Direct2.5: Diverse Text-to-3D Generation via Multi-view 2.5D Diffusion
—— Supplementary Material ——

Yuanxun Lu[1] * Jingyang Zhang[2] Shiwei Li[2] Tian Fang[2] David McKinnon[2]
Yanghai Tsin[2] Long Quan[3] Xun Cao[1] Yao Yao[1] †

[1]Nanjing University
luyuanxun@smail.nju.edu.cn, {caoxun,yaoyao}@nju.edu.cn

[2]Apple
{jingyang_zhang,shiwei,fangtian,dmckinnon,ytsin}@apple.com

[3]The Hong Kong University of Science and Technology
quan@cse.ust.hk

Due to the space limitation of the main paper, we provide supplementary materials to give an auxiliary demonstration. In this PDF file, we will present a detailed description of the implementation details, additional evaluation and discussions, and more results. We also provide a project page to present video results for better visualization. Project page: https://nju-3dv.github.io/projects/direct25.

1. Implementation Details

In this section, we describe more implementation details of the proposed system, including data preparation, iterative updating, inference time, and another texturing implementation.

1.1. Dataset Preparation

We use the Objaverse [3] dataset for 2.5D training data generation, which is a large-scale 3D object dataset containing 800K high-quality models. We use the captions provided by Cap3d [5] as text prompts, which is the best 3D dataset caption method currently. Each object is firstly normalized at the center within a bounding box $[-1, 1]^3$, and we render the scene from 32 viewpoints uniformly distributed in azimuth angles between $[0°, 360°]$. The elevation is set to $0°$ and camera FoV is set to $60°$. The camera distance from the origin $(0, 0, 0)$ is set to a fixed distance equal to 1.5 times the focal length in normalized device coordinates. For lighting, we use a composition of random lighting selected from point lighting, sun lighting, spot lighting, and area lighting. RGB images and normal maps in world coordinates are rendered using a rasterizer-based renderer for each object.

Besides, we also adopt a large-scale 2D image-text dataset to improve the generation diversity following mvdream [8]. Specifically, we use the COYO-700M dataset [1], which also contains metadata like resolution and CLIP scores [6], etc. We filter the dataset with both width and height greater than 512, aesthetic scores [7] greater than 5, and watermark scores lower than 0.5, which results in a 65M-size subset. Though the filtered dataset is reduced to $1/10$ of the original size, it is still much larger than the 3D dataset. Actually, we do not consume the whole dataset within the designated training time. In the following, we describe the specific dataset usage for two proposed multi-view diffusion model training.

Text-to-normal multi-view diffusion model. As we want to generate high-quality and multi-view consistent normal maps from a single text prompt input, we are able to use all valid normal map renderings in Objaverse [3]. We filter the dataset by sorting the CLIP similarities between RGB images and captions and selecting the top 500K objects to keep a high text-image consistency. We take a similar 2D & 3D joint training strategy with MVDream [8], where 3D data and 2D data are randomly chosen in each batch with a probability of 80% and 20%, respectively. This trick can guarantee the same expected number of instances to be seen in each training step because 4 views are from the same object for 3D dataset. Also for 3D data, we add a special tag *normal map* to the end of captions to indicate the normal map prediction task. During inference, we also add this postfix to the prompt for normal map predictions.

Normal conditioned RGB multi-view diffusion model. Some samples in the Objaverse dataset has cartoonish appearance, and we would like to filter out these samples.

*This project was performed during Yuanxun Lu's internship at Apple.
†Corresponding Author

Specifically, we first filter the dataset to obtain renderings whose aesthetic scores are larger than 5, which results in a 130K subset. Then, we compute the CLIP scores between the remaining images and two pre-defined positive and negative quality description prompts [1]. We compute the ratio of the positive scores and negative scores and select the top 10K data as our training dataset. We found that this strategy successfully selected the high-quality renderings in the dataset, and works better than training on all rendering data.

