsity loss [8] and eikonal loss [19]. We linearly reduce the sparsity and eikonal loss weights from 1 to 0 throughout the training process.

Noise schedule. The PRD training incorporates progressive noise addition to the denoised latents (see Line 8 in Algorithm 1). Being adapted from SD [10], our native 3D generator follows the DDPM [3] noise schedule in training. During inference, we employ DDIM [12].

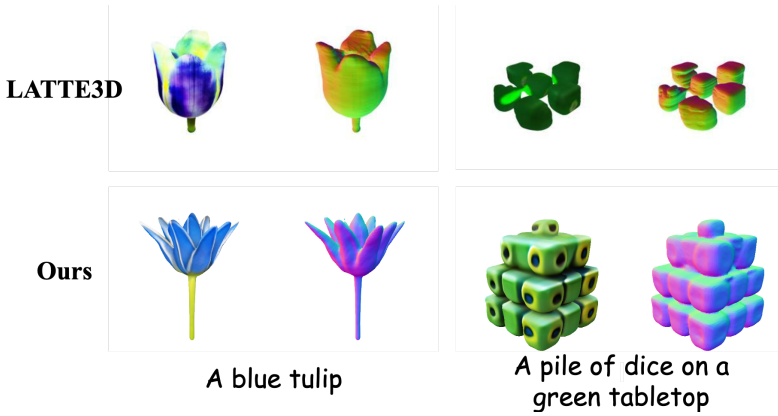

Figure S3. Qualitative comparison with LATTE3D [17].

3. More Qualitative Comparison Results

Comparison with methods adapting SD as native 3D generators. Since the codes or trained models of current SD-based native 3D generators [5, 7] are not publicly available, we conduct our comparisons by using their visual results presented in the original publications. The qualitative comparisons with PI3D [5] and HexaGen3D [7] are presented in Fig. S1 and Fig. S2, respectively. As both the two compared methods employ data-driven training, they inherit pose inconsistencies existed in the 3D training datasets [1], leading to the issue of occasional pose misalignment. This can be clearly observed from PI3D's result of *'A dalmatian wearing a fireman's hat'* shown in Fig. S1, where the dog is oriented sideways. The comparison results demonstrate our method's superior visual fidelity with the input prompts. These improvements are attributed to our proposed Progressive Rendering Distillation (PRD) scheme, which utilizes multi-view teachers in training without requiring 3D training data.

Comparison with native 3D generators trained with score distillation. We further compare our approach with existing methods that employ score distillation for native 3D generator training. Specifically, we compare against the current state-of-the-art method, LATTE3D [17]. Since the code or model of LATTE3D is unavailable, we conduct qualitative comparisons using their published results. The visual comparisons are presented in Fig. S3. It can be seen that our method demonstrates notable improvements in both texture fidelity and geometric accuracy. For example, in *'A blue tulip'*, our model captures more natural flower textures, while in *'A pile of dice on a green tabletop'*, our model achieves more precise geometric structures. These improvements can be attributed to our strategic adaptation of SD as the backbone architecture, which allows our model to leverage its powerful generative capabilities.

4. Expanding Training Corpus

Since our proposed training scheme does not require 3D ground truth data, it can be easily up-scaled to a large amount of text prompts. We collect a total number of 1.7 million text prompts from HuggingFace that were used to generate images by DALL-E and Midjourney. This corpus has more unnatural prompts than the Objaverse [1], and it is more challenging. To the best of our knowledge, our work is the first that can process more than 1 million training data. Our model, trained on this expanded dataset, achieves enhanced visual quality, as demonstrated in Fig. 1 in the main paper and Fig. S4, Fig. S5 in this supplementary file.

	C.S. ↑	R@1 ↑
w/o SD	63.0	20.1
w/o MV	67.4	25.9
w/o RD	41.5	11.4
w/ All (Proposed)	**68.2**	**32.3**

Table S1. Ablation study on jointly using SD, MV and RD as multi-view teachers.

