Freeplane: Unlocking Free Lunch in Triplane-Based Sparse-View Reconstruction Models

Wenqiang Sun[1,3], **Zhengyi Wang**[2,3], **Shuo Chen**[2], **Yikai Wang**[2], **Zilong Chen**[2,3], **Jun Zhu**[†2,3], **Jun Zhang**[†1]
[1]Department of Electronic and Computer Engineering, HKUST
[2]Department of Computer Science and Technology, Tsinghua University [3]ShengShu
wsunap@connect.ust.hk;
{wang-zy21, chenshuo20, chenz122}@mails.tsinghua.edu.cn;
yikaiw@outlook.com; dcszj@tsinghua.edu.cn; eejzhang@ust.hk;

Abstract

Creating 3D assets from single-view images is a complex task that demands a deep understanding of the world. Recently, feed-forward 3D generative models have made significant progress by training large reconstruction models on extensive 3D datasets, with triplanes being the preferred 3D geometry representation. However, effectively utilizing the geometric priors of triplanes, while minimizing artifacts caused by generated inconsistent multi-view images, remains a challenge. In this work, we present **Fre**quency modulat**e**d tri**plane** (**Freeplane**), a simple yet effective method to improve the generation quality of feed-forward models without additional training. We first analyze the role of triplanes in feed-forward methods and find that the inconsistent multi-view images introduce high-frequency artifacts on triplanes, leading to low-quality 3D meshes. Based on this observation, we propose strategically filtering triplane features and combining triplanes before and after filtering to produce high-quality textured meshes. These techniques incur no additional cost and can be seamlessly integrated into pre-trained feed-forward models to enhance their robustness against the inconsistency of generated multi-view images. Both qualitative and quantitative results demonstrate that our method improves the performance of feed-forward models by simply modulating triplanes. All you need is to modulate the triplanes during inference.

1 Introduction

Recently, generative models have showcased substantial progress in generating realistic images and videos through the integration of extensive datasets and efficient architectures. When it comes to 3D generation, however, the limited datasets constrain generative models from producing high-quality 3D assets that can be used for industrial manufacturing and artistic creation. To resolve this issue, DreamFusion and subsequent works [33, 17, 51] have tried to distill 2D image prior from Stable Diffusion [36] into 3D scene representation using the Score Distillation Sampling (SDS) technique. Nevertheless, the SDS-based per-scene optimization encounters challenges of 3D inconsistency and time-consuming computations, making it impractical for real-world scenarios.

To achieve fast and generalizable 3D generative models, early studies [8, 2] develop the triplane-based 3D GAN architecture, benefiting from the computation-efficient and expressive nature of triplanes. Despite generating high-quality 3D assets, the generalization capacity of these models is constrained by their unstable training process and non-scalable architectures. Inspired by the achievements of the

[†]Corresponding author

Preprint. Under review.

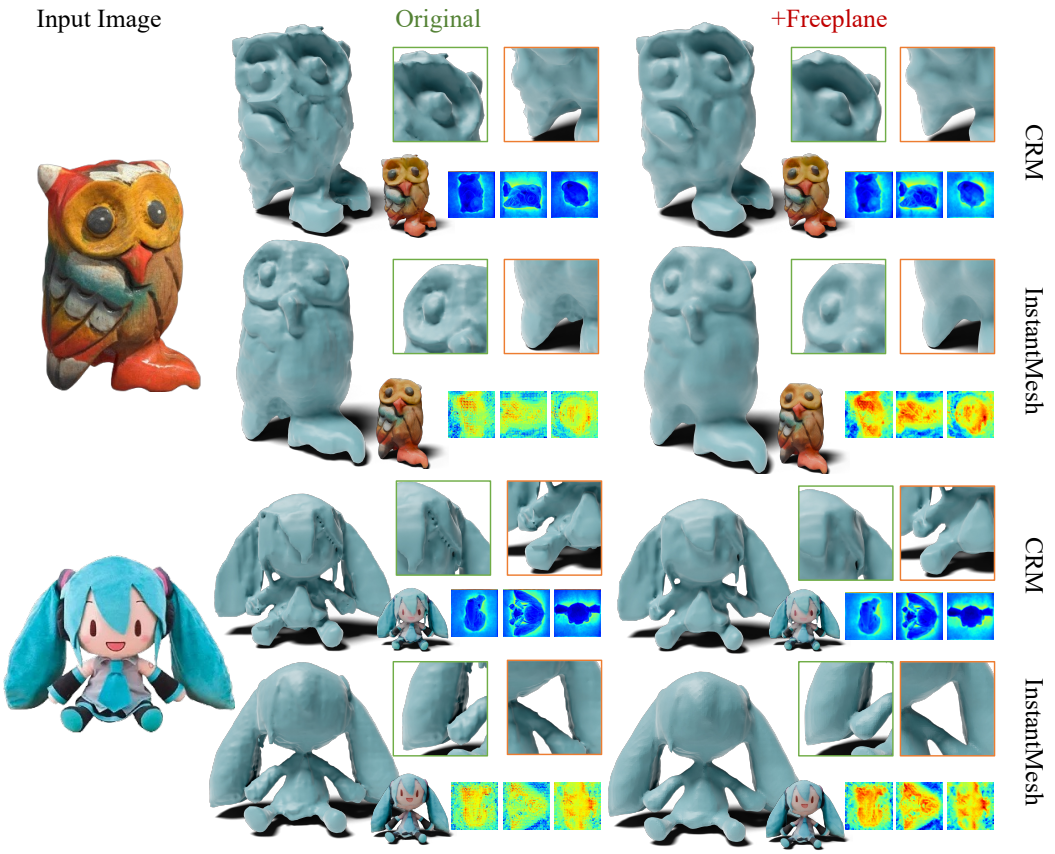


Figure 1: We present **Freeplane**, a method that substantially refines the mesh quality of feed-forward generative models without additional costs: no training or fine-tuning, no extra memory required, and only a few lines of code.

transformer architecture [46, 31] in natural language processing, LRM [11] introduces a transformer-based reconstruction model to reconstruct high-quality 3D meshes from a single image. Equipped with the scalable architecture and large-scale 3D datasets, LRM has become a promising backbone for 3D generation. Built upon LRM, Instant3D [14] and InstantMesh [58] introduce a multi-view diffusion model to achieve a fast single image to 3D generation process. To fully utilize the 3D geometry prior in triplanes, CRM [52] suggests generating highly detailed triplanes from multi-view images using a UNet-based convolutional network. Leveraging high-resolution triplanes as strong priors, CRM generates textured meshes with detailed geometry. Nevertheless, CRM fails to achieve satisfactory 3D generation because high-resolution triplanes are highly sensitive to inconsistencies in generated multi-view images.

