Make-It-3D: High-fidelity 3D Creation from A Single Image with Diffusion Prior
Supplementary Material

A. Broad Impact

We have presented *Make-It-3D*, a novel approach to create novel views from a single image of general genre. Make-It-3D first hallucinates the 3D geometry by the usage of depth prior at the frontal view and the geometry prior of a pretrained diffusion model to ensure plausibility at novel views. Motivated by the fact that human eyes are more sensitive to texture over geometry, we thus reuse the coarse 3D geometry estimated from the implicit representation as well as the texture from the reference image, and specifically "inpaints" the texture of explicit 3D representation at occluded regions, ultimately producing compelling novel view renderings with highly-detailed texture.

Our primary aim is to advance the research of generative modeling from 2D to 3D. Without relying on 3D training data that is hardly accessible in scale, this work tackles the 3D synthesis problem by lifting 2D generated images to 3D. This way essentially builds on the assumption that a diffusion model not only generates 2D observations but also implicitly contains rich 3D understanding of the scene. Thus, using our technique, one can generate a 3D scene that can be immersively viewed by merely using a 2D diffusion model. Compared to DreamFusion and Magic3D, our work produces more diverse 3D synthesis results with significantly improved realism. On top of creatively generated images, this work also performs well on real images with complicated structures.

We hope this work opens the door towards high-quality 3D synthesis and inspires more following works along this way. While we have demonstrated the ability to synthesize novel views in 360 degree, it is still non-trivial to produce holistically plausible 3D objects when viewed from large viewpoints. Moreover, while this work aims for 3D synthesis from a single image, the same pipeline is applicable to the few-shot scenario where a few multi-view images can be obtained. In addition, it would be fruitful to generalize the proposed technique to augment the quality of 4D synthesis. We will release the code to facilitate the research in this emerging area.

B. Additional Implementation Details

B.1. Coarse stage

Scene representation and rendering. We use the explicit-implicit representation from Instant-NGP [2] to implement the NeRF representation in the coarse optimization stage, where we choose 16-level hash encoding of size 2^{19} and dimension 32, with a 3-layer MLP with 64 hidden units to decode the density and color for each spatial location. During volumetric rendering, we sample 96 points for each ray, including 64 points for uniform sampling and 32 for importance sampling. We initialize the density field as a Gaussian sphere, which leads to faster convergence and more stable training. Specifically, we initialize the density as $\sigma_{\text{init}} = d * \exp(-||x||^2/(2\mu^2))$, where we set density bias $d = 5$ and $\mu = 0.2$; x denotes the distance between the ray point and the scene center.

Camera setting. Following the camera sampling method used in [3], we randomly sample camera distance from 0.8 to 1.2, and the field-of-view (FOV) from 40 to 80 degrees. We find that randomly sampling FOV is instrumental to mitigate the artifacts that arise in large rendering view angles.

Augmentation and Regularization. To encourage the network to focus more on the foreground and avoid adversarial samples that hack the pretrained diffusion model, we train NeRF with a random background augmentation. Specifically, during training, we randomly jitters the background color of both the reference alpha image and NeRF rendering. During inference, we render the scene with a white background. Furthermore, following [3], we use three types of geometric regularization including sparsity, opacity and smoothness.

B.2. Refine stage

Point cloud rasterization. Following [1], we rasterize neural points V to multi-scale feature maps $\mathcal{S}(i, V)$, $i \in [0, K)$, $K = 3$. We use a differentiable point rasterizer implemented by PyTorch3D [4] to assign every pixel a neural descriptor and a binary scalar that indicates a non-empty pixel. We consider the binary mask as a point-based occupancy mask.

