CRM: Single Image to 3D Textured Mesh with Convolutional Reconstruction Model

Zhengyi Wang[1,2], Yikai Wang[1,2], Yifei Chen[1], Chendong Xiang[1,2], Shuo Chen[1], Dajiang Yu[1], Chongxuan Li[3], Hang Su[1] and Jun Zhu[⋆1,2]

[1] Dept. of Comp. Sci. & Tech., BNRist Center, Tsinghua-Bosch Joint ML Center, Tsinghua University;
[2] ShengShu, Beijing, China;
[3] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China.

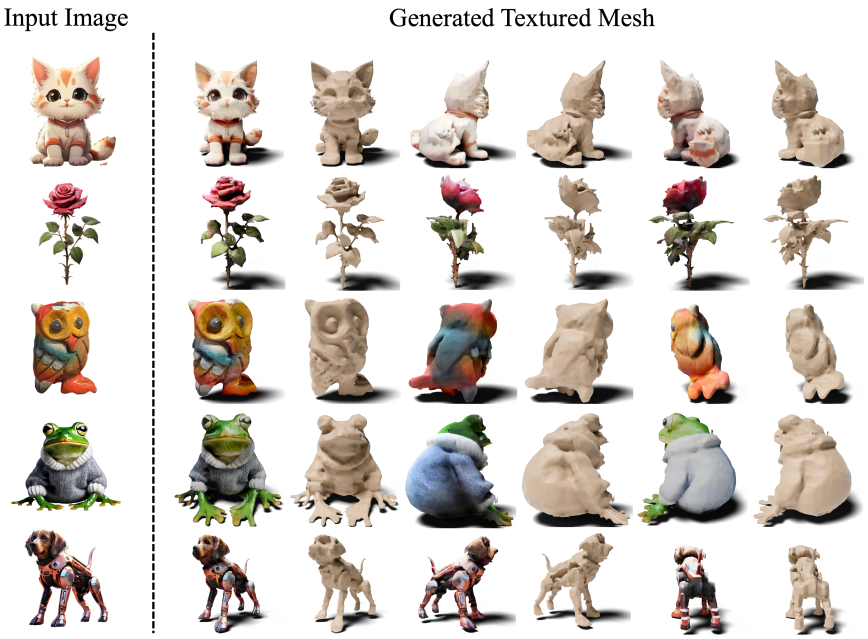

Fig. 1: CRM generates high-fidelity textured mesh from single image in **10 seconds**.

Abstract. Feed-forward 3D generative models like the Large Reconstruction Model (LRM) [18] have demonstrated exceptional generation speed. However, the transformer-based methods do not leverage the geometric priors of the triplane component in their architecture, often leading to sub-optimal quality given the limited size of 3D data and

⋆ The Corresponding author.

slow training. In this work, we present the Convolutional Reconstruction Model (CRM), a high-fidelity feed-forward single image-to-3D generative model. Recognizing the limitations posed by sparse 3D data, we highlight the necessity of integrating geometric priors into network design. CRM builds on the key observation that the visualization of triplane exhibits spatial correspondence of six orthographic images. First, it generates six orthographic view images from a single input image, then feeds these images into a convolutional U-Net, leveraging its strong pixel-level alignment capabilities and significant bandwidth to create a high-resolution triplane. CRM further employs Flexicubes as geometric representation, facilitating direct end-to-end optimization on textured meshes. Overall, our model delivers a high-fidelity textured mesh from an image in just 10 seconds, without any test-time optimization.

Keywords: 3D Generation · Textured Mesh · Diffusion Models

1 Introduction

In recent years, generative models have witnessed significant advancements, largely attributed to the fast growth in data size. Transformers [56], in particular, have achieved high-performance results across various domains including language [3], image [1, 38] and video generation [2]. However, the domain of 3D generation presents unique challenges. Unlike the abundance of other modal's data, 3D data is comparatively scarce. The creation of 3D data requires specialized expertise and considerable time, leading to a situation where the largest 3D datasets, namely Objaverse [12, 13], only contain millions of 3D content, much smaller than image datasets like Laion [46] which contains 5 billion images.

Despite this, recent developments have introduced some transformer-based methods [18, 21, 60, 64, 71] like LRM [18] for creating 3D content from single or multi-view images in a feed-forward manner. Among these models, the triplane has emerged as a popular component due to its efficiency in generating high-resolution 3D results with minimal memory cost. However, reliance on transformer-based networks for generating triplane patches has not utilized the geometric priors inherent to the triplane concept, leading to sub-optimal results in terms of quality and fidelity, and long training time.

To address the above challenges, this paper presents a new Convolutional Reconstruction Model (CRM) with high generation quality as well as fast training. Given the limited amount of 3D contents, CRM builds on a key hypothesis that it is beneficial to explore geometric priors in architecture design. Namely, we observe from the visualization of triplane [5, 6, 48] that triplane exhibits spatial correspondence of input six orthographic images, as shown in Fig. 2. The silhouette and texture of the input images have a natural alignment with the triplane structure. This motivates us to (1) use six orthographic images as input images to reconstruct 3D contents, which align well as the triplane feature, instead of other arbitrarily chosen poses; (2) use a U-Net convolutional network to map the input images to a rolled-out triplane by exploring the strong pixel-level alignment

between the input and output. Furthermore, the significant bandwidth capacity of our U-Net enables a direct transformation of the six orthographic images into the triplane, yielding highly detailed outcomes. Besides, we also add Canonical Coordinate Map (CCM) to the reconstruction network, a novel addition that enriches the model's understanding of spatial relations and geometry.

For the task of 3D generation from a single image, as the six orthographic images and CCMs are not directly available, we train a multi-view diffusion model conditioned on the input image to generate the six orthographic images and another diffusion model to generate the CCMs conditioned on the generated six orthographic images. Both diffusion models are trained on a filtered version of the Objaverse dataset [13]. To further enhance quality and robustness, we implement training improvements for the multi-view diffusion models, including Zero-SNR [26], random resizing, and contour augmentation.