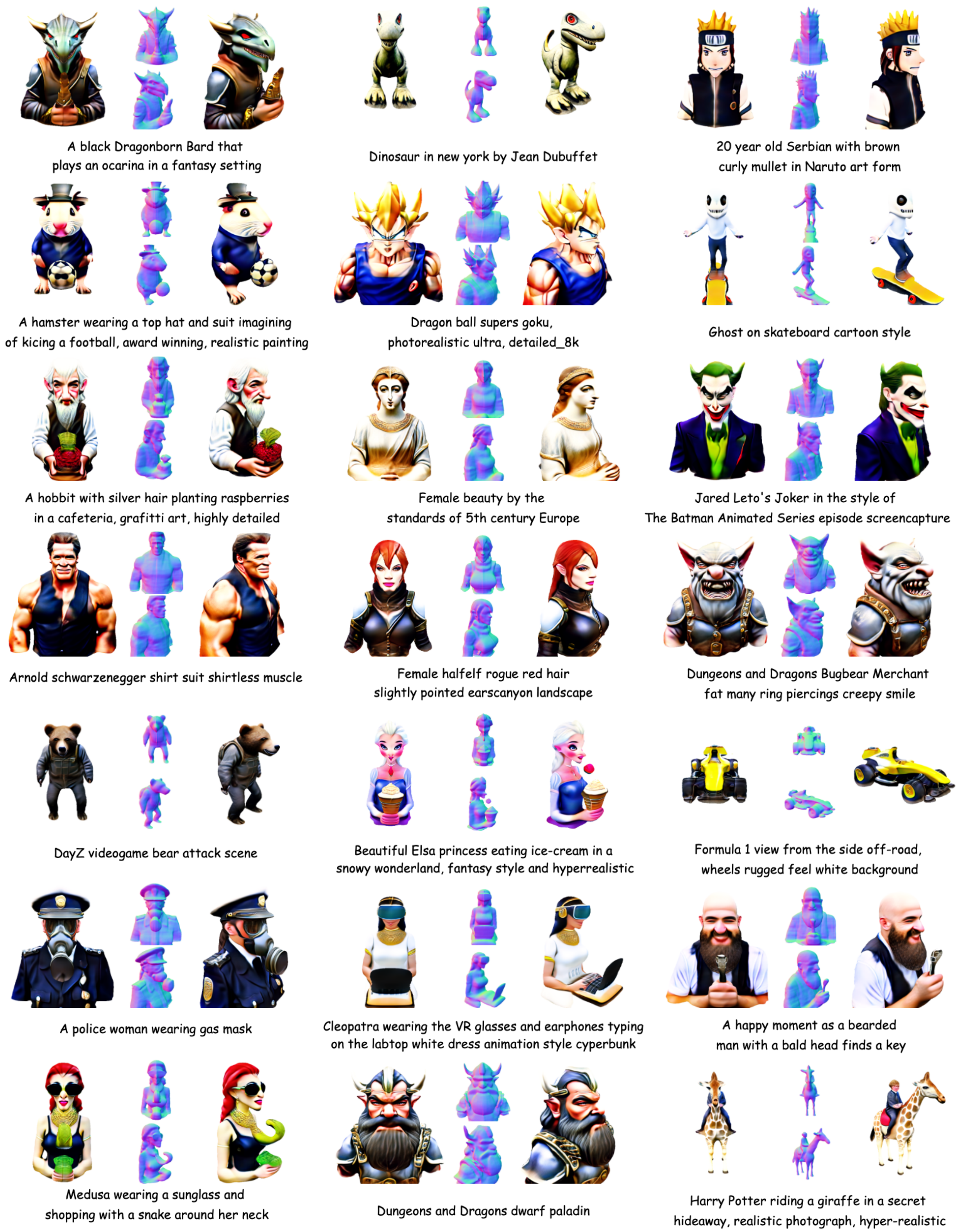

Figure S4. More results of our model trained with expanded corpus.

Figure S5. More results of our model trained with expanded corpus.