Background regularization. To handle pixels without cor-

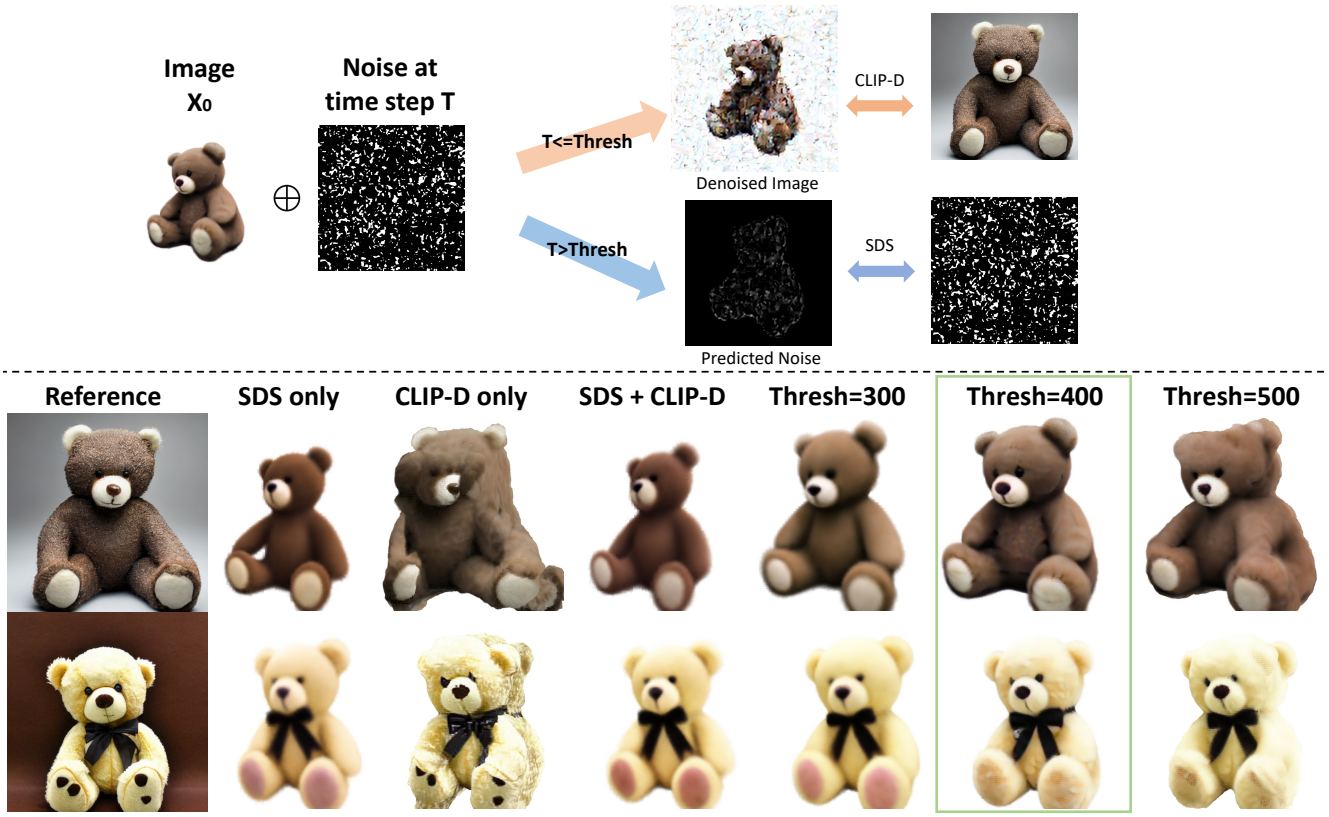

Figure 1: Analysis of SDS and CLIP-D loss.

	LPIPS↓	Contextual↓	CLIP↑
SDS	0.3045	2.29	86.04%
CLIP-D	**0.1260**	2.43	80.27%
SDS+CLIP-D	0.2772	2.32	84.01%
Thresh=300	0.1757	2.19	87.40%
Thresh=400	0.1427	**1.74**	**87.50%**
Thresh=500	0.1696	2.23	86.09%

Table 1: Ablation study on SDS and CLIP-D loss on the test benchmark. We compute LPIPS under the reference view, and the other two metrics under novel views. "Thresh" denotes the boundary of time steps using SDS or CLIP-D in the denoising process.

responding point cloud projection, we assign a learnable descriptor as the background. During texture enhancement optimization, we additionally add a regularization to encourage the scene to be rendered with a white background according to the binary occupancy mask mentioned above.

Deferred neural rendering. For deferred rendering of the point clouds, we use a 2D U-Net architecture with gated convolutions [5]. It contains 3 down- and up-sampling layers to integrate multi-scale feature maps and output the final RGB image.

C. Additional Ablation Study and Analysis

C.1. Analysis of SDS and CLIP-D loss

As mentioned in Sec 3.1, in the coarse stage, we use the diffusion prior by applying score distillation sampling (SDS) scheme on novel view renderings. It can successfully encourage the generated scene to match the conditioned text prompt. However, as an image-based 3D content creation model, we need to prioritize the faithfulness between created 3D and the reference image. Although we add pixel-wise constrain under the reference view for optimization, SDS provides a strong geometric prior and enforces the optimized scene to be a plausible result according to the text condition. Constraints under a single view can be limited. Thus the created results may not be rigorously aligned with the reference image (See Figure 1).

