Direct and Explicit 3D Generation from a Single Image

Supplementary Material

In Appendix A, we provide an example of our generated multi-view outputs, additional comparisons on single-image 3D reconstruction, an analysis on the number of views, and a comparison with monocular depth estimators. In Appendix B, we provide more details on our model architecture. In Appendix C, we describe our experimental settings in more detail. In Appendix D, we show how to extend our approach to obtain rigged and posed meshes. In Appendix E, we discuss the limitations of our method. We also include a supplementary video that compares our method's results against baseline methods and shows additional results of our approach.

A. Additional Results

Our Multi-view Outputs. Given an input image, our method generates depth map along with RGB and Gaussian feature maps in six orthographic views (relative camera poses from the front, back, left, right, top, and bottom). In Fig. 1, we present an example of our multi-view predictions.

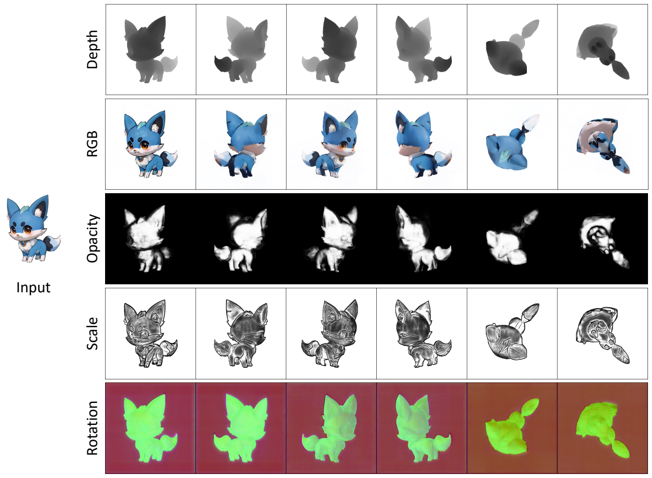

Figure 1. An example of our generated multi-view depth, RGB, and Gaussian feature images. For rotation of quaternion $\mathbf{q} \in \mathbb{R}^4$, we visualize its last three channels.

Additional Comparison on Single-image 3D Reconstruction. In Fig. 3, we provide additional qualitative comparisons with other methods on generated textured meshes on the GSO dataset [3]. Our results appear to have higher quality and better details in both geometry and texture.

Number of Views. We conduct an empirical study on the relationship between the number of views for RGB and depth images used to reconstruct a 3D object and the quality of its reconstruction. We use Objaverse dataset [2] for this study, which contains a wide range of 800K objects. We randomly sample 1,000 objects from 18 high-level categories on Objaverse dataset. We make sure that the number of objects we sample from each category matches the original percentage of that category. For the sampled objects, we attempt to reconstruct textured mesh using 4, 6, 8, or 14 views of RGB and depth images, and report the quality of the reconstruction. The view names in orders are front, back, left, and right, top, bottom, right-top-front, right-top-back, right-bottom-front, right-bottom-back, left-top-front, left-top-back, left-bottom-front, and left-bottom-back. We use screened Poisson surface reconstruction [5] to obtain the mesh. As shown in Tab. 1, increasing the number of views consistently improves reconstruction quality. The effect diminishes after 6 views, where the improvement from 4 to 6 views is significant, but further gains from 6 to 8 or 14 views are relatively smaller.

Views	CD↓	IoU↑	PSNR↑	SSIM↑	LPIPS↓
4	0.0078	0.7468	23.75	0.926	0.060
6	0.0070	0.7661	25.44	0.938	0.049
8	0.0068	0.7687	25.67	0.939	0.046
14	0.0062	0.7780	26.37	0.946	0.041

Table 1. Comparison of reconstruction quality for different numbers of views on Objaverse dataset.

Comparison with monocular depth estimators. We compare our method with other single-image depth estimation methods in Tab. 2. This study is conducted on 3D objects using the same GSO [3] evaluation dataset as in the main text. For a fair comparison, we use our predicted depth map for the front (input) view as our single-view depth estimation result. Following prior works [7, 8], we evaluate and report the mean absolute value of the relative error in depth space (AbsRel).