1.2. Iterative Updating

In most cases, a single run of the pipeline is enough to generate high-quality results. However, for some topologies, there may be large areas unobserved by the 4 perspectives (e.g., large planar areas on the top of the object). To address this issue, we could iteratively update rendered images from novel views by the inpainting [4] pipeline to refine the texture. Specifically, we compute an inpainting mask indicating the unseen areas at a new camera viewpoint, and the invisible areas are edited given a certain noise strength. In Fig. 2, we present the results of the iterative updating. In this example, we inpaint the top views of the generated bread and fuse the resulted RGB images back to the generated model. As shown in the figure, the top areas of the bread are unseen during the first generation, and we inpaint the unseen areas in the second run. The inpainting mask is used to ensure that only the unseen areas would be modified, while other regions are kept unchanged. The final generated model (Fig 2 (e)) demonstrates the effectiveness of the strategy. During experiments, we found that 1 or 2 iterations suffice to recover the unseen areas.

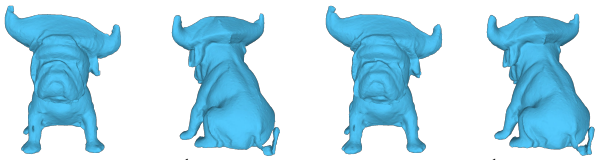
1.3. Inference Time

Compared to SDS optimization-based methods which typically take over half an hour, our method is efficient enough to generate high-quality results in 10 seconds: On a single Nvidia A100 GPU, the denoising process of the two multi-view diffusion models each takes around 2.5 seconds for 50 DDIM steps. The explicit geometry optimization takes around $2 \sim 3$ seconds for 200 optimization steps, which depends on the triangle mesh complexity. The final texture fusion takes around 1.5 seconds. the efficiency and diversity of the proposed system enable selection from batch generated samples, which greatly increases the practicality for prototyping and digital content creation. For iterative updating, typically 1-3 passes are enough to paint the unseen areas and can be finished in less than one minute, which is still much faster than the previous SDS optimization-based methods.

[1] Positive prompt: realistic, 4K, vivid, highly detailed, high resolution, high quality, photography, HD, HQ, full color; Negative prompt: cartoon, flat color, simple texture, ugly, dark, bad anatomy, blurry, pixelated obscure, unnatural colors, poor lighting, dull, unclear, cropped, lowres, low quality, artifacts, duplicate, morbid, mutilated, poorly drawn face, deformed, dehydrated, bad proportions

views \ steps	50	100	200	400	600
4-view	0.78 / 1.57	0.81 / 1.42	0.82 / 1.11	0.82 / 1.20	0.81 / 1.22
8-view	0.81 / 1.20	0.85 / 1.18	0.85 / 0.99	0.84 / 1.15	0.85 / 1.21
16-view	0.82 / 1.14	0.85 / 0.91	0.86 / 1.06	0.86 / 1.07	0.86 / 1.06

Table 1. Normal consistency (↑) and Chamfer-Distance (↓) evaluation for fast meshing under view and optimization step numbers.



(a) Opti. (400$^{\text{th}}$ step) (b) Opti. (600$^{\text{th}}$ step)

Figure 1. Visualization of more than 200 optimization steps.

1.4. Alternative Texturing Implementation

Besides the mentioned texturing method in the main paper, we also propose an alternative optimization-based texturing method as our open-source version. Similar to geometry optimization, we optimize the texture map in UV space by minimizing the reconstruction loss of the multi-view RGB images. Specifically, we adopt the L_1 RGB loss, SSIM loss, and a total variation (TV) loss on the UV texture map as a regularization. The weights for these three losses are set to 1.0, 10.0, and 1.0. In experiments, we found that only $50 - 100$ steps are enough to generate satisfactory results, and the optimization takes only about 1 second.

2. Geometry-Appearance Disentangled Generation

Due to the two-stage setting in the proposed method, one could generate random RGB images while keeping the geometry fixed, which enables geometry-appearance disentangled generation and offers better control over the generation process. Fig. 2 demonstrates the disentangled generation results. It demonstrates that users can fix the satisfying generated geometry and then proceed to appearance generation.