Finally, as directly optimizing high-quality textured meshes is challenging, we adopt Flexicubes [47] as the geometry representation to falicitate gradient-based mesh optimization. This is unlike previous works [18,71] that use alternative 3D representation like NeRF [36] or Gaussian Splatting [19]. Such methods often involve extra procedure steps to obtain textured meshes [52], although they can produce detailed visualizations. With our designs, we are able to train CRM with textured mesh as the final output in an end-to-end manner, and our approach has a more straightforward inference pipeline and better mesh quality. Overall, our method can generate high-fidelity textured mesh within 10 seconds, as shown in Fig. 1.

2 Related Works

2.1 Score Distillation for 3D Generation

DreamFusion [39] proposes a technique called Score Distillation Sampling (SDS) (also known as Score Jacobian Chaining [57]). It utilizes large scale image diffusion models [44,45] to iteratively refine 3D models to align with specific prompts or images. Thus it can generate 3D content without training on 3D dataset. Along this line, ProlificDreamer [63] proposes Variational Score Distillation (VSD), a principled variational framework which greatly mitigates the over-saturation problems in SDS and improves the diversity. Zero123 [30], MVDream [49], ImageDream [59] and many others [22,41,42] further improve the results and mitigate the multi-face problems using diffusion models fine-tuned on 3D data. [33,40] explore amortized score distillation. Many other works [7–9,11,20,23–25,27,51,53,55,62,65,70] improve the results a lot, in either speed or quality. However, methods based on score distillation usually take from minutes to hours to generate single object, which is computationally expensive.

2.2 3D Generation with Sparse View Reconstruction

Several approaches aim to generate multi-view consistent images and then create 3D contents using sparse views reconstruction. For example, SyncDreamer [31]

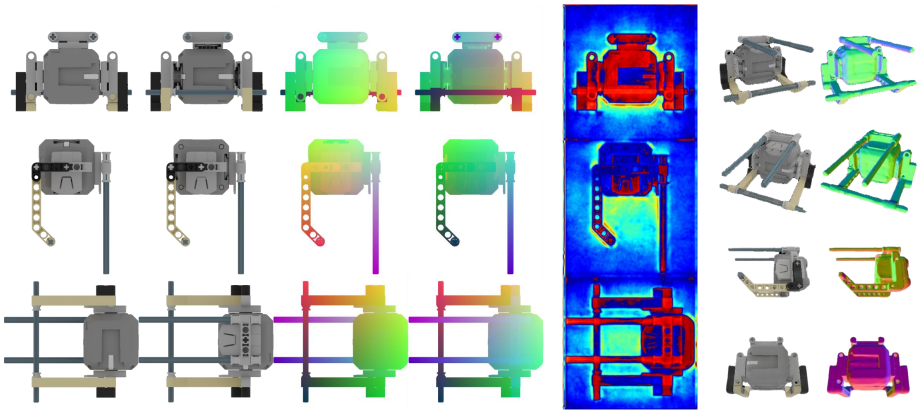

Fig. 2: One of our key motivation is that the triplane shares a strong spatial alignment with the input six orthographic images. (a) The six orthographic images of a input shape. (b) The six orthographic CCMs. (c) The triplane (mean value of all channels) output by our U-Net, which spatially aligns with the input images. (d) The textured mesh output by our convolutional reconstruction model.

generates multi-view consistent images and then uses NeuS [58] for reconstruction. Wonder3D [32] improves the results with cross-domain diffusion. Direct2.5 [34] improves the results with 2.5D diffusion. However, one common issue of these methods is that they need test-time optimization for reconstruction with sparse views, which may lead to extra computing and compromise the final quality

2.3 Feed-forward 3D Generative Models

Some works try to generate 3D objects using a feed forward model [4, 10, 15, 16, 66, 69]. Feed-forward methods demonstrate significantly faster generation speeds compared to the two types of methods mentioned above. Recently there are some works trained on larger 3D dataset Objaverse [13]. One-2-3-45 [29] generates multi-view images and then feed the images into a network to get the 3D object. LRM series [18, 21, 60, 64] improve the quality of generated results with a transformer-based architecture. TGS [71] and LGM [52] use Gaussian Splatting [19] as the geometry representation. There are also many other works [28, 35, 54, 68] that improve the results with different techniques. Despite these advancements, there remains room for improvement in the network architecture or geometry representation. Our approach utilizes a network with a strategically designed architecture and an end-to-end training approach producing meshes directly as the final output.

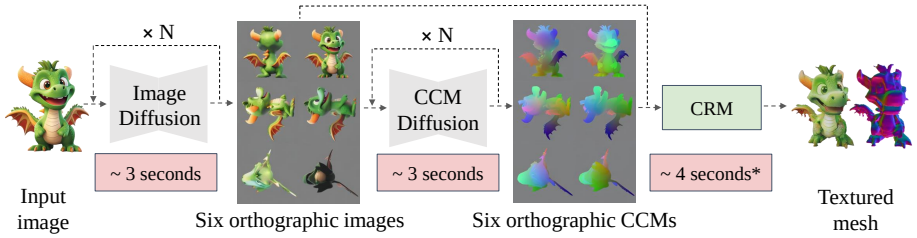

Fig. 3: Overall pipeline of our method. The input image is fed into a multi-view image diffusion model to generate six orthographic images. Then another diffusion model is used to generate the CCMs conditioned on the six images. The six images along with the CCMs are send into CRM to reconstruct the final textured mesh. The whole inference process takes around 10 seconds on an A800 GPU. *The 4 seconds includes the U-Net forward (less than 0.1s), querying surface points for UV texture and file I/O.