	MiDaS [7]	DPT [8]	Omnidata [4]	Ours
AbsRel (%)↓	17.3	13.5	12.6	**6.37**

Table 2. Comparisons on single-image depth estimation.

B. Additional Model Details

Network Architectures. As mentioned in the main text, we add a depth branch to the Stable Diffusion U-Net and incorporate epipolar attention into the Stable Diffusion VAE decoder. We compare our U-Net and decoder architectures

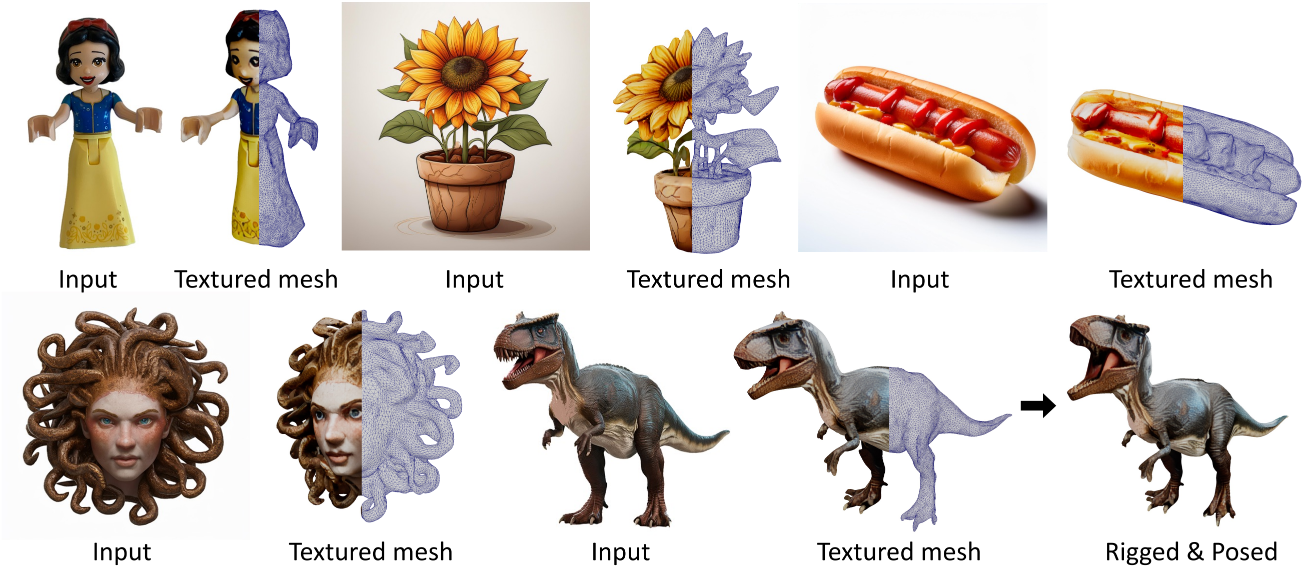

Figure 2. Refined coarse quality triangulated meshes and a rigged and re-posed example (bottom right).

U-Net	Stable Diffusion	Ours
Input/Output	B, 4, H/8, W/8	B*6, 8, H/8, W/8
Down Blocks	CrossAttnDownBlock2D CrossAttnDownBlock2D CrossAttnDownBlock2D DownBlock2D	CrossAttnDownBlockMV2D x 2 (RGB, Depth) CrossAttnDownBlockMV2D CrossAttnDownBlockMV2D DownBlock2D
Middle Block	UNetMidBlockMV2DCrossAttn	UNetMidBlockMV2DCrossAttn
Up Blocks	UpBlock2D CrossAttnUpBlock2D CrossAttnUpBlock2D CrossAttnUpBlock2D	UpBlock2D CrossAttnUpBlockMV2D CrossAttnUpBlockMV2D CrossAttnUpBlockMV2D x 2 (RGB, Depth)

Table 3. Comparison between our U-Net and Stable Diffusion U-Net [9].