3. More Evaluations

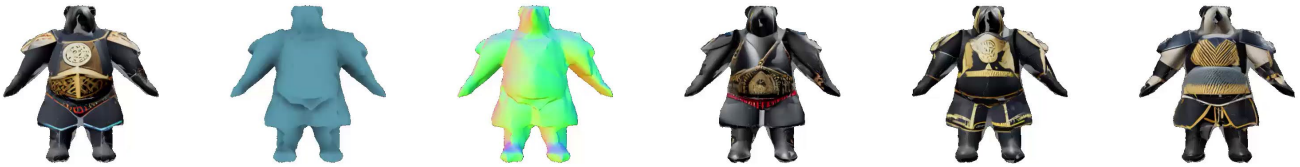
Here we present more evaluations of the proposed fast meshing algorithm. Specifically, We present normal consistency and Chamfer Distance ($\times 10^{-3}$) evaluation w.r.t. optimization steps (Tab. 1) of the fast meshing on 15 chosen Objaverse meshes with ground truth normal maps. In most cases, we found no obvious improvements beyond 200 steps, as also shown in Fig. 1.

The number of views for reconstruction is constrained by the diffusion model: SD2.1 512x512 resolution aligns

(a) Single-Pass Generation Results (c) Inpainting Mask under the New View

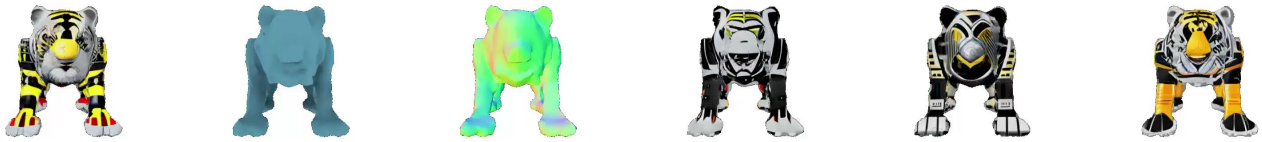
(b) Rendered Images under a Top View (d) Generated Results under the New View (e) Final Inpainted Models

Figure 2. Demonstration of the iterative updating. (a) is the single-pass generated multi-view RGB images given a prompt "a freshly baked loaf of sourdough bread on a cutting board". (b) shows the rendered results of the single-pass generated model. As seen, the top area remains uncolored. (c) shows the generated inpainting mask under the new view, where the white areas denote the areas that are invisible and need to be inpainted. (d) is the inpainted results under the new view given the previously rendered results and the visibility mask. (e) demonstrates the final generated mesh under the top view and two side-top views. The previous uncolored areas now have been inpainted with reasonable and coherent colors.

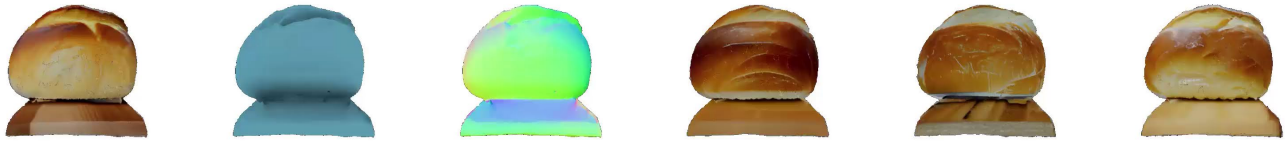


a DSLR photo of a bear dressed in medieval armor

a zoomed out DSLR photo of a wizard raccoon casting a spell



a DSLR photo of a robot tiger



a freshly baked loaf of sourdough bread on a cutting board

Figure 3. Demonstration of Geometry-appearance disentangled generation. Due to the two-stage sequential setting, our method greatly increases the control ability of the content generation results.

with four 256x256 views. Our early experiments suggested optimal quality when the total pixel count aligns with the base model's resolution. So we also do this ablation study by feeding ground truth to the optimization like the previous one. We find that more views lead to slightly better reconstruction, and leave this for future work.

4. Discussions

In the following, we provide a detailed discussion about the settings of our system, including the two-stage sequential models, and normal predictions v.s. depth predictions.