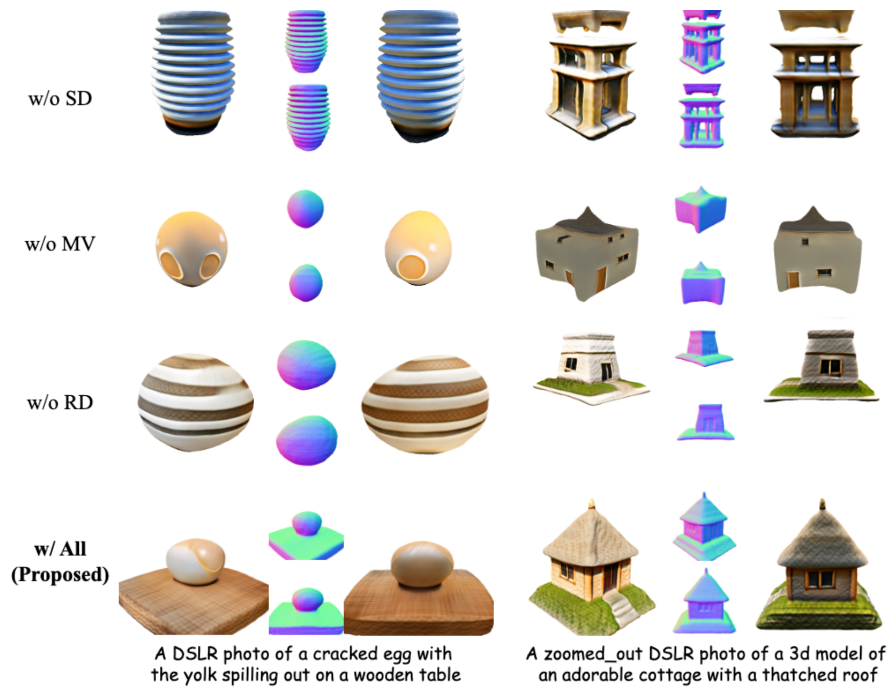

Figure S6. Visualizations for the ablation study on jointly using SD, MV and RD as multi-view teachers.

5. More Ablation Studies

The necessity of multiple teachers. We employ SD [10], MV [11] and RD [9] as teachers for multi-view supervision of RGB, normal and depth maps. Here we perform ablation studies by systematically removing individual components.

First, as visualized by **w/o SD** in Fig. S6, when SD is removed, leaving only MV and RD as teachers, the model can collapse to generate results inconsistent with text prompts. For example, given the prompt *'A DSLR photo of a cracked egg with the yolk spilling out on a wooden table'*, the model collapses to generating a stack of discs. This occurs because training for multi-view generation may impair MV and SD's text understanding capabilities, resulting in outputs that diverge from the specified text descriptions. SD can prevent from training collapse and improve the generation stability. Second, the importance of MV is demonstrated by the visualizations of **w/o MV** in Fig. S6. Without multi-view RGB supervision, the generated results tend to show repetitive and redundant contents across different viewpoints. For instance, multiple *'egg yolks'* might appear in the results of *'A DSLR photo of a cracked egg with the yolk spilling out on a wooden table'*. Finally, the importance of RD is validated by the visualizations of **w/o RD**. We can see that adding normal and depth constraints enhances text-consistency in the outputs, such as the generated *'wooden table'* in the results of *'A DSLR photo of a cracked egg with the yolk spilling out on a wooden table'*. Overall, the combination of SD, MV, and RD as teachers achieves the best results, as validated by the visualization of **w/ All** and the metrics shown in Tab. S1.

The necessity of dual rendering. We use a dual rendering framework that integrates mesh rasterization [4] and volume rendering [19] for 3D output supervision, as detailed in Sec. 2. The effectiveness of this dual approach is demonstrated through quantitative and qualitative evaluations in Tab. S2 and Fig. S7, respectively. Without volume rendering, relying solely on mesh rasterization leads to training collapse and invalid mesh extraction. The results labeled as **w/o Volume Rendering** in Fig. S7 demonstrate that training converges to a state where the SDF's zero-level set vanishes, resulting in mesh extraction failure and empty space. Conversely, using only volume rendering, which is constrained to low-resolution training, fails to produce high-quality mesh geometry, leading to rough and coarse textural details, as shown by **w/o Mesh Rasterization** in Fig. S7. For example, it fails to produce the shining gold texture for the prompt *'A DSLR photo of a toilet made out of gold'*. Moreover, without direct mesh supervision, volume rendering-based methods may produce geometrically invalid structures. This limitation is evident in the result of *'A DSLR photo of aerial view of a ruined castle'*, where the extracted meshes exhibit incorrect structural features and poor textures, manifesting as gray regions in parts of the mesh. As shown by **w/ Both** in Fig. S7 and supported by the superior metrics in Tab. S2, our dual rendering approach enables stable training while producing meshes with detailed textures and well-defined geometric structures.