Decoder	Stable Diffusion	Ours
Input	B, 4, H/8, W/8	B*6, 8, H/8, W/8
Output	B, 3, H, W	B*6, 12, H, W
Blocks	UpDecoderBlock2D UpDecoderBlock2D UpDecoderBlock2D UpDecoderBlock2D	(Epipolar) AttnUpDecoderBlock2D (Epipolar) AttnUpDecoderBlock2D (Epipolar) AttnUpDecoderBlock2D (Epipolar) AttnUpDecoderBlock2D

Table 4. Comparison between our decoder and the VAE decoder in Stable Diffusion [9].

with those in Stable Diffusion in Tab. 3 and Tab. 4, respectively.

Training Configuration. The following training configurations are applied to the fine-tuning of both the U-Net and the latent decoder.

```
training config:
optimizer="adam",
adam_beta1=0.9,
adam_beta2=0.999,
adam_eps=1e-8,
learning_rate=1e-4,
weight_decay=0.01,
gradient_clip_norm=1.0,
    ema_decay=0.9999,
    mixed_precision_training=bf16
```

C. Additional Experimental Settings

Compensating Global Similarity Using Iterative Closest Point. As we perform 3D reconstruction from single view images, global scale and rigid pose of the underlying objects cannot be resolved uniquely, introducing a global similarity ambiguity. Therefore, before applying geometric metrics such as Chamfer Distance and Volume IoU, we perform similarity alignment of our estimated shape with the ground-truth shape following standard practice of prior works (as listed in Table 1 from the main document). Specifically, We extended scale adaptive ICP [10] to identify optimal scale factors along each coordinate axes, in addition to its original uniform scale and translation.

D. Application: Refining Extracted Textured Mesh for Deformations

Here we show how our approach can be extended to obtain rigged and posed meshes. Our initial mesh is reconstructed by screened Poisson surface reconstruction [5], which typically consists of millions of uneven triangles with possible unnecessary outlier pieces. To improve the quality of the triangles and reduce their number for better rigging and posing, we perform additional refinement steps. First, we remove any small pieces that are disconnected from the main component. Next, we generate a cage mesh that encapsulates the original mesh, following the method described in [11]. We then perform Non-rigid ICP [1] to register the cage mesh to the original mesh. The registered cage mesh, now aligned with the original mesh, allows us to control the number and quality of the triangles, resulting in the final refined output mesh. In Fig. 2, we provide examples of refined meshes, including one that has been rigged and re-posed.

Method	CD↓	IoU↑	PSNR↑	SSIM↑	LPIPS↓
Ours	0.0135	0.7339	17.85	0.851	0.159
Ours-Persp.	0.0138	0.7272	17.70	0.848	0.159

Table 5. Comparison between our model and a variant that is trained with perspective images as the input.

E. Limitations

One limitation of our approach is the assumption that the input images are orthogonal, which may lead to distortion in the generated results, even though we do not see many visual artifacts when using perspective images as input in inference.

We tried training the model using perspective images with fixed focal length, and obtained results similar to but slightly worse than our main model trained based on orthogonal views (Tab. 5). Also note that the model trained using perspective images is still specific to the camera type. Therefore, developing a model that can handle images from various camera types remains an open and interesting research direction [6].