Therefore, we need to relax the strong geometric guidance provided by SDS and add more image-level constraints under multi-views. We achieve this goal by simultaneously maximizing the image-level similarity between the reference image and the novel view renderings denoised by the diffusion model, named as a diffusion CLIP loss $\mathcal{L}_{\text{CLIP-D}}$. Compared with introducing this constraint directly on novel view renderings, the CLIP-D encourages the pretrained dif-

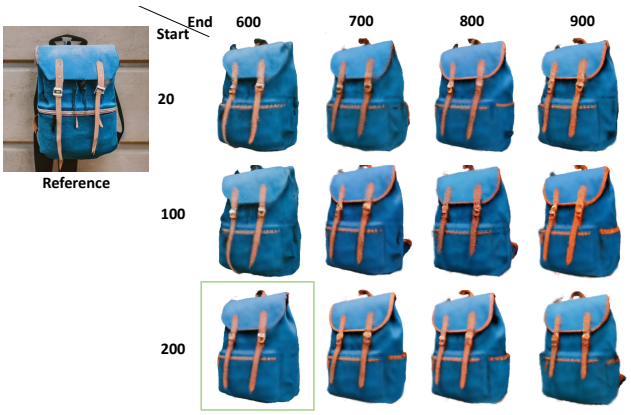

Figure 2: Analysis of the time step range in SDS process. We visualize novel view results in the coarse stage that are trained with different time step ranges (from start to end).

Figure 3: Analysis of texture initialization and point descriptors.

fusion model to provide better guidance to generate more faithful 3D content with the reference image.

In view of this, we conduct several experiments to study the effect of SDS and CLIP-D loss during optimization, which is shown in Figure 1. Results show that using only SDS generates high-quality and plausible geometry, but the optimized 3D does not align with the image. On the contrary, using only CLIP-D preserves the appearance of the reference image, but fails to generate good geometry. A simple solution is to combine the two losses, but this does not fully address the non-alignment issue. To achieve a balance between geometric quality and appearance alignment, we introduce an optimization strategy by setting a threshold of sampling steps. Specifically, we optimize CLIP-D loss at small timesteps and optimize SDS at large steps. We conduct several qualitative and quantitative studies on different threshold settings, which are shown in Figure 1 and Table 1. During training, we randomly sample noise step T from 200 to 600, and we find that $T = 400$ could balance the geometric quality and the appearance alignment.

C.2. Analysis of various sampling time step ranges

We also investigate the effect of various sampling time steps in SDS process. The experimental results are shown in Figure 2. We conduct several experiments using different sampling ranges. We observe that adding noise at large time steps can improve the geometry quality but reduce the

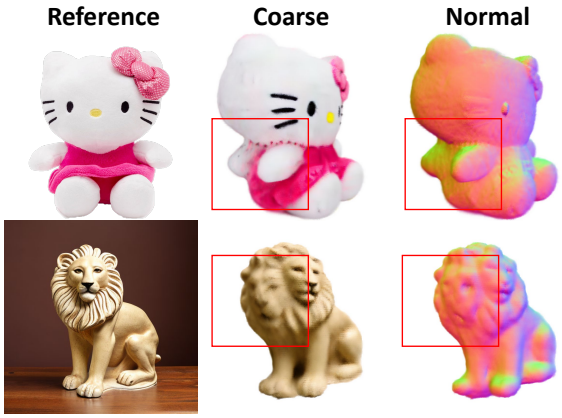

Figure 4: Failure cases due to the geometry ambiguity.

alignment and potentially saturate textures. And the diffusion prior does not provide adequate supervision at small time steps. In our method, we exclude small and large time steps and instead randomly sample time step T from 200 to 600.

C.3. Analysis of texture initialization and point descriptors

We conduct ablation studies on texture enhancement process. We explore the importance of the initialized unseen texture from NeRF and point descriptor. The qualitative results are shown in Figure 3. We can see that texture initialization is crucial for global texture enhancement. And only optimizing point color without descriptor outputs artifacts and cannot produce reasonable results.

D. Additional Results

In this section, we provide additional results of creating 3D models from different reference images using our method. The results are shown on Figure 5, Figure 6, and Figure 7. Results show that our method has a strong ability on creating high-fidelity 3D content including high-quality geometries and textures using a single reference image.