These feed-forward methods, consisting of a multi-view diffusion model and a sparse-view reconstruction model, encounter a common challenge: the quality of generated meshes is inevitably influenced by the 3D inconsistency of multi-view images. Existing works mainly focus on training more powerful multi-view diffusion models and scalable reconstruction models, while the internal properties of triplanes remain largely under-explored. In light of this, we raise the question for 3D generation: **is it possible to alleviate the artifacts caused by inconsistent multi-view images directly within the triplanes?**

In this work, we propose a simple yet effective method that surprisingly addresses the aforementioned challenge with just a few additional lines of code. Specifically, we examine the role of triplanes in feed-forward models and find that inconsistent multi-view images inherently introduce high-frequency artifacts into the triplanes, resulting in rough and uneven meshes. Drawing on our analysis, we propose two strategies to improve the 3D generation quality without additional cost. One is frequency modulation, applied to the triplanes, which reduces the negative effects caused by inconsistent multi-

view images. The other is the combination of triplanes before and after filtering to produce detailed textured meshes, which utilizes the former to predict the SDF values and the latter to acquire the texture. We refer to our approach as **Fre**quency modulat**e**d tri**plane** (**Freeplane**), which is free of additional training and computational overhead.

Our Freeplane framework can be seamlessly integrated with existing feed-forward methods, improving the generation quality within a few lines of code. We conduct a comprehensive evaluation of our approach, employing CRM [52] and InstantMesh [58] (two state-of-the-art open-sourced feed-forward generative models) as the base models. By incorporating Freeplane during the testing phase, these models show a noticeable enhancement in the smoothness and geometric details of generated meshes. The visualization results depicted in Fig. 1 confirm the effectiveness of Freeplane in reducing the artifacts arising from inconsistent multi-view images. Our contributions are summarized as follows:

- We reveal the role of triplanes in feed-forward models and uncover the performance degradation caused by inconsistent multi-view images, which is further verified by empirical studies and addressed by our approach.
- We introduce a simple yet effective method, denoted as **Freeplane**, which fully utilizes the geometry priors in triplanes while alleviating the artifacts caused by inconsistent multi-view images. It broadly improves the smoothness and geometry details of generated meshes without requiring additional training.
- The proposed approach can be seamlessly integrated into existing triplane-based sparse-view reconstruction models. Both qualitative and quantitative experimental results demonstrate that our approach exhibits apparent generation quality enhancement across different methods, showing the robustness of Freeplane.

2 Related Work

2.1 Neural Fields

The neural radiance field (NeRF) [27] is a popular representation in the 3D vision area, which uses deep neural networks to represent 3D scenes as continuous functions. Following works [1, 42, 30, 32] extend the neural fields to large-scale scenes and sparse-input reconstruction tasks. However, it takes a long time to train these NeRF-based models, usually from hours to days. To accelerate the training process while keeping a low memory cost, TensoRF [3] proposes to formulate the radiance field as a 4D tensor and factorize the 4D tensor into low-rank components. In light of this, LRM chooses the triplanes as the concise and scalable 3D representation to build a generalized reconstruction model.

Although NeRF can be transformed into meshes using Marching Cubes [23], the quality of extracted meshes is not assured. Previous methods aim to convert the neural fields into the implicit surface representation, such as the Signed Distance Function (SDF) [60, 48, 16], to produce smooth and detailed meshes. Moreover, some proposed differentiable marching cubes (DiffMC) techniques [35, 53] improve the mesh reconstruction quality and speed. Recently, to present the high-fidelity geometric details, Flexicube [39] proposes to reconstruct the mesh from features of grids by dual marching cube [38]. The grid features include the SDF values, weights and deformation. In addition, the textures are queried from the surface. To produce high-resolution textured meshes, recent feed-forward models [52, 58] adopt Flexicubes as the geometry representation.

2.2 3D Generation

Generative models, such as Generative Adversarial Networks (GANs) [9] and Diffusion Models [36], have made remarkable progress in creating realistic and diverse images and videos. In the context of 3D generation, some early attempts directly utilize 3D assets or multi-view datasets to train 3D generative models with GANs [56, 2, 8] and diffusion models [25, 29, 12, 10, 15, 28]. Despite achieving relatively satisfactory 3D shape generation, their progress is hindered by the scale, quality, and diversity of 3D datasets. DreamFusion [33] proposes leveraging the pre-trained image diffusion models [36, 37] to optimize the 3D models with a technique called Score Distillation Sampling (SDS). Follow-up approaches aim to achieve faster optimization or more high-quality generation [17, 47, 4, 51, 44, 5, 63]. However, these SDS optimization-based methods rely on the per-scene

optimization, which are usually time-consuming, taking from minutes to hours to generate a single 3D object. In addition, Zero123 [20] introduces the view-conditioned diffusion models to enhance the 3D consistency of SDS-based optimization. To improve the multi-view consistency, the following works [41, 49, 21, 22] propose generating the multi-view images simultaneously. Meanwhile, some studies [18, 21, 22] try to acquire sparse-view images from a single image and then optimize the 3D assets based on the sparse-view reconstruction technique. However, a primal issue is that these methods require test-time reconstruction, potentially resulting in additional time consumption and degraded quality.