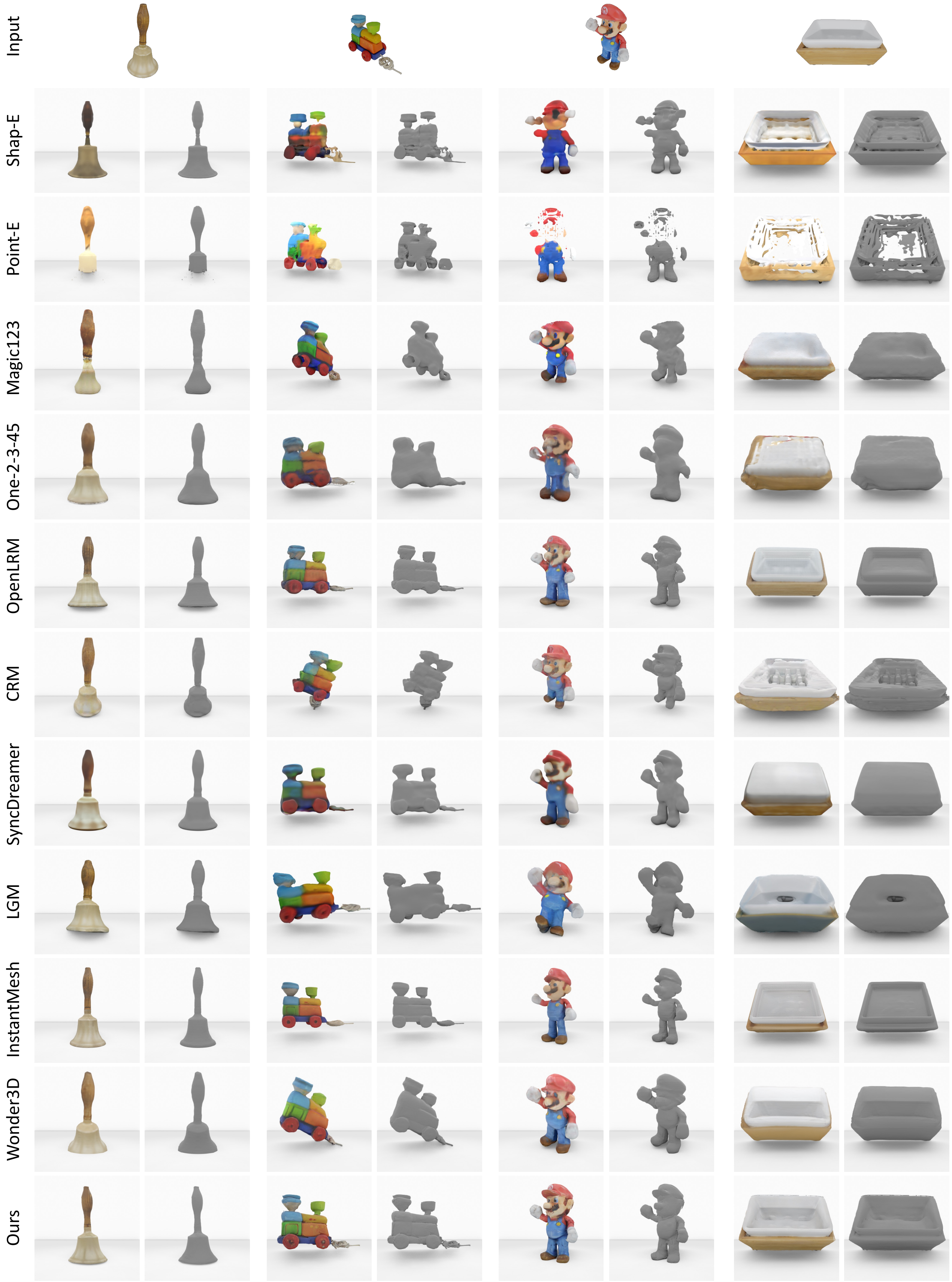

Figure 3. Additional comparisons on generated textured meshes.

References

[1] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007. 3

[2] 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

[3] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 1

[4] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 1

[5] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 1, 3

[6] Peng Li, Yuan Liu, Xiaoxiao Long, Feihu Zhang, Cheng Lin, Mengfei Li, Xingqun Qi, Shanghang Zhang, Wenhan Luo, Ping Tan, et al. Era3d: High-resolution multiview diffusion using efficient row-wise attention. *arXiv preprint arXiv:2405.11616*, 2024. 3

[7] 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. 1

[8] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 1

[9] 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

[10] Yusuf Sahillioğlu and Ladislav Kavan. Scale-adaptive icp. *Graphical Models*, 116:101113, 2021. 3

[11] Silvia Sellán, Jack Luong, Leticia Mattos Da Silva, Aravind Ramakrishnan, Yuchuan Yang, and Alec Jacobson. Breaking good: Fracture modes for realtime destruction. *ACM Transactions on Graphics*, 2022. 3