3 Method

In this section, we illustrate the detailed design of our method (shown in Fig. 3). Given a single input image, our model first utilizes multi-view diffusion models (Sec. 3.1) to generate six orthographic images and the canonical coordinates maps (CCMs). Then we develop our **convolutional reconstruction model** (CRM, Sec. 3.2) to reconstruct 3D textured mesh from the images and CCMs.

3.1 Multi-view Diffusion Model

We first explain the design of the multi-view diffusion model to generate six orthographic view images from a single input image. Instead of training from scratch, which is typically extremely expensive, we initialize the diffusion models using the checkpoint of ImageDream [59], a high-performance diffusion model for the task of multi-view images generation from a single image. The original ImageDream supports 4 views generation. We expand it to include 6 views by adding two more perspectives (up and down). We use another diffusion model which is conditioned on the generated six views to generate the canonical coordinate map. The conditional RGB image is concatenated with the noisy canonical coordinate map. It is also initialized from ImageDream checkpoint. Both the diffusion models are fine-tuned on the Objaverse [13] dataset.

To further improve the quality and robustness of our results, we introduce several enhancements: **(1) Zero-SNR Training.** We use the zero-SNR trick as mentioned in [26]. This can alleviate the problem resulting from the discrepancy between the initial Gaussian noise during sampling and the noisiest training sample. **(2) Random Resizing.** A naive implementation would make the model tends to generate objects that occupy the entire image. To mitigate this, we randomly resize the objects when training. **(3) Contour Augmentation.** We find that the model tends to predict the backview color largely relying on the

contour of the input view. To make the model insensitive to the contour, we randomly change the contour color during training.

3.2 Convolutional Reconstruction Model

We now move to introduce the detailed architecture of the convolutional reconstruction model (CRM). As outlined in Fig. 4, given the input six images and CCMs, a convolutional U-Net is used to map the input images along with the CCMs to a rolled-out triplane. Then the rolled-out triplane is reshaped into the triplane. Small multi-layer perceptions (MLPs) are used to decode the triplane features into SDF values, texture color and Flexicubes parameters. Lastly, these values are used to get texture mesh by dual marching cubes. Below, we explain the key components of CRM in detail.

Triplane Representation We choose triplane as the 3D representation, because it can achieve high resolution 3D results with 2D computation consumption. It projects each query grid cell to axis-aligned orthogonal planes (xy, xz, and yz planes) and then aggregates the feature from each planes. Then the feature is decoded by 3 tiny MLPs with 2 hidden layers to get the SDF values along with deformation, color and Flexicubes weights, respectively. Further, to avoid the unnecessary entanglements of different planes, we use rolled-out triplane [61].

Canonical Coordinates Map (CCM) We also add CCM as input [22], which contains extra geometry information. This is different from previous works, which typically use pure RGB images as the input to predict the 3D object [18]. Using pure RGB images make it extremely hard to predict the correct geometry, and sometime the geometry degrades (details in Sec. 4.3). Formally, CCM is the coordinates of each point in canonical space. It contains 3 channels whose values are within $[0, 1]$, representing the coordinates in the canonical space.

UNet-based Convolutional Network Our key insight is that the triplane is spatially aligned with the input six orthographic images and CCMs, as shown in Fig. 2. To match the rolled-out triplane, the six images and CCMs are arranged in a similar way. We render the six images and CCMs at a resolution of 256×256. They are split into two groups, with each group holding three images. These images in the four groups are then combined to create four larger images, each with a resolution of 256×768, allowing for spatial alignment. By concatenating these four groups, we form a 12-channel input. Next, a convolutional U-Net processes this input to produce the output triplane.

Compared to transformer-based methods [18, 21, 64, 71], our U-shape design has a larger bandwidth in preserving the input information, leading to highly detailed triplane features and finally elaborate textured meshes. Moreover, the convolutional network fully utilizes the geometry prior of the spatial correspondance of triplanes and input six orthograhic images, which greatly fasten the

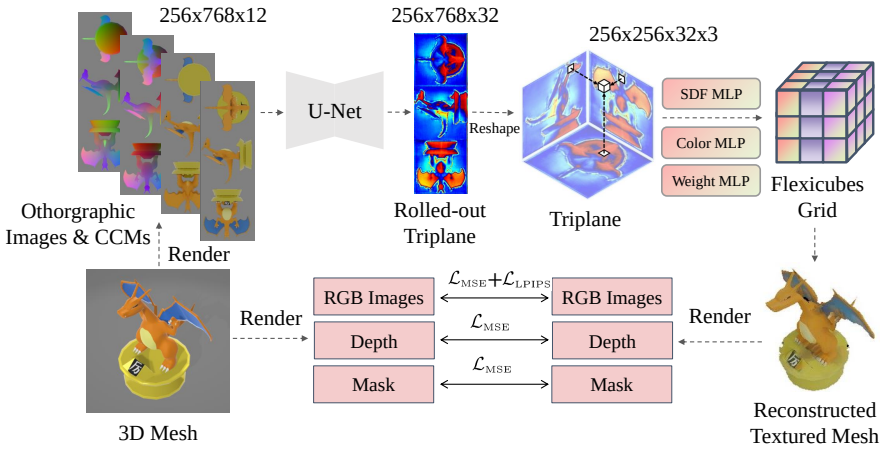

Fig. 4: Architecture along with training pipeline of CRM. We render the 3D mesh into six orthographic images and CCMs. Then the images and CCMs are concatenated and fed into the U-Net. The output triplane is decoded by small MLP networks to form the feature grid of Flexicubes, then textured mesh is get by dual marching cubes. During training, we render the color images, depth maps and masks from GT mesh and reconstructed mesh for supervision.

convergence and stabilize the training. Our model can get reasonable reconstruction results at very early stage of training (around 20 minutes of training from scratch). Also, our model can be trained with a much smaller batch size 32 (compared to transformer-based LRM that uses a batch size of 1024), which makes that all of our experiments can be conducted on an 8-GPU-cards machine. The overall training cost of our reconstruction model is only 1/8 than LRM. More details are shown in the experiments (see Sec. 4.1).