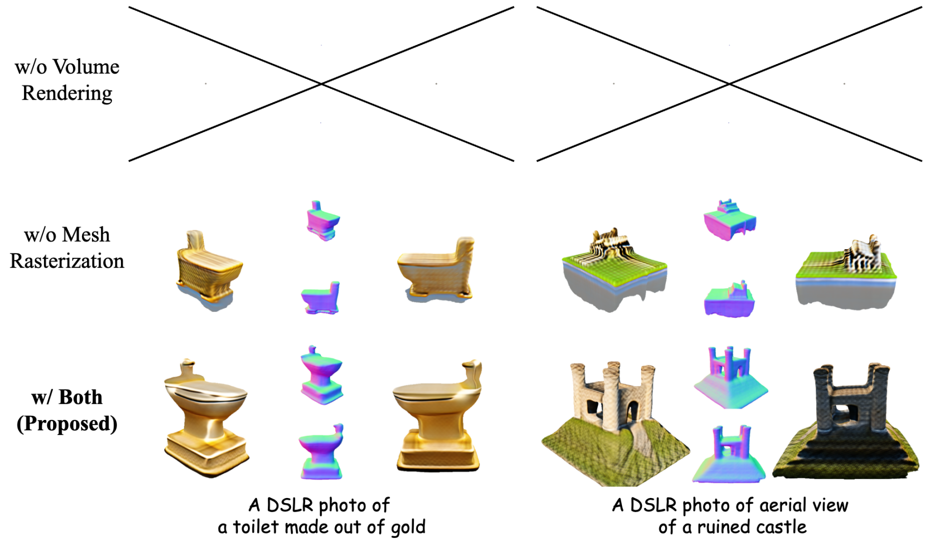

Figure S7. Ablation study on dual rendering. The cross mark means the model fails to generate mesh due to training instability.

Figure S8. Visualization for the ablation study on the LoRA rank in PETA.

	C.S. ↑	R@1 ↑
w/o Volume Rendering	25.1	0.01
w/o Mesh Rasterization	67.4	25.9
Joint (Proposed)	**68.2**	**32.3**

Table S2. Ablation study on the dual renders.

	C.S. ↑	R@1 ↑
rank=8	62.9	15.6
rank=16 (Proposed)	**68.2**	**32.3**
rank=32	66.2	26.6

Table S3. Ablation study on the LoRA rank in PETA.

The choice of LoRA rank. We demonstrate the significance of using a LoRA rank of 16 in our Parameter-Efficient Triplane Adaption (PETA). With a lower rank of 8, shown as **Rank=8** in Fig. S8, the model exhibits insufficient learning capacity, as evidenced by its failure to generate the top hat structure for the prompt *'A capybara wearing a top hat, low poly'*. However, setting a higher rank, such as 32, can also lead to unreasonable geometric outputs. As shown in **Rank=32** in Fig. S8, some unwanted platform structures appear at the bottom of results of *'A capybara wearing a top hat, low poly'* and *'A zoomed out DSLR photo of a baby dragon'*. Such artifacts stem from the inherent generation biases in both MV and SD, as their training dataset [1] contains numerous examples where objects rest on square platforms. As a result, the multi-view teachers are fitted to generate outputs with similar structures. When the LoRA rank is set too high, the native 3D generator tends to learn and reproduce the biases from the teachers during the distillation. Setting the rank to a balanced value of 16 enables the model to generate text-aligned 3D results while avoiding the incorporation of undesirable biases into the 3D generation model. Denoted as **Rank=16**, both qualitative results in Fig. S8 and quantitative results in Fig. S8 show that a rank of 16 yields the best performance.

References

[1] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A universe of 10m+ 3d objects. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 3, 7

[2] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[4] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8496–8506, 2023. 6

[5] Ying-Tian Liu, Yuan-Chen Guo, Guan Luo, Heyi Sun, Wei Yin, and Song-Hai Zhang. Pi3d: Efficient text-to-3d generation with pseudo-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19915–19924, 2024. 2, 3

[6] Zhiyuan Ma, Yuxiang Wei, Yabin Zhang, Xiangyu Zhu, Zhen Lei, and Lei Zhang. Scaledreamer: Scalable text-to-3d synthesis with asynchronous score distillation. *arXiv preprint arXiv:2407.02040*, 2024. 1

[7] Antoine Mercier, Ramin Nakhli, Mahesh Reddy, Rajeev Yasarla, Hong Cai, Fatih Porikli, and Guillaume Berger. Hexagen3d: Stablediffusion is just one step away from fast and diverse text-to-3d generation, 2024. 2, 3

[8] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022. 2

[9] 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, 6

[10] 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, 6

[11] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 1, 6

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

[13] Lingchen Sun, Rongyuan Wu, Zhiyuan Ma, Shuaizheng Liu, Qiaosi Yi, and Lei Zhang. Pixel-level and semantic-level adjustable super-resolution: A dual-lora approach. *arXiv preprint arXiv:2412.03017*, 2024. 1, 2

[14] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 1

[15] Xinyue Wei, Fanbo Xiang, Sai Bi, Anpei Chen, Kalyan Sunkavalli, Zexiang Xu, and Hao Su. Neumanifold: Neural watertight manifold reconstruction with efficient and high-quality rendering support. *arXiv preprint arXiv:2305.17134*, 2023. 1

[16] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *arXiv preprint arXiv:2406.08177*, 2024. 1

[17] Kevin Xie, Jonathan Lorraine, Tianshi Cao, Jun Gao, James Lucas, Antonio Torralba, Sanja Fidler, and Xiaohui Zeng. Latte3d: Large-scale amortized text-to-enhanced3d synthesis, 2024. 3

[18] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 1

[19] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2, 6

[20] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024. 1