Two-stage sequential architecture. As demonstrated in Sec. 2, a two-stage sequential architecture naturally enables the geometry-appearance disentangled generation and provides more freedom on both geometry and appearance generation. Besides, using a combined pipeline also leads to a double GPU memory requirement compared to the sequential setting, which could become a great burden under the multi-view setting. This challenge becomes much more severe when one increases the spatial resolution of the diffusion model, e.g. from 256 to 512 or even 1024. Finally, the sequential model has better multi-view and geometry-appearance consistency. Instead of the generation normal maps, we use the ones rendered from the optimized mesh for the texture diffusion model input. On the one hand, the rendered normal maps are guaranteed to be consistent. On the other hand, it provides better alignment between the generated RGB images and the actual geometry. For the above reasons, our system takes the two-stage sequential as our architecture.

Normal v.s. Depth. Another alternative choice for our system is to use depth instead of normal. Because normal is the first-order derivative of the depth, it is free from scale ambiguity and provides a higher tolerance for multi-view inconsistency. Optimizing depth value directly requires much higher multi-view accuracy and therefore decreases the robustness of the geometry optimization system. Previous work [9] also found that using normal priors performs better than the depth priors, which also supports our assumption. Secondly, normal serves as a better conditioning signal for RGB generation because it generally has better alignment than depth. For example, sharp normal changes result in RGB discontinuity because of shading, but in this case depth may still be smooth. Therefore, we adopted normal as our shape representations and found it worked well.

5. Limitations and Failure Cases

In the main paper, we briefly discuss the limitations of the proposed pipeline and here we present more discussions.

Multi-view consistency. The multi-view RGB/normals are generated by the self-attention mechanism in the multi-view diffusion models without any physical-based supervision, which means the multi-view consistency is not guaranteed. This is an inherent issue in multiview diffusion models like MVDream, which is known to be prone to geometric misalignment. Our first stage text-to-normal diffusion model also suffers from this issue, while we found that the issue on the normal model is smaller than that on the RGB model. Besides, the second stage adopts multiview-consistent rendered normals as input, which relieves challenges faced by one-stage models like MVDream. We will include this part in future work.

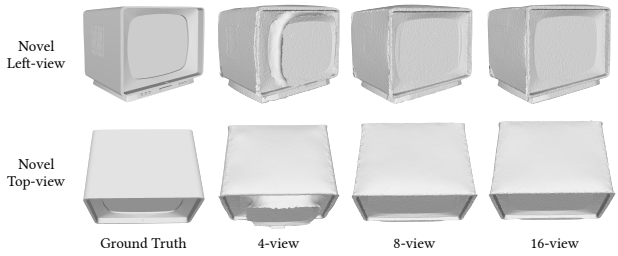


Figure 4. Failure case. The normal-based reconstruction system suffered from the depth ambiguity issue. In this example, 4-view reconstruction fails on the television screen and introduces artifacts. Using more views solves this problem.

Limited view numbers and failure cases. Because the number of views is small, areas such as top, bottom, and concavity cannot be fully observed, and thus their geometry or appearance cannot be well reconstructed. Apart from the iterative update scheme, the multi-view diffusion model can be further extended to handle more views.

We also emphasize that the limited view of normal maps may not provide sufficient information for reconstruction, leading to a degraded performance as shown in Tab. 1. We also present an example in Fig. 4. This is due to the intrinsic issue brought about by the normals being second-order derivatives of world positions, which introduces ambiguities in shape, i.e., we only know its direction in the world coordinate system, but not its specific depth, because the normal maps are the same for any depth. As shown in Fig. 4, the television screen failed to reconstruct given only 4 views normals. The full screen overall protrudes, but under the training viewpoints, there is no difference in normals - they are pointing in the same direction. This is an inherent issue with reconstruction based on normals, and using more viewpoints can greatly alleviate this problem. Our reconstruction system also suffers from this issue, and more views could lead to more accurate reconstruction.

Texture quality. For the appearance, we finetune a multi-view normal-conditioned diffusion model for efficiency. However, the ability to generate realistic images is degraded due to the texture quality of the 3D training samples and their rendering quality. Apart from further enhancing the training samples, we can also apply the state-of-the-art tex-

ture generation systems [2] for non-time-sensitive tasks.

6. More Results

We present more results of the proposed method on the following pages, including the various generation ability (Fig 5) and more generation results (Fig. 6, 7, 8).

7. Additional Video Results

We present video results of the proposed method in our project page: https://nju-3dv.github.io/projects/direct25.