Flexicubes Geometry Previous generic 3D generation methods mostly adopt NeRF [36] or Gaussian splatting [19] as the geometry representation, which relies on extra procedures like Marching Cubes (MC) to extract the iso-surface, suffering from topological ambiguities and struggling to represent high-fidelity geometric details. In this work, we use Flexicubes [47] as our geometry representation. It can get meshes from the features on the grid by dual marching cubes [37] during training. The features include SDF values, deformation and weights. The texture is obtained by querying the color at the surface. Flexicubes enables us to train our reconstruction model with textured mesh as the final output in an end-to-end manner.

Loss Function Finally, to train our CRM model, we use a combination of MSE loss $\mathcal{L}_{\text{MSE}}$ and LPIPS loss [67] $\mathcal{L}_{\text{LPIPS}}$ for the rendered images for texture, similar as LRM [18]. To further enhance the geometry, we also include the depth map

Fig. 5: Qualitative comparison with baselines. Our models generates high-fidelity results with better geometry and texture.

and mask for supervision [16]. The overall loss function is

$$\begin{aligned}\mathcal{L} = &\mathcal{L}_{\text{MSE}}(\boldsymbol{x}, \boldsymbol{x}^{\text{GT}}) + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}}(\boldsymbol{x}, \boldsymbol{x}^{\text{GT}}) \\ &+ \lambda_{\text{depth}}\mathcal{L}_{\text{MSE}}(\boldsymbol{x}_{\text{depth}}, \boldsymbol{x}^{\text{GT}}_{\text{depth}}) + \lambda_{\text{mask}}\mathcal{L}_{\text{MSE}}(\boldsymbol{x}_{\text{mask}}, \boldsymbol{x}^{\text{GT}}_{\text{mask}}) + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}},\end{aligned} \tag{1}$$

where $\boldsymbol{x}$, $\boldsymbol{x}_{\text{depth}}$ and $\boldsymbol{x}_{\text{sil}}$ represent the RGB image, depth map and mask rendered from the reconstruction textured mesh, respectively. And $\boldsymbol{x}^{\text{GT}}$, $\boldsymbol{x}^{\text{GT}}_{\text{depth}}$ and $\boldsymbol{x}^{\text{GT}}_{\text{mask}}$ are rendered from the ground truth textured mesh. $\mathcal{L}_{\text{reg}}$ is the mesh-quality regularizers introduced in Flexicubes [47]. λ_{LPIPS}, λ_{depth}, λ_{mask} and λ_{reg} are the coefficients that balance each loss.

4 Experiments

4.1 Experimental Setting

Dataset We filter the Objaverse [13] dataset, removing scene-level objects and low quality meshes, and get around 376k valid high quality objects as the training set. We reuse the rendered images from SyncDreamer [31] which contain

16 images per shape at the resolution of 256×256, and additionally render 6 orthographic images and CCM with the same lighting and resolution.

Network Architecture The reconstruction model contains around 300M parameters. The U-Net contains $[64, 128, 128, 256, 256, 512, 512]$ channels, with attention blocks at resolution $[32, 16, 8]$. We set the Flexicubes grid size as 80.

Implementation Details The reconstruction model was trained on 8 NVIDIA A800 80GB GPU cards for 6 days with $110k$ iterations. The model was trained with batch size 32 (32 shapes per iteration). At each iteration, we randomly sampled 8 views among the total 16 images for each shape for supervision. We used the Adam optimizer with learning rate $1e{-}4$. The coefficients that balancing each loss were set as $\lambda_{\text{LPIPS}} = 0.1$, $\lambda_{\text{depth}} = 0.5$, $\lambda_{\text{mask}} = 0.5$ and $\lambda_{\text{reg}} = 0.005$. To enhance robustness against minor inconsistency in the generated multi-view images, we introduced small Gaussian noise to the inputs in both training and inference.

The diffusion models for both six orthographic images and CCMs were trained on 8 NVIDIA A800 80GB GPU cards for 2 days with $10k$ iterations. The gradient accumulation is set as 12 steps, yielding a total batch size of 1536. We used the Adam optimizer with learning rate $5e-5$. During sampling both diffusions were sampled with 50 steps using DDIM [50].

4.2 Comparison with baselines

Qualitative Results To validate the effectiveness of our method, we qualitatively compare our results with previous works including Wonder3d [32], SyncDreamer [31], Magic123 [41], One-2-3-45 [29] and OpenLRM [17]. Since LRM [18] is not open-sourced, we use OpenLRM [17], an open-sourced implementation of LRM for comparisons. For the other baselines, we use their official codes and checkpoints. As for the input images for testing, we choose two from GSO [14] dataset, one downloaded from web and one generated by text-to-image diffusion model. The results are shown in Fig. 5. It can be seen from the figure that our method generates 3D textured meshes with better texture and geometry than all other baselines. This is because our reconstruction model fully utilizes the spatial alignment of input six orthographic images and output triplane. Also, our model can generate with only 10 seconds, much faster than most of the baselines. Our method is trained in an end-to-end manner with textured mesh as final output, thus avoiding a time-consuming post-processing for converting to mesh as in [52].

Additionally, we visualize the generated meshes comparing to the previous work LRM [18] and a concurrent work LGM [52]. Since LRM is not open-sourced, we use the meshes from their project page. The results are shown in Fig. 6. It can be seen from the figure that our results have better texture. Our method also has smoother geometry than LRM and better geometry details than LGM.

In Fig. 7, we show more results of the high-fidelity textured meshes generated from single image by our method.