More recently, pioneering research has attempted to generate high-quality 3D objects using a feed-forward model [11, 14, 64, 43, 52, 58, 54, 62], demonstrating stronger generalization capabilities and faster generation speeds compared to earlier methods. LRM [11] firstly adopts the transformer-based architecture to achieve high-quality and fast 3D object creation. Instant3D [14] combines a multi-view diffusion model with the transformer-based reconstruction model to generate 3D meshes from a single text or image prompt. Building on the LRM series [11, 14], InstantMesh [58] integrates Flexicubes into the transformer-based framework to enhance the smoothness and geometric details. MeshLRM [54] redesigns the transformer-based architecture and introduces more efficient training strategies. Additionally, a differentiable marching cube [53] is adopted to improve the rendering speed and quality. Instead of employing the transformer-based architecture, CRM [52] suggests generating high-resolution triplanes from orthographic multi-view images with the UNet backbone and utilizing Flexicubes [39] as the geometry representation. Taking into account the fast rendering speed and explicit representation, some studies [64, 43, 59, 50, 62] adopt Gaussian Splatting [13] as the 3D geometry representation to facilitate a fast training process. Specifically, GS-LRM [62] extends the architecture to the scene generation task. Overall, our proposed approach can be adapted to existing triplane-based feed-forward models.

3 Methodology

Our approach is broadly adopted in triplane-based feed-forward models with a multi-view diffusion and a sparse-view reconstruction model. As shown in Fig. 3, the input image I is sent to the multi-view diffusion model to generate multi-view images, which are then fed into the decoder to acquire the triplanes. Subsequently, triplanes with and without Freeplane are combined together to predict the geometry and texture.

In Sec. 3.1.1 and 3.1.2, we provide a brief introduction about triplanes and feed-forward models. Sec. 3.2 shows our key observation and analysis about triplanes. In addition, we present the details of our approach: **Freeplane**.

3.1 Preliminaries

3.1.1 Triplanes

TensoRF [3] proposes to swap the original MLP used in NeRF [27] with a feature volume to accelerate the training process. It factorizes the 4D tensors into low-rank multiple vactors and matrices:

$$\mathcal{T} = \sum_{r=1}^{R_1} v_r^1 \circ M_r^{2,3} + \sum_{r=1}^{R_2} v_r^2 \circ M_r^{1,3} + \sum_{r=1}^{R_3} v_r^3 \circ M_r^{1,2} \tag{1}$$

where v_r^1, v_r^2, v_r^3 are vector factors, and $M_r^{2,3}$, $M_r^{1,3}$, $M_r^{1,2}$ are matrix factors.

3.1.2 Feed-forward Methods

As mentioned before, tripalne-based feed-forward models [14, 52, 58] contain a multi-view diffusion model and a sparse-view reconstruction model. Zero123 [20] first explores the 3D prior information in pre-trained diffusion models and achieves arbitrary-view image synthesis. To further enhance the 3D consistency, the following works [41, 21, 40, 22, 49, 55, 24] try to fine-tune the pre-trained diffusion model to generate multi-view images simultaneously. The sparse-view reconstruction model aims to produce 3D meshes from the generated multi-view images, utilizing triplanes as the geometric representation. Triplanes play a significant role in linking 2D images to 3D representations, but they possess distinct properties depending on how they are acquired. Specifically, current methods treat

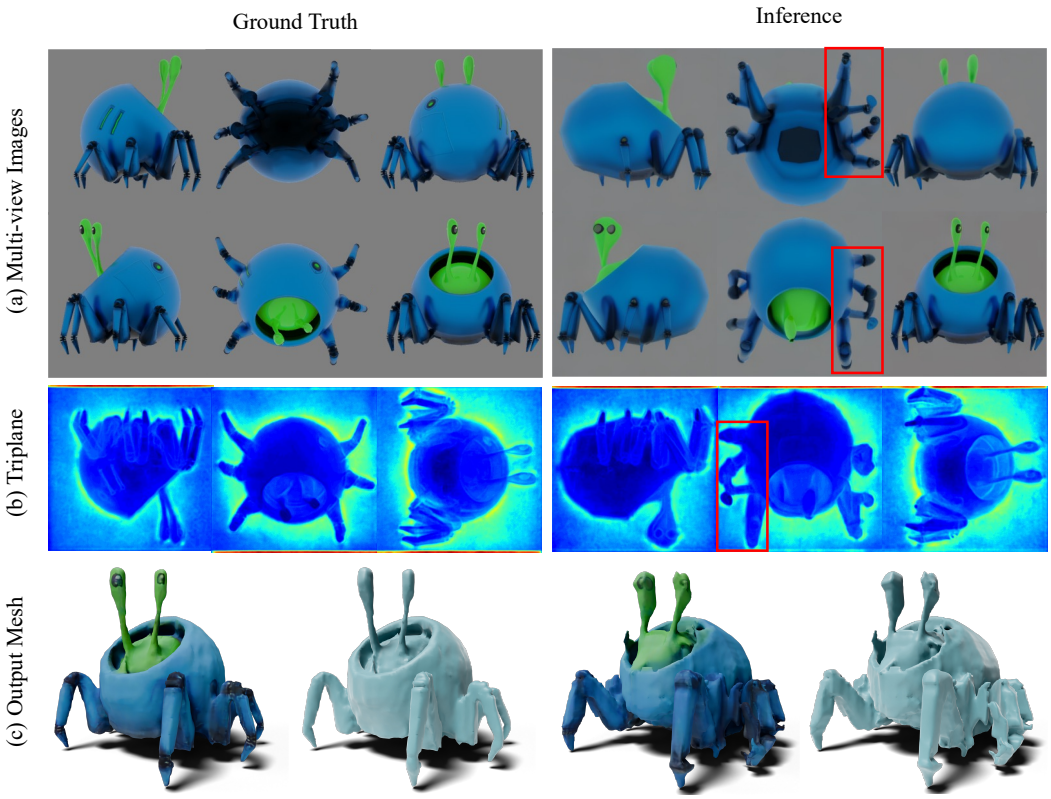


Figure 2: Our key observation is that inconsistent multi-view images cause high-frequency artifacts on the triplanes, resulting in low-quality meshes. Given ground-truth images, CRM can generate a smooth and highly detailed mesh. However, during the inference stage, using the inconsistent multi-view images from diffusion models causes apparent artifacts on the triplanes, leading to low-quality 3D assets.

triplanes either as 2D images or as a distinct 3D modality. Considering the generation quality and whether the model is open-source, we choose CRM [52] and InstantMesh [58] as our main focuses.

