

Fancy123: One Image to High-Quality 3D Mesh Generation via Plug-and-Play Deformation

Supplementary Material

A. Implementation details

A.1. Hyper parameters

We use the same camera settings as InstantMesh [13]. The default camera FOV (Field of View) angle is 30° , and the default distance from the camera to the origin is 4. The multiview images are from six views, where the elevation angle is absolute and the azimuth angle is relative. The elevation angles are $[20.0, -10.0, 20.0, -10.0, 20.0, -10.0]$ degrees, and the azimuth angles are $[30.0, 90.0, 150.0, 210.0, 270.0, 330.0]$ degrees, respectively.

The resolution G of the 2D deformation field grid is 20. We empirically set our loss weights w_1-w_6 for $\mathcal{L}_{\text{MSE}_1}$, $\mathcal{L}_{\text{mask}_1}$, $\mathcal{L}_{\text{smooth}_{2D}}$, $\mathcal{L}_{\text{MSE}_2}$, $\mathcal{L}_{\text{mask}_2}$, \mathcal{L}_{Lap} to be 1.0, 1.0, 0.001, 1.0, 0.1, and $1e5$, respectively. During the quantitative evaluation, the threshold for F-score is set to 0.2 following InstantMesh [13].

A.2. Geometry refinement

As mentioned in the second-to-last sentence of ??, we follow Unique3D [12] to refine the geometry of the mesh. Specifically, we first generate multiview normal maps as reference. Then, we optimize mesh vertex coordinates to approximate reference normal maps.

Multiview normal map generation. Any existing normal estimation method can be adopted here. Our backbone InstantMesh’s associated multiview diffusion model is finetuned from Zero123++ v1.2 [5], which also provides a multiview normal generation model [6]. This model creates multiview normal maps conditioned on both the input image and the generated multiview RGB images. Since InstantMesh uses white-background multiview RGB images, but the original Zero123++ uses gray-background ones for normal generation, so we first multiply the white-background images with a scale factor to make the whole image darker and the background gray, then feed the darker image to the normal generation pipeline. Although the RGB image is darker, we find that the generated normals are good. For the LGM [8] backbone in ??, we use the normal estimation model [1] from Unique3D [12] to generate multiview normal maps. Since both LGM and Unique3D use the same four views (front, back, left, right), Unique3D’s trained normal diffusion model works well on LGM.

Mesh vertex optimization. To refine the geometry, we optimize vertex vertices of the mesh to approximate the generated normal maps. Specifically, in each iteration, we render the mesh’s normal maps, and calculate the following losses

for backpropagation: MSE loss, mask loss, and expansion loss between rendered and reference normal maps, and a 3D Laplacian smooth loss. The MSE loss and the mask loss measure the average squared differences between the RGB and alpha channel values of corresponding pixels, respectively. The expansion loss is a regularization method proposed by Unique3D [12]. It measures the mean squared difference between the original vertex positions and their positions after being moved along their normals. The 3D Laplacian smooth loss is introduced in ??. The weights for MSE, mask, expansion and 3D Laplacian loss are 1.0, 1.0, 0.1, and $1e5$, respectively. Please refer to Unique3D [12] for more details.

A.3. Camera pose estimation

As mentioned in “Camera pose estimation” of ??, when the input image I^{in} has a relatively high absolute elevation angle, we need to estimate I^{in} ’s camera pose. Specifically, we first perform a coarse-to-fine grid search over all possible elevation angles, and then we further add a small optimization loop to further optimize the camera parameters. The details are presented below:

Coarse-to-fine search Since our backbone InstantMesh uses relative azimuth and absolute elevation angles, we only need to find a suitable elevation. As for azimuth, simply setting it to zero would ensure the alignment between the generated mesh and the input image. For other camera parameters, we simply set them as the default setting as in Appendix A.1. In the first stage (coarse), we search all elevation angles from -90° to 90° with the step of 3° . For each elevation angle, we use it to render our mesh, and calculate the LPIPS score between the rendered result and the input image. Among all angles, we adopt the angle ele_1 with the lowest LPIPS score, and feed it to the second stage. In the second (fine) stage, we search from $ele_1 - 3$ to $ele_1 + 3$ with the step of 1, and still adopt the angle with the lowest LPIPS score.

Optimization loop. We optimize the camera position for 100 iterations to minimize the difference between the mesh’s rendered result and the input image.

A.4. Alignment with GT for evaluation.

As mentioned in the paragraph before “qualitative results” in ??, some methods do not align with GT by default when comparing rendered results and ground truth images for quantitative evaluation, so we conduct an alignment process for them. Specifically, in all our baselines, there are

two methods that need this alignment: InstantMesh [13] and LGM [8].