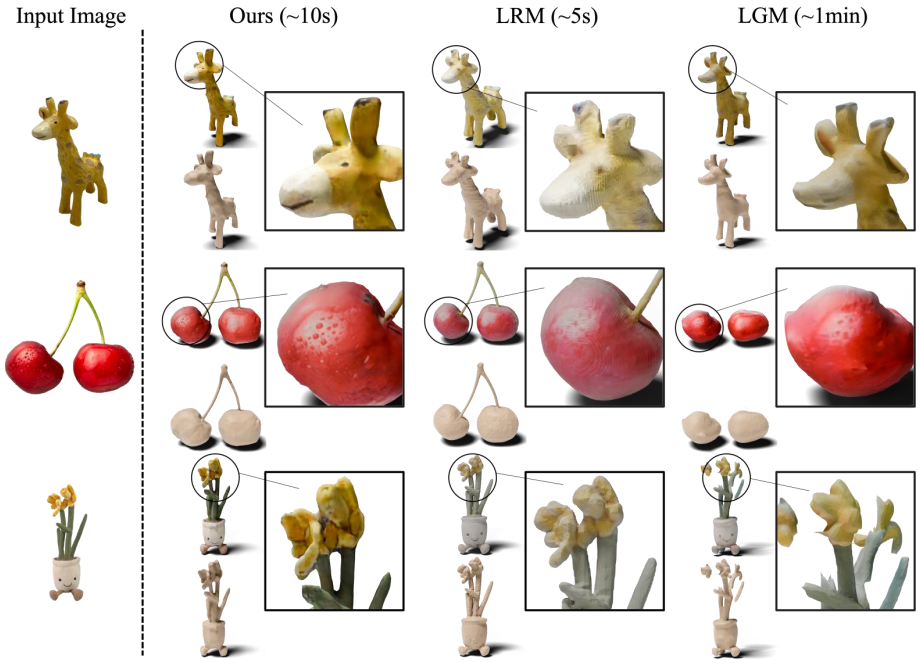

Fig. 6: Qualitative comparison with LRM [18] and LGM [52]. Our models generates high-fidelity results with detailed texture and smooth geometry.

Quantitative Results In line with previous studies [32], we evaluate our method using the Google Scanned Objects (GSO) dataset [14] which is not included in our training dataset. We randomly choose 30 shapes and render a single image with size of 256×256 as input for evaluation. To ensure the generated mesh accurately aligned with the ground truth mesh, we carefully adjust their pose and scale them to fit within the $[-0.5, 0.5]$ box. For mesh geometry evaluation, we report Chamfer Distance (CD), Volumn IoU and F-Score (with a threshold of 0.05, following One-2-3-45 [29]), which measure the geometry similarity between the reconstructed mesh and ground truth mesh. The results are shown in Table 1. It can be seen from the table that our method outperforms all of the baselines, which demonstrates the effectiveness of our method for geometry quality.

Furthermore, for evaluating mesh texture, we render 24 images at 512×512 resolution at elevation angles of 0, 15, and 30 degrees, for the generated meshes and ground-truth meshes respectively. For each elevation, the 8 images are evenly distributed around a full 360-degree rotation. Then we assessed them using several metrics: PSNR, SSIM, LPIPS and Clip-Similarity, which measure the resemblance in appearance between the reconstructed mesh and the original ground truth mesh. The results are shown in Table 2. It shows that the generated

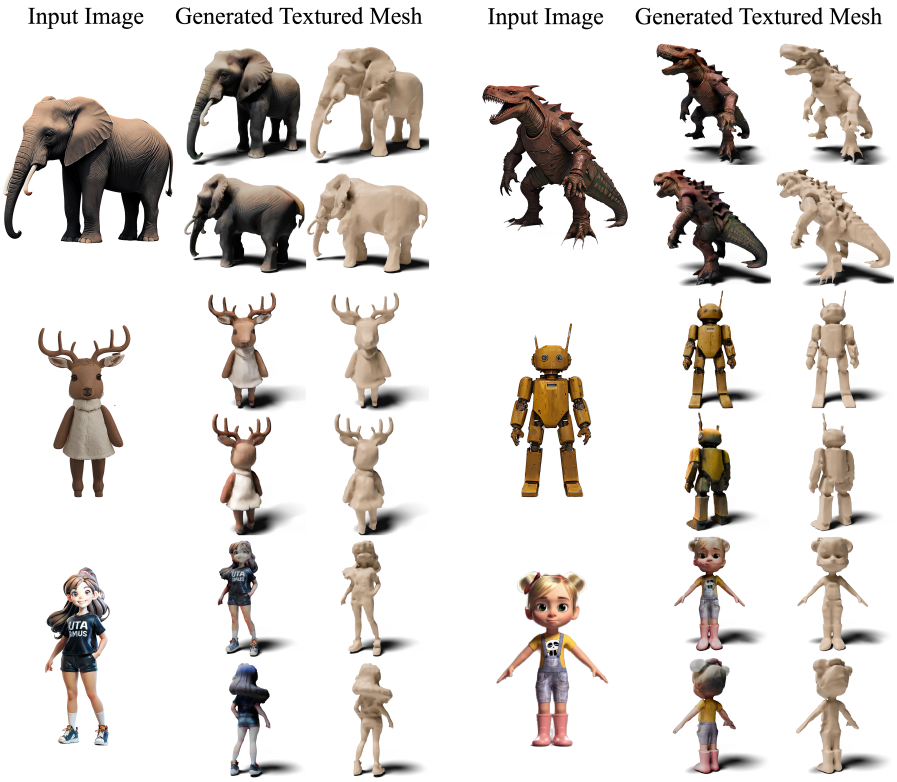

Fig. 7: More Results of the generated results of our method from a single image.

textured meshes by our model surpass those of all baselines in appearance, which demonstrates the effectiveness of our method for texture quality.

Additionally, we carry out experiments to evaluate the effectiveness of our single image to multi-view diffusion models. We use PSNR, SSIM, and LPIPS to measure the similarity between the generated multi-view images and ground truth multi-view images. For this analysis, we use four views of the generated images (left, right, front, back) from each model under comparison, setting the background color to grey (value 128). We compare with SyncDreamer [31] and Wonder3D [32]. The outcomes of this evaluation are documented in Table 3. It can be seen from the table that our method outperforms all the baselines.