Treating triplanes as 2D images, CRM adopts a variant of ImageDream [49] and a UNet-based reconstruction model to produce highly detailed textured meshes. Generated from the multi-diffusion model, orthographic images are fed into the UNet to acquire high-resolution triplanes. On the other hand, adhering to the design in LRM, InstantMesh regards triplanes as a novel modality and connects them with 2D images using the cross-attention mechanism. In addition, InstantMesh opts for a fine-tuned Zero123++ [40], which includes multiple views from positive and negative elevations. To generate smooth meshes with geometric details, both CRM and InstantMesh adopt Flexicubes as the geometry representation.

3.2 Free Lunch in Triplane-Based Sparse-View Reconstruction Models

While these sparse-view reconstruction models achieve high-quality 3D generation, they encounter a gap between training and inference. It is noted that during the training phase, consistent multi-view images are used to create high-quality triplanes. However, in the inference stage, multi-view images generated by a fine-tuned diffusion model inevitably exhibit inconsistencies. Although data perturbation methods [43, 58] are employed during training to enhance the model's robustness to inconsistencies, these augmented images fundamentally differ from those generated by the diffusion model. As shown in Fig. 2, the inconsistency of multi-view images causes noticeable artifacts on the triplanes, ultimately leading to low-quality meshes. Specifically, despite the global consistency of these multi-view images, there are significant conflicts and inconsistencies in certain local areas, resulting in high-frequency artifacts on the triplanes. A similar issue also appears in InstantMesh,

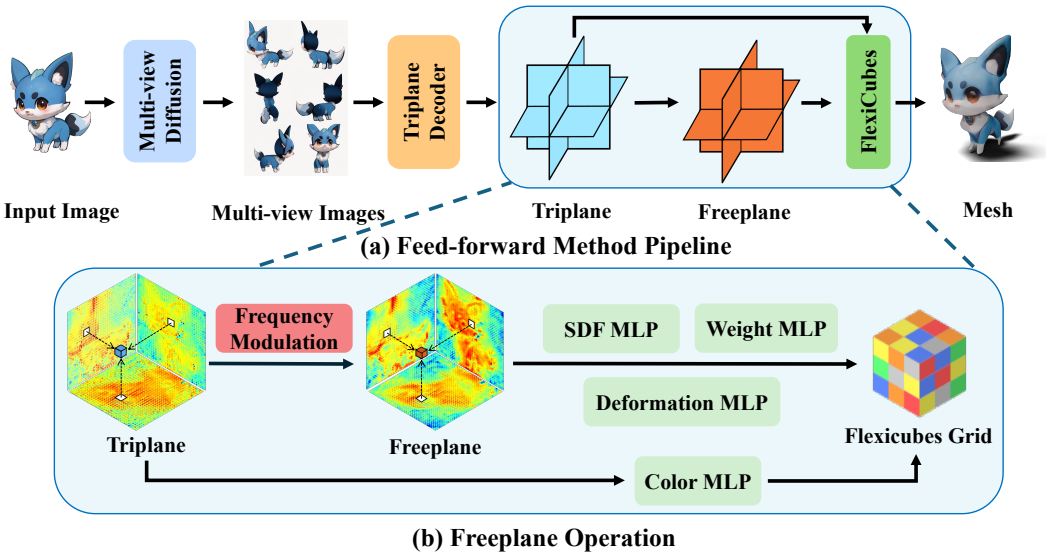


Figure 3: **Freeplane Framework. (a) Feed-forward Pipeline.** A single image is input into the multi-view diffusion model to generate six-view images, which are then fed into the triplane decoder. By querying the triplane features, Flexicubes are extracted to produce the textured mesh. We adopt the Freeplane approach on the triplanes. **(b) Freeplane Operation.** Low-frequency filtering is applied to modulate the original triplanes. Triplanes before filtering are used to compute texture-related features, while those after filtering are utilized to predict mesh geometry.

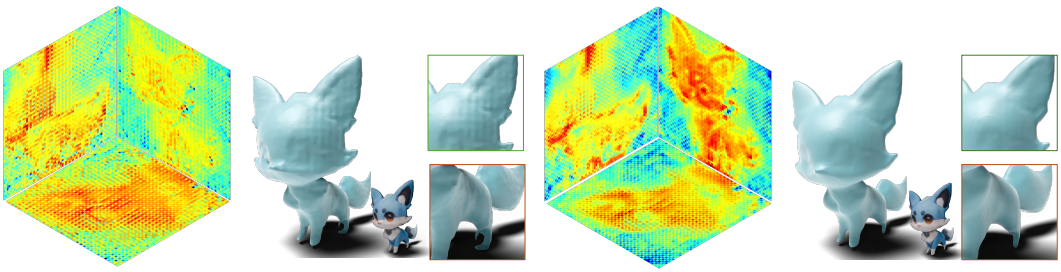
where the inconsistency introduces high-frequency noises on the triplanes. The visualization results of InstantMesh are provided in the Appendix A.4.

Building on the aforementioned discovery, we present our simple yet effective approach, denoted as **Freeplane**. It effectively alleviates artifacts caused by inconsistent multi-view images, which is achieved by erasing the conflicts on the triplanes. Our method drastically improves the generation quality without additional costs. We use G_M to represent the multi-view diffusion model, and G_D to denote the triplane decoder. Triplanes from G_D are denoted as $\boldsymbol{t}$, where the feature map of the i-th channel is $\boldsymbol{t_i}$. As shown in Fig. 3, the triplane features are then fed into the SDF, weight, deformation, and color MLPs to extract the Flexicube grids.