Please check it for better visualization.

References

[1] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022. 1

[2] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023. 5

[3] 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

[4] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2

[5] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *arXiv preprint arXiv:2306.07279*, 2023. 1

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[7] Christoph Schuhmann. Improved aesthetic predictor. https://github.com/christophschuhmann/improved-aesthetic-predictor, 2022. 1

[8] 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. 1

[9] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. In *European Conference on Computer Vision*, pages 139–155. Springer, 2022. 4

Figure 5. More Diverse Generation Results. Our method avoids the common mode-seeking problem by SDS and generates diverse results.

Figure 6. Results Gallery. Given text prompts as description input, our method outputs high-quality textured triangle mesh in only 10 seconds. The generated multi-view normal and RGB images are shown beside the rendered models. Prompts for the above left column results are R1) a baby bunny sitting on top of a stack of pancakes, R2) a beautiful rainbow fish, R3) a DSLR photo of an astronaut standing on the surface of mars, R4) a steam engine train, high resolution, R5) a DSLR photo of a delicious croissant, and R6) a beautiful dress made out of garbage bags, on a mannequin. Studio lighting, high quality, high resolution. Prompts for the above right column results are R1) a bald eagle carved out of wood, R2) a DSLR photo of a robot tiger, R3) a DSLR photo of a teal moped, R4) a turtle standing on its hind legs, wearing a top hat and holding a cane, R5) a zoomed out DSLR photo of a marble bust of a fox head, and R6) a DSLR photo of a corgi puppy.

Figure 7. More generation results. Prompts for the above results from top to bottom and left to right are R1-l) a beagle in a detective's outfit, R1-2) a blue jay standing on a large basket of rainbow macarons, R1-3) a blue motorcycle, R2-1) a dragon-cat hybrid, R2-2) a DSLR photo of a bald eagle, R2-3) a DSLR photo of a bulldozer, R3-1) a DSLR photo of a hippo wearing a sweater, R3-2) a DSLR photo of a pair of tan cowboy boots, studio lighting, product photography, R3-3) a DSLR photo of a plate piled high with chocolate chip cookies, R4-1) a DSLR photo of a porcelain dragon, R4-2) a DSLR photo of a puffin standing on a rock, R4-3) a DSLR photo of a red-eyed tree frog, R5-1) a DSLR photo of a squirrel wearing a leather jacket, R5-2) a DSLR photo of a tarantula, highly detailed, R5-3) a DSLR photo of a toy robot, R6-1) a DSLR photo of an ice cream sundae, R6-2) an old vintage car, R6-3) a DSLR photo of an ornate silver gravy boat sitting on a patterned tablecloth, R7-1) a frazer nash super sport car, R7-2) a freshly baked loaf of sourdough bread on a cutting board and R7-3) a highland cow.

Figure 8. More generation results. Prompts for the above results from top to bottom and left to right are R1-l) a lionfish, R1-2) a marble bust of a mouse, R1-3) a metal sculpture of a lion's head, highly detailed, R2-1) a pig wearing a backpack, R2-2) a rabbit, animated movie character, high detail 3d model, R2-3) a ripe strawberry, R3-1) a snail on a leaf, R3-2) a squirrel dressed like Henry VIII king of England, R3-3) a wide angle DSLR photo of a colorful rooster, R4-1) a zoomed out DSLR photo of a beautiful suit made out of moss, on a mannequin. Studio lighting, high quality, high resolution, R4-2) a zoomed out DSLR photo of a fresh cinnamon roll covered in glaze, R4-3) a zoomed out DSLR photo of a model of a house in Tudor style, R5-1) a cute steampunk elephant, R5-2) a DSLR photo of a delicious croissant, R5-3) a DSLR photo of a plush t-rex dinosaur toy, studio lighting, high resolution, R6-1) a DSLR photo of an elephant skull, R6-2) a flower made out of metal, R6-3) a hotdog in a tutu skirt, R7-1) a shiny red stand mixer, R7-2) a wide angle zoomed out view of Tower Bridge made out of gingerbread and candy and R7-3) a zoomed out DSLR photo of a wizard raccoon casting a spell.