4.3 Ablation Study and Analysis

Reconstruction Results on Early Training Stage An advantage of our CRM is that it is easy to train. In fact, we find that CRM starts to show reasonable results at very early stage of training. The results are shown in Fig. 8. The results are good even with barely 280 iterations (only 20 minutes of training).

Table 1: Quantitative comparison for the geometry quality between our method and baselines for single image to 3D textured mesh generation. We report the metrics of Chamfer Distance, Volumn IoU and F-score on GSO dataset.

Method	Chamfer Dist.↓	Vol. IoU↑	F-Sco. (%)↑
One-2-3-45 [29]	0.0172	0.4463	72.19
SyncDreamer [31]	0.0140	0.3900	75.74
Wonder3D [32]	0.0186	0.4398	76.75
Magic123 [41]	0.0188	0.3714	60.66
TGS [71]	0.0172	0.2982	65.17
OpenLRM [17, 18]	0.0168	0.3774	63.22
LGM [52]	0.0117	0.4685	68.69
Ours	**0.0094**	**0.6131**	**79.38**

Table 2: Quantitative comparison for the texture quality between our method and baselines for single image to 3D textured mesh generation. We report the metric of PSNR, SSIM, LPIPS and Clip [43]-Similarity on GSO dataset.

Method	PSNR↑	SSIM↑	LPIPS↓	Clip-Sim↑
One-2-3-45 [29]	13.93	0.8084	0.2625	79.83
SyncDreamer [31]	14.00	0.8165	0.2591	82.76
Wonder3D [32]	13.31	0.8121	0.2554	83.70
Magic123 [41]	12.69	0.7984	0.2442	85.16
OpenLRM [17, 18]	14.30	0.8294	0.2276	84.20
LGM [71]	13.28	0.7946	0.2560	85.20
Ours	**16.22**	**0.8381**	**0.2143**	**87.55**

We conjecture that the fast convergence results from the strong geometry prior in our architecture design.

Training Time of CRM In Fig. 9 we compare the training cost between our method (reconstruction model only) and two baselines, LRM [18] and LGM [52]. We measure the training cost by the training days multiplying the amount of used NVIDIA A100/A800 GPU cards. It can be seen that our model takes much smaller training time than the two baselines. This is because our model utilizes the spatial correspondence between input six orthographic images/CCMs and triplanes, which serves as a strong prior that makes the training easier.

Importance of Input CCM We examine the importance of the CCMs that are concatenated to the input images. To compare, we train a reconstruction model that takes only the six RGB images as input, without CCMs. The results are shown in Fig. 10. It can be seen that the results of the geometry degrade a lot without CCM input. This is because CCM provides important geometry information for the model, especially when the geometry is complex.

Table 3: Quantitative comparison between our method and baselines for novel view synthesis of the multi-view diffusion model. We report the metric of PSNR, SSIM and LPIPS on GSO dataset.

Method	PSNR↑	SSIM↑	LPIPS↓
SyncDreamer [31]	20.30	0.7804	0.2932
Wonder3D [32]	23.76	0.8127	0.2210
Ours	**29.36**	**0.8721**	**0.1354**

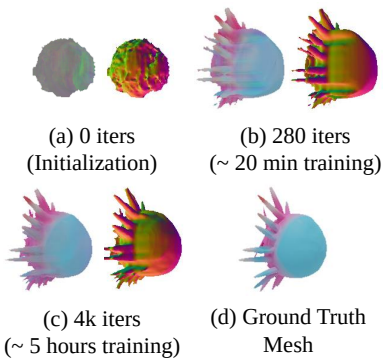

Fig. 8: Reconstruction results on unseen samples during early stage of training.

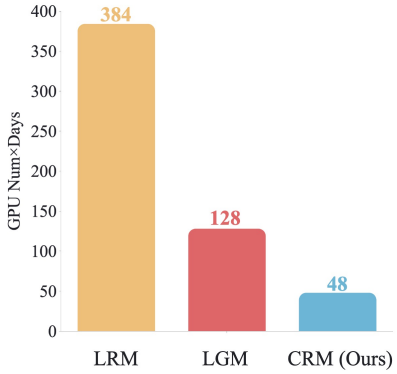

Fig. 9: Training cost comparison. Our model require much less computation cost than other baselines.

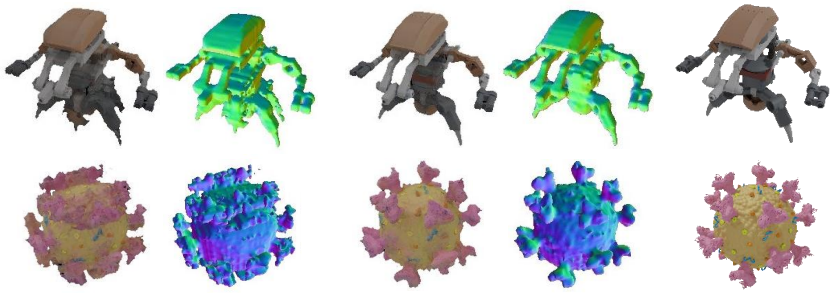

Fig. 10: The CCM concatenated to the input images is beneficial for our model. (a) Without providing CCM, the model outputs a geometry which is reasonable, but not very good. (b) The shape reconstructed using our full model with CCM input, with a much better geometry. (c) Ground truth mesh rendered from the same pose.

Design of Multi-view Diffusion Here we examine the effectiveness of the design of the multi-view diffusion models. Starting from the baseline that naively

Table 4: Ablation study on the design of multi-view diffusion on novel view synthesis. We report the metrics of PSNR, SSIM and LPIPS on GSO dataset.