A key insight of our approach is that artifacts on the triplanes are essentially high-frequency components, consisting of either highly detailed conflicts or scattered noises. Consequently, a naive low-pass filter that aims to purify the high-frequency parts can be adopted to smooth the artifacts. However, we find that solely using an ordinary low-pass filter on the triplanes results in poor mesh quality. We believe that this is because the global low-frequency filtering causes a significant loss of geometric details. In our earlier analysis, we have demonstrated that the inconsistency primarily occurs in the local areas of triplanes. Building on this key observation, we propose a strategic filtering method that targets and removes high-frequency components in the surrounding areas. Moreover, to avoid the oversmoothed geometric surface, we further introduce a constraint to maintain the triplanes' boundary. Notably, our approach offers several impressive benefits. First, it erases the local artifacts on the generated triplanes to mitigate the negative influence brought by inconsistent multi-view images. Secondly, our Freeplane maintains the edge geometric information in triplanes, ensuring the apparent geometry details of 3D creations. The triplanes and 3D meshes in Fig. 4 confirm the effectiveness of our approach. In addition, as shown in Fig. 3, we propose using the triplanes before filtering to query color and the triplanes after filtering to predict geometry, ensuring fully utilize the texture priors in the pre-trained model.

Concretely, we adopt local frequency modulation for each channel in the triplanes and concate them together. Mathematically, our approach is presented as follows:

$$\boldsymbol{t}_{i,m} = \sum_{m \in t_i} \mathcal{K}_{l,\sigma}(|m-n|)\boldsymbol{t}_{i,n}, \tag{2}$$



(a) Triplane (b) Mesh w/o. Freeplane (c) Freeplane (d) Mesh w. Freeplane

Figure 4: We present triplanes and meshes w/o. and w. **Freeplane** from InstantMesh. Our approach erases the noises and produces apparent boundaries on the triplanes, leading to a smooth surface with geometry details.

where $\boldsymbol{t}_{i,m}, \boldsymbol{t}_{i,n}$ are the values of pixel m, n in the i-th channel of triplanes $\boldsymbol{t}$, and $\mathcal{K}_l$ is a low-pass kernel function to compute the local average values. In addition, the kernel size σ of $\mathcal{K}_l$ determines the range of neighboring pixels involved in the computation. To preserve the edge geometric details, we introduce another low-pass kernel filter, denoted as $\mathcal{K}_r$:

$$\boldsymbol{t}_{i,m} = \sum_{m \in t_i} \mathcal{K}_{l,\sigma}(|m-n|)\mathcal{K}_{r,\sigma}(|\boldsymbol{t}_{i,m} - \boldsymbol{t}_{i,n}|)\boldsymbol{t}_{i,n}. \tag{3}$$

For the practical implementation, we utilize the Gaussian kernel as the kernel function and use the Bilateral Filter [45] package in pytorch.

4 Experiments

4.1 Implementation details

We choose the open-sourced CRM and InstantMesh as our base models, which are both first-class 3D feed-forward generative models. They are pre-trained on the extensive 3D dataset Objaverse [6], providing them with robust generalization capabilities. During the inference stage, we freeze the entire model and only adjust the triplanes generated by the triplane decoder. Additionally, we apply a bilateral filter, setting the kernel size to 9 in CRM and 3 in InstantMesh. As shown in Fig. 6, we provide the kernel used in our experiments. Since the bilateral filter needs to consider edges and is not a linear kernel, we also provide the Gaussian kernel used here.

4.2 Main Results

Qualitative Results We evaluate our method on both synthetic and real-world datasets, including MVImageNet [61] and OmniObject3D [57]. Selected cases are presented in Fig. 5 and more results can be found in the Appendix A.2. It can be seen that our approach enhances the smoothness of the generated mesh without compromising the texture quality. Additionally, the inference time is nearly the same as that of the base model, effectively providing a "free lunch" for 3D generation.

Quantitative Results Similar to previous works [52, 58], we evaluate the effectiveness of our approach on the Google Scanned Objects (GSO) [7] dataset which is not included in the training dataset. We randomly select 30 shapes as our GSO test dataset. For each object, we render a 512×512 size front-view image as the input image.

To evaluate different methods, we follow previous reconstruction works [26, 19, 34] to report Volume IoU, Chamfer Distance and Normal Consistency Score(NCS). Volume IoU and Chamfer distance describe the geometry quality between the generated mesh and ground-truth mesh, while the Normal Consistency Score tells the smoothness of generated meshes. Prior to testing, we carefully align the pose between the evaluated mesh and the ground truth mesh and scale them to fit within a $[-0.5, 0.5]^3$ box. The results are shown in table 1.