E. Limitations

Our method suffers from some geometry ambiguity, such as Janus problem or over-flat geometry [3]. A depth prior can reduce this issue. However, since we only add depth constrain at a single view, the geometry ambiguity may still exist under other views. We show some failure cases in Figure 4.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *Computer Vision–ECCV 2020: 16th European Conference,*

Figure 5: Additional results by *Make-It-3D*. The first column is the reference image. We show high-fidelity results including normal maps under the reference view and novel views.

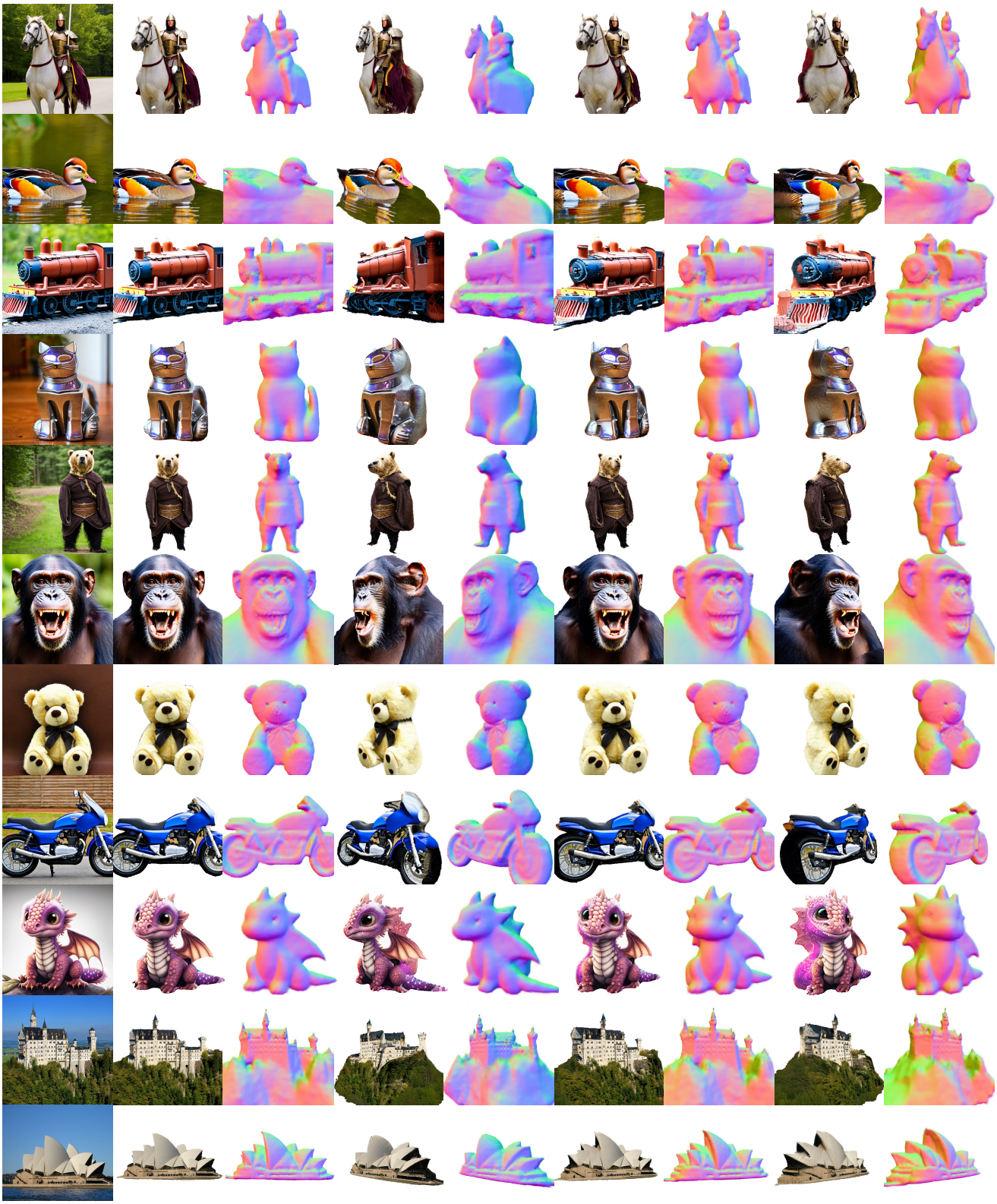

Figure 6: Additional results by *Make-It-3D*. The first column is the reference image.

Figure 7: Additional results by *Make-It-3D*. The first column is the reference image.

Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16, pages 696–712. Springer, 2020. 1

[2] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 1

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

arXiv:2209.14988, 2022. 1, 3

[4] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 1

[5] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4471–4480, 2019. 2