Method	PSNR ↑	SSIM ↑	LPIPS↓
ImageDream (6 view)	28.99	0.8565	0.1497
+ Zero-SNR	29.13	0.8598	0.1498
+ Random Resizing	**29.36**	**0.8721**	**0.1354**
+ Contour Augmentation	28.92	0.8681	0.1444

(a) Input Image (b) Segmentation Result (c) w/o Contour Augmentation (d) w/ Contour Augmentation (e) Contour Augmentated Image

Fig. 11: Demonstration of contour augmentation. (a) Given an input image, (b) off-the-shelf segmentation model sometimes provides imperfect results. (c) Without contour augmentation, the predicted backview color is sensitive to the contour. (d) With contour augmentation, the model predicts reasonable result. (e) We demonstrate how we augment the input image during training.

fine-tunes the pre-trained ImageDream model with 2 additional views, we sequentially add the the proposed techniques on the training. We examine the results on a subset of GSO, comparing the similarity of the generated novel view images with the ground truth images using PSNR, SSIM and LPIPS metrics. The results are shown in table 4. It can be seen that both the Zero-SNR trick and random resizing are beneficial. Note that the contour augmentation does not improve the quantitative metrics. However, we find that this trick makes the model more robust to in the wild input images (Fig. 11).

5 Conclusion

In this work, we present a convolutional reconstruction model (CRM) for creating high-quality 3D models from a single image. Our approach effectively utilizes the spatial relationship between input images and the output triplane, leading to improved textured meshes, with significantly less training cost compared to previous transformer-based methods [18]. The model operates on an end-to-end training basis, directly outputting textured meshes. Overall, our method can produce detailed textured meshes in just 10 seconds.

Potential Negative Impact. Similar to many other generated models, our CRM may be used to generate malicious or fake 3D contents, which may need additional caution.

Acknowledgements

This work was supported by Natural Science Foundation of China (No.s 62350080, 62076147), Tsinghua Institute for Guo Qiang, and the High Performance Computing Center, Tsinghua University. J.Z is also supported by the New Cornerstone Science Foundation through the XPlorer Prize.

References

1. Bao, F., Nie, S., Xue, K., Cao, Y., Li, C., Su, H., Zhu, J.: All are worth words: A vit backbone for diffusion models. In: CVPR (2023)
2. Brooks, T., Peebles, B., Homes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), `https://openai.com/research/video-generation-models-as-world-simulators`
3. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020)
4. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16123–16133 (2022)
5. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision. pp. 333–350. Springer (2022)
6. Chen, H., Gu, J., Chen, A., Tian, W., Tu, Z., Liu, L., Su, H.: Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. arXiv preprint arXiv:2304.06714 (2023)
7. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. arXiv preprint arXiv:2303.13873 (2023)
8. Chen, Y., Zhang, C., Yang, X., Cai, Z., Yu, G., Yang, L., Lin, G.: It3d: Improved text-to-3d generation with explicit view synthesis (2023)
9. Chen, Z., Wang, F., Liu, H.: Text-to-3d using gaussian splatting. arXiv preprint arXiv:2309.16585 (2023)
10. Cheng, Y.C., Lee, H.Y., Tuyakov, S., Schwing, A., Gui, L.: SDFusion: Multimodal 3d shape completion, reconstruction, and generation. In: CVPR (2023)
11. Decatur, D., Lang, I., Aberman, K., Hanocka, R.: 3d paintbrush: Local stylization of 3d shapes with cascaded score distillation. arXiv preprint arXiv:2311.09571 (2023)
12. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. Advances in Neural Information Processing Systems **36** (2024)
13. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13142–13153 (2023)

14. Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T.B., Vanhoucke, V.: Google scanned objects: A high-quality dataset of 3d scanned household items. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 2553–2560. IEEE (2022)
15. Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., Fidler, S.: Get3d: A generative model of high quality 3d textured shapes learned from images. Advances In Neural Information Processing Systems **35**, 31841–31854 (2022)
16. Gupta, A., Xiong, W., Nie, Y., Jones, I., Oğuz, B.: 3dgen: Triplane latent diffusion for textured mesh generation. arXiv preprint arXiv:2303.05371 (2023)
17. He, Z., Wang, T.: Openlrm: Open-source large reconstruction models. `https://github.com/3DTopia/OpenLRM` (2023)
18. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. arXiv preprint arXiv:2311.04400 (2023)
19. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)
20. Kim, S., Lee, K., Choi, J.S., Jeong, J., Sohn, K., Shin, J.: Collaborative score distillation for consistent visual synthesis (2023)
21. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. arXiv preprint arXiv:2311.06214 (2023)
22. Li, W., Chen, R., Chen, X., Tan, P.: Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. arXiv preprint arXiv:2310.02596 (2023)
23. Li, Y., Dou, Y., Shi, Y., Lei, Y., Chen, X., Zhang, Y., Zhou, P., Ni, B.: Focaldreamer: Text-driven 3d editing via focal-fusion assembly (2023)
24. Liang, Y., Yang, X., Lin, J., Li, H., Xu, X., Chen, Y.: Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching (2023)
25. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 300–309 (2023)
26. Lin, S., Liu, B., Li, J., Yang, X.: Common diffusion noise schedules and sample steps are flawed. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5404–5411 (2024)
27. Liu, F., Wu, D., Wei, Y., Rao, Y., Duan, Y.: Sherpa3d: Boosting high-fidelity text-to-3d generation via coarse 3d prior (2023)
28. Liu, M., Shi, R., Chen, L., Zhang, Z., Xu, C., Wei, X., Chen, H., Zeng, C., Gu, J., Su, H.: One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. arXiv preprint arXiv:2311.07885 (2023)
29. Liu, M., Xu, C., Jin, H., Chen, L., Xu, Z., Su, H., et al.: One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. arXiv preprint arXiv:2306.16928 (2023)
30. Liu, R., Wu, R., Hoorick, B.V., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object (2023)
31. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: Syncdreamer: Generating multiview-consistent images from a single-view image. arXiv preprint arXiv:2309.03453 (2023)
32. Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., et al.: Wonder3d: Single image to 3d using cross-domain diffusion. arXiv preprint arXiv:2310.15008 (2023)