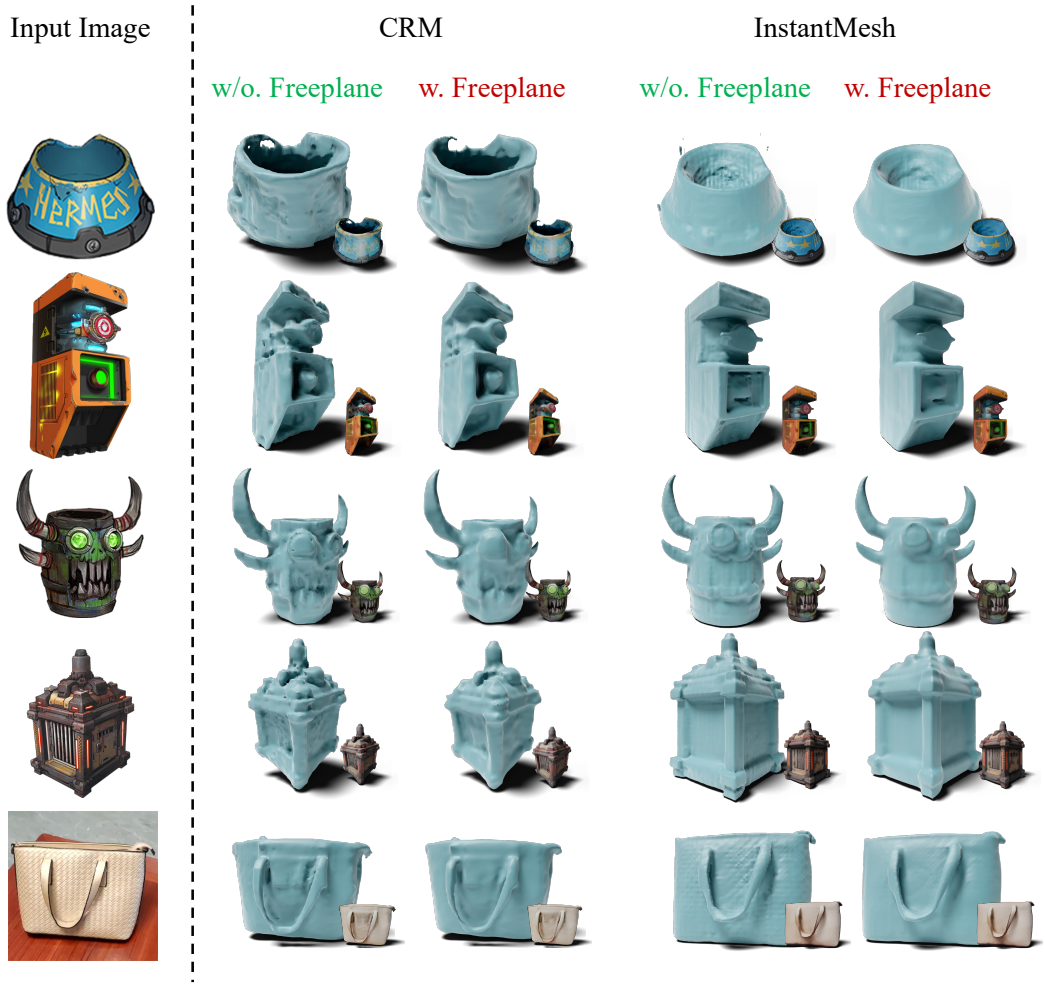


Figure 5: Qualitative Results: With Freeplane, both CRM and InstantMesh are able to generate smoother meshes with fewer artifacts while preserving their textures.

Table 1: Quantitative comparison for the geometry quality. We report the metrics of Volume IoU, Chamfer Distance and NCS on GSO dataset.

	CRM [52]		InstantMesh [58]	
	base	w. Freeplane	base	w. Freeplane
Vol. IoU.↑	0.422	0.434	0.429	0.448
Chamfer Dist.↓($\times 10^{-3}$)	16.06	16.12	16.34	14.73
NCS ↑(%)	67.01	67.30	68.20	69.37

4.3 Ablation Study

4.3.1 Frequency Filtering

Based on our formulation, we adopt different frequency filtering strategies, including bilinear, GaussianBlur, and BilateralFilter.

Bilinear Bilinear interpolation calculates new pixel values based on the weighted average of the four nearest pixels surrounding the desired location.

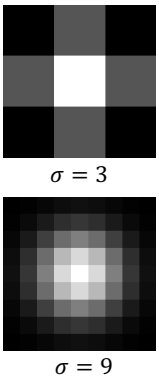


Figure 6: Gaussian kernel.

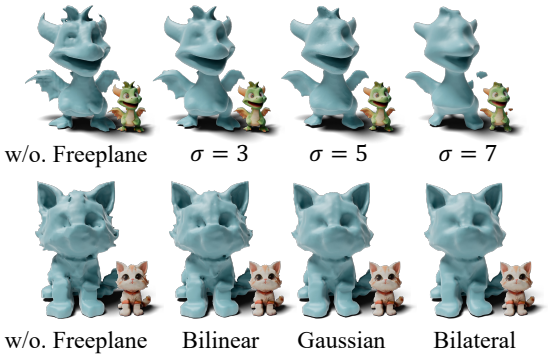


Figure 7: The first row is from InstantMesh, using a bilateral filter with varying kernel sizes. The second row is from CRM with different filterings.

Gaussian Blur Gaussian blur is a widely used image blurring technique to reduce image noise and detail. It works by convoluting the image with a Gaussian kernel.

Bilateral Filter The bilateral filter is an edge-preserving and noise-reducing filter. It applies a weighted average of neighboring pixels to each pixel, considering both spatial distance and intensity difference.

As shown in Fig. 7, in comparison to the original CRM, all these filters enhance the smoothness and geometric details of the generated meshes. Bilinear, being a fixed local averaging method, underperforms when the triplane resolution is high. Both Gaussian blur and bilateral filter produce similar effects, but the bilateral filter achieves superior local details. This is because the bilateral filter ensures local smoothing while preserving the triplane boundaries.

4.3.2 Kernel Size

We evaluate the effectiveness of our approach using various kernel sizes. As shown in Fig. 7, larger kernels result in smoother surfaces, though excessively large kernels can cause a loss of geometric details. Overall, the kernel size is correlated to the resolution of triplanes. Specifically, higher resolution triplanes necessitate larger kernels to properly modulate high-frequency components.

4.4 Limitations

While our method enhances the geometric smoothness and 3D coherence of the generated meshes, some limitations remain unresolved. First, our approach exhibits limited effectiveness in modeling slender structures, resulting in occasional omission of tiny details. Second, the choice of the frequency modulation strategy relies on the model architecture and the objects being generated, making it non-trivial to generalize. Integrating super-resolution and frequency modulation networks into the large reconstruction model could potentially enhance its capabilities, possibly requiring the retraining of feed-forward models from scratch. Due to the huge computation cost required, we leave this issue for the community to explore further. Nonetheless, we consider Freeplane as a simple yet effective baseline method to leverage the geometric priors of triplanes while maintaining robustness against inconsistent multi-view images.

5 Conclusion

In this work, we present Freeplane, a simple yet effective approach that significantly enhances the generation quality of 3D feed-forward models without requiring any additional costs. Focusing on the primary challenge posed by inconsistent multi-view images, we analyze the key role of triplanes in feed-forward models. Our study reveals the correlation between the inconsistency of multi-view images and the local high-frequency artifacts in triplanes. Leveraging this insight, we suggest strategically filtering out high-frequency conflicts in the triplanes and combining triplanes

before and after filtering to produce high-quality textured meshes. The artifact filtering operation effectively mitigates the negative effects caused by inconsistent multi-view images, leading to a smooth mesh surface while preserving geometric details. Our approach can be seamlessly integrated into existing triplane-based feed-forward models, enhancing their generation quality and robustness against inconsistent multi-view images. We hope that our research will inspire further exploration into constructing a training pipeline to exploit the frequency characteristics of triplanes.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[2] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022.

[3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022.

[4] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22246–22256, 2023.

[5] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *arXiv preprint arXiv:2309.16585*, 2023.

[6] 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.

[7] 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.

[8] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[10] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023.

[11] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

[12] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[14] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023.

[15] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12642–12651, 2023.

[16] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023.

[17] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

[18] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Minghua Liu, Xiaoshuai Zhang, and Hao Su. Meshing point clouds with predicted intrinsic-extrinsic ratio guidance. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 68–84. Springer, 2020.

[20] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023.

[21] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023.

[22] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023.

[23] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.

[24] Yuanxun Lu, Jingyang Zhang, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, Long Quan, Xun Cao, and Yao Yao. Direct2. 5: Diverse text-to-3d generation via multi-view 2.5 d diffusion. *arXiv preprint arXiv:2311.15980*, 2023.

[25] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021.

[26] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[27] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[28] Shentong Mo, Enze Xie, Ruihang Chu, Lanqing Hong, Matthias Niessner, and Zhenguo Li. Dit-3d: Exploring plain diffusion transformers for 3d shape generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[29] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[30] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022.

[31] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[32] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

[33] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[34] Albert Pumarola, Artsiom Sanakoyeu, Lior Yariv, Ali Thabet, and Yaron Lipman. Visco grids: Surface reconstruction with viscosity and coarea grids. *Advances in Neural Information Processing Systems*, 35:18060–18071, 2022.

[35] Edoardo Remelli, Artem Lukoianov, Stephan Richter, Benoit Guillard, Timur Bagautdinov, Pierre Baque, and Pascal Fua. Meshsdf: Differentiable iso-surface extraction. *Advances in Neural Information Processing Systems*, 33:22468–22478, 2020.

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. 2022 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021.

[37] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[38] Scott Schaefer and Joe Warren. Dual marching cubes: Primal contouring of dual grids. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, pages 70–76. IEEE, 2004.

[39] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023.

[40] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.

[41] 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.

[42] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022.

[43] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.

[44] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.

[45] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE, 1998.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[47] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023.

[48] 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.

[49] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.

[50] Yikai Wang, Xinzhou Wang, Zilong Chen, Zhengyi Wang, Fuchun Sun, and Jun Zhu. Vidu4d: Single generated video to high-fidelity 4d reconstruction with dynamic gaussian surfels. *arXiv preprint arXiv:2405.16822*, 2024.

[51] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

[52] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034*, 2024.

[53] 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. 2023.

[54] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality mesh. *arXiv preprint arXiv:2404.12385*, 2024.

[55] Haohan Weng, Tianyu Yang, Jianan Wang, Yu Li, Tong Zhang, CL Chen, and Lei Zhang. Consistent123: Improve consistency for one image to 3d object synthesis. *arXiv preprint arXiv:2310.08092*, 2023.

[56] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.

[57] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023.

[58] 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.

[59] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024.

[60] 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.

[61] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023.

[62] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *arXiv preprint arXiv:2404.19702*, 2024.

[63] Ruowen Zhao, Zhengyi Wang, Yikai Wang, Zihan Zhou, and Jun Zhu. Flexidreamer: Single image-to-3d generation with flexicubes. *arXiv preprint arXiv:2404.00987*, 2024.

[64] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. *arXiv preprint arXiv:2312.09147*, 2023.

A Appendix

A.1 Metric Explanation

As mentioned in the main paper, we follow ONet [26] to report volume IoU, Chamfer distance and normal consistency score(NCS) to evaluate baselines and our method. To compute the normal consistency score, let $\mathcal{M}_{\text{pred}}$ and $\mathcal{M}_{\text{gt}}$ represent the sets of points inside or on the predicted mesh and ground truth mesh, respectively. The NCS can then be determined using the following formula:

$$\text{NCS}(\mathcal{M}_{\text{pred}}, \mathcal{M}_{\text{gt}}) = \frac{1}{2}\left(\frac{1}{N}\sum_{i=1}^{N}\cos(\theta_i) + \frac{1}{N}\sum_{i=1}^{N}\cos(\phi_i)\right) \tag{4}$$

Where θ_i is the angle between the predicted normal vector and its nearest true normal vector, ϕ_i is the angle between the true normal vector and its nearest predicted normal vector and $\cos(\theta_i)$ and $\cos(\phi_i)$ are the cosine similarities, computed as the dot product of the vectors normalized by their magnitudes.

A.2 More Results

We provide additional qualitative results using images from the website, open-source real-world datasets, and our own photographed images. As shown in Fig. 8, our approach can be integrated into triplane-based sparse-view reconstruction models to improve the smoothness and geometric details.

A.3 Pseudo Code

We provide the pseudo code for our approach in Algorithm 1. Our approach includes the frequency modulation and the combination of triplanes before and after filtering to generate textured meshes.