33. Lorraine, J., Xie, K., Zeng, X., Lin, C.H., Takikawa, T., Sharp, N., Lin, T.Y., Liu, M.Y., Fidler, S., Lucas, J.: Att3d: Amortized text-to-3d object synthesis (2023)
34. Lu, Y., Zhang, J., Li, S., Fang, T., McKinnon, D., Tsin, Y., Quan, L., Cao, X., Yao, Y.: Direct2.5: Diverse text-to-3d generation via multi-view 2.5d diffusion (2023)
35. Mercier, A., Nakhli, R., Reddy, M., Yasarla, R., Cai, H., Porikli, F., Berger, G.: Hexagen3d: Stablediffusion is just one step away from fast and diverse text-to-3d generation. arXiv preprint arXiv:2401.07727 (2024)
36. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
37. Nielson, G.M.: Dual marching cubes. In: IEEE visualization 2004. pp. 489–496. IEEE (2004)
38. Peebles, W., Xie, S.: Scalable diffusion models with transformers (2023)
39. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
40. Qian, G., Cao, J., Siarohin, A., Kant, Y., Wang, C., Vasilkovsky, M., Lee, H.Y., Fang, Y., Skorokhodov, I., Zhuang, P., Gilitschenski, I., Ren, J., Ghanem, B., Aberman, K., Tulyakov, S.: Atom: Amortized text-to-mesh using 2d diffusion (2024)
41. Qian, G., Mai, J., Hamdi, A., Ren, J., Siarohin, A., Li, B., Lee, H.Y., Skorokhodov, I., Wonka, P., Tulyakov, S., Ghanem, B.: Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors (2023)
42. Qiu, L., Chen, G., Gu, X., Zuo, Q., Xu, M., Wu, Y., Yuan, W., Dong, Z., Bo, L., Han, X.: Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d (2023)
43. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
44. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
45. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022)
46. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: Laion-5b: An open large-scale dataset for training next generation image-text models (2022)
47. Shen, T., Munkberg, J., Hasselgren, J., Yin, K., Wang, Z., Chen, W., Gojcic, Z., Fidler, S., Sharp, N., Gao, J.: Flexible isosurface extraction for gradient-based mesh optimization. ACM Transactions on Graphics (TOG) **42**(4), 1–16 (2023)
48. Shi, R., Wei, X., Wang, C., Su, H.: Zerorf: Fast sparse view 360 {\deg} reconstruction with zero pretraining. arXiv preprint arXiv:2312.09249 (2023)
49. Shi, Y., Wang, P., Ye, J., Long, M., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. arXiv preprint arXiv:2308.16512 (2023)
50. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
51. Sun, J., Zhang, B., Shao, R., Wang, L., Liu, W., Xie, Z., Liu, Y.: Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior (2023)

52. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation (2024)
53. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023)
54. Tochilkin, D., Pankratz, D., Liu, Z., Huang, Z., , Letts, A., Li, Y., Liang, D., Laforte, C., Jampani, V., Cao, Y.P.: Triposr: Fast 3d object reconstruction from a single image. arXiv preprint arXiv:2403.02151 (2024)
55. Tsalicoglou, C., Manhardt, F., Tonioni, A., Niemeyer, M., Tombari, F.: Textmesh: Generation of realistic 3d meshes from text prompts. arXiv preprint arXiv:2304.12439 (2023)
56. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2023)
57. Wang, H., Du, X., Li, J., Yeh, R.A., Shakhnarovich, G.: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation (2022)
58. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction (2023)
59. Wang, P., Shi, Y.: Imagedream: Image-prompt multi-view diffusion for 3d generation. arXiv preprint arXiv:2312.02201 (2023)
60. Wang, P., Tan, H., Bi, S., Xu, Y., Luan, F., Sunkavalli, K., Wang, W., Xu, Z., Zhang, K.: Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. arXiv preprint arXiv:2311.12024 (2023)
61. Wang, T., Zhang, B., Zhang, T., Gu, S., Bao, J., Baltrusaitis, T., Shen, J., Chen, D., Wen, F., Chen, Q., et al.: Rodin: A generative model for sculpting 3d digital avatars using diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4563–4573 (2023)
62. Wang, X., Wang, Y., Ye, J., Wang, Z., Sun, F., Liu, P., Wang, L., Sun, K., Wang, X., He, B.: Animatabledreamer: Text-guided non-rigid 3d model generation and reconstruction with canonical score distillation
63. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In: Thirty-seventh Conference on Neural Information Processing Systems (2023), https://openreview.net/forum?id=ppJuFSOAnM
64. Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzstein, G., Xu, Z., et al.: Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. arXiv preprint arXiv:2311.09217 (2023)
65. Yu, X., Guo, Y.C., Li, Y., Liang, D., Zhang, S.H., Qi, X.: Text-to-3d with classifier score distillation (2023)
66. Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K.: Lion: Latent point diffusion models for 3d shape generation (2022)
67. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric (2018)
68. Zheng, X.Y., Pan, H., Guo, Y.X., Tong, X., Liu, Y.: Mvd^2: Efficient multiview 3d reconstruction for multiview diffusion (2024)
69. Zheng, X.Y., Pan, H., Wang, P.S., Tong, X., Liu, Y., Shum, H.Y.: Locally attentional sdf diffusion for controllable 3d shape generation. arXiv preprint arXiv:2305.04461 (2023)
70. Zhu, J., Zhuang, P.: Hifa: High-fidelity text-to-3d generation with advanced diffusion guidance (2023)
71. Zou, Z.X., Yu, Z., Guo, Y.C., Li, Y., Liang, D., Cao, Y.P., Zhang, S.H.: Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. arXiv preprint arXiv:2312.09147 (2023)