Algorithm 1 Freeplane pseudo code.

```
# Input list:
# input multi-view images (img): [b, 6, c, h, w]
# frequency_type: BilateralFilter
# kernel_size: k
# resolution: resolution
# ctx: nvdiffrast.torch.RasterizeCudaContext()

# Output:
# 3D mesh: vertice, face, uvs, mesh_tex_idx, texture_map

# Freeplane framework
triplane = decoder(img) # produce triplanes from multi-view images
freeplane = BilateralFilter(triplane, kernel_size=k) # filter the original triplanes
vertice, face = get_geometry_prediction(freeplane) # get_geometry_prediction from freeplane
uvs, mesh_tex_idx, gb_pos, tex_hard_mask = xatlas_uvmap(ctx, vertice, face, resolution) # get uv map
texture = get_texture_prediction(triplane, gb_pos, tex_hard_mask) # query the texture field
background_feature = torch.zeros_like(texture) # build background features
img_feature = torch.lerp(background_feature, texture, tex_hard_mask) # extract the texture
texture_map = img_feature.permute(0, 3, 1, 2).squeeze(0)

return vertice, face, uvs, mesh_tex_idx, texture_map
```

A.4 Multi-view Inconsistency in InstantMesh

We provide the visualization results for InstantMesh. As shown in Fig. 9, the inconsistent multi-view images cause some apparent artifacts on the generated meshes. Our approach Freeplane substantially enhance the smoothness and geometric details in final meshes.

A.5 Social Impact

As with many other generative models, our approach could be used to produce malicious 3D content, warranting additional caution.

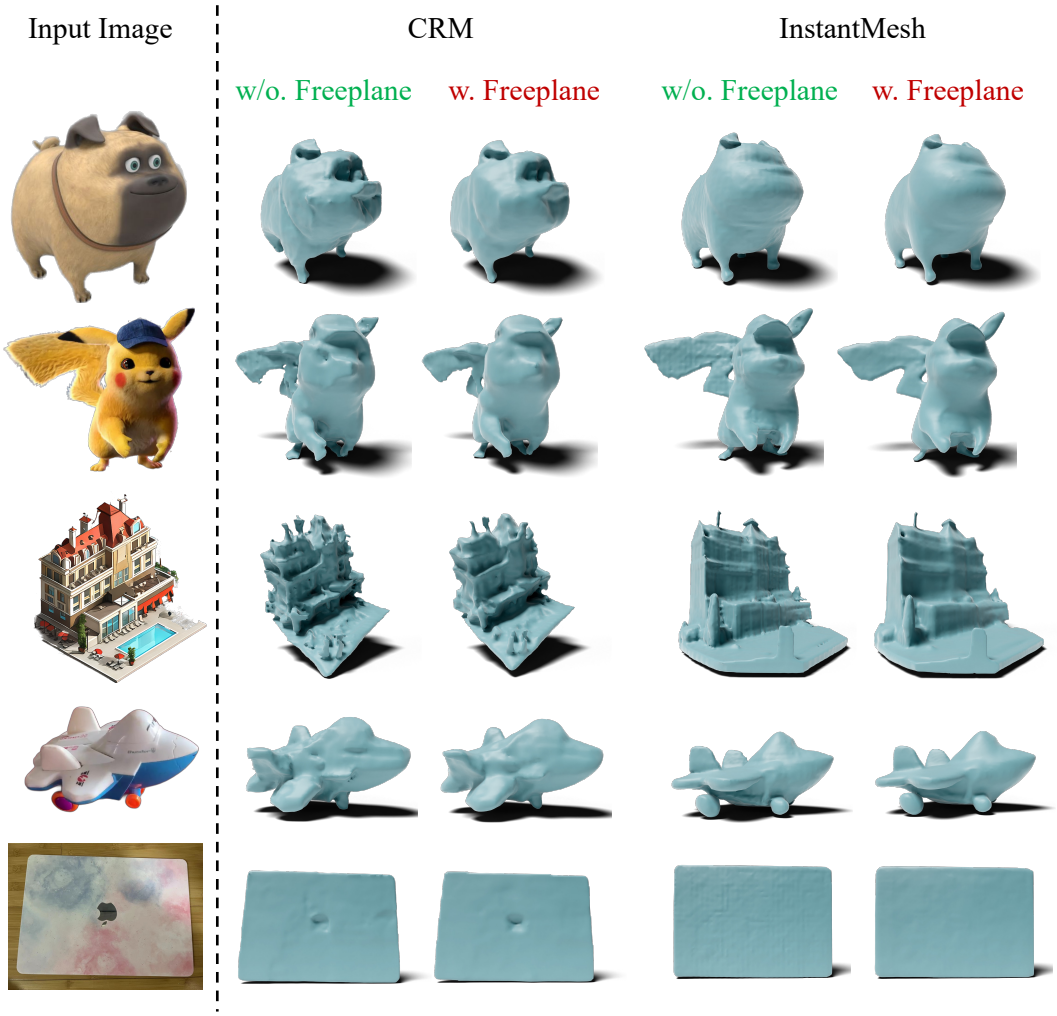


Figure 8: More results. Please zoom in to see the details of the meshes.

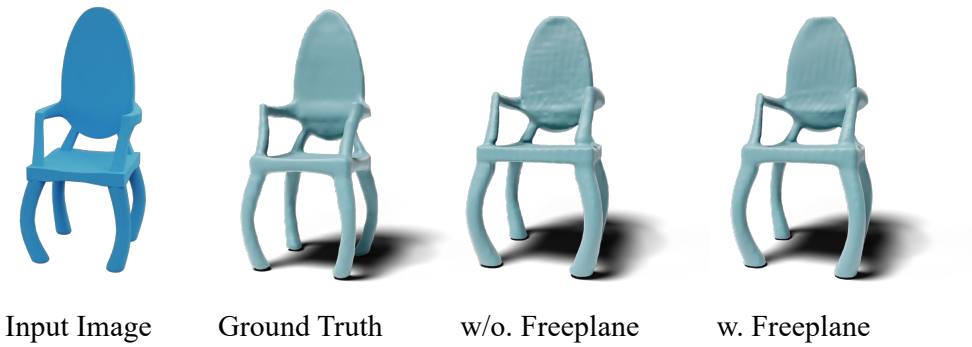


Figure 9: Visualization results for InstantMesh. Ground Truth means that ground-truth images are used to generate meshes, while w/o. Freeplane is the original InstantMesh, which leverages the multi-view images from Zero123++ as the inputs. Our approach Freeplane significantly improves the mesh quality.