- For InstantMesh, we directly adopt the method in Appendix A.3, since both InstantMesh and our Fancy123 use the same camera settings.
- For LGM, we use a similar coarse-to-fine grid search strategy and choose the camera parameters with the lowest LPIPS score as in Appendix A.3, except for the following differences:
 - We search both elevation and azimuth angles for LGM, as both angles are absolute.
 - In the first stage (coarse), we search elevation and azimuth angles from -90° to 90° , with the step size of 10° , resulting in $19 \times 19 = 361$ combinations.
 - In the second stage (fine), we search angles around the best angles found in the first stage. Specifically, we search from the best elevation angle ± 9 degrees and the best azimuth angle ± 9 degrees, with a step size of 2.

A.5. Other mesh deformation methods

As shown in ??, we compare our Jacobian-field-based 3D mesh deformation with other two methods: vertex replacement and 3D deformation field. Here we provide details on these two methods.

Vertex replacement: Directly optimizing mesh vertex coordinates instead of Jacobian fields. The losses are the same as when using Jacobian fields. From ??, we can see that this strategy fails to keep smoothness and plausibility. This is consistent with the observations in Fig. 3 of [3], where the authors compare Jacobian-based and vertex-replacement-based mesh deformation and the former is significantly more globally coherent.

3D deformation field: Using 3D deformation fields as a 3D grid to control the deformation of each mesh vertex, similar to the 2D deformation field we adopt in our appearance enhancement module. We optimize the offset of each grid vertex, and the offset of each mesh vertex is the linear interpolation of nearby grid vertices. We can see in ?? that this strategy still cannot ensure global smooth and plausible deformation.

B. More Quantitative results

B.1. elevation angle =elevation angle=0°

As mentioned in ?? “Datasets”, we provide results of using frontal views (elevation = azimuth = 0°) here in Tab. 1. Although our Fancy123 archives the best scores in 5 out of 7 metrics, we recommend focusing more on the qualitative results, see Appendix B.2.

Method	Appearance Metrics					Geometry Metrics	
	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑	CLIP-Sim. ↑	CD ↓	F-Score ↑
LGM [8]	81.29	0.4287	12.11	0.574	0.700	0.037	0.7605
TripoSr [10]	71.65	0.3805	13.06	0.608	0.724	0.030	0.8047
CRM [11]	82.18	0.4122	13.01	0.592	0.714	0.035	0.7768
InstantMesh [13]	53.00	0.3671	13.25	0.617	0.780	0.023	0.8451
Unique3D [12]	64.11	0.3821	12.79	0.601	0.754	0.033	0.7675
SF3D [2]	58.15	0.3665	13.75	0.595	0.749	0.026	0.8279
Ours	46.05	0.3521	13.79	0.633	0.803	0.024	0.8384

Table 1. Quantitative comparisons of our method against baseline methods (ranked by initial paper release time) for the single-image-to-3D-mesh task.

B.2. Inadequacies of existing quantitative metrics

As mentioned in ?? “Quantitative results”, the quantitative metrics often do not align with human perception.

Misalignment between metrics and human perception. Fig. 1 shows more examples. From (a) to (h), each sample shows the input image and rendering results of two different generated meshes.

- Initial Mesh: mesh generated by InstantMesh.
- Ghosting Mesh: unprojecting multiview images without 2D deformation to initial mesh.
- Clear Mesh: unprojecting multiview images after 2D deformation to the initial mesh.
- Deformed Mesh: deforming the clear mesh to match the input image.

Under each method’s generated mesh’s rendering result, we list the metrics comparing this very image and the corresponding input image (GT), together with human perception results. We can see that, for each sample, the mesh on the right is better by human perception, without blurring or ghosting, but the metrics disagree. Specifically: (1) the metrics only measure “similarity to GT” while ignoring significant artifacts, e.g. the thin black line under the eye in the red box of (a). (2) Even only considering “similarity to GT”, the metrics can differ from human perception, see (e) where “Clear Mesh” is too slim but its score is higher.

Contradiction between the ambiguous task and the unique GT. For the single-image-to-3D task, the input only contains one image from a single view, so the generated mesh’s novel views have various possibilities. However, existing metrics use one unique GT mesh for comparison, which is thus improper. Fig. 2 illustrates some examples of various possible backsides from a single-view image. Under such circumstances, though different backsides are plausible and would be equally treated by human perception, their metrics would differ significantly, as the metrics only compare with a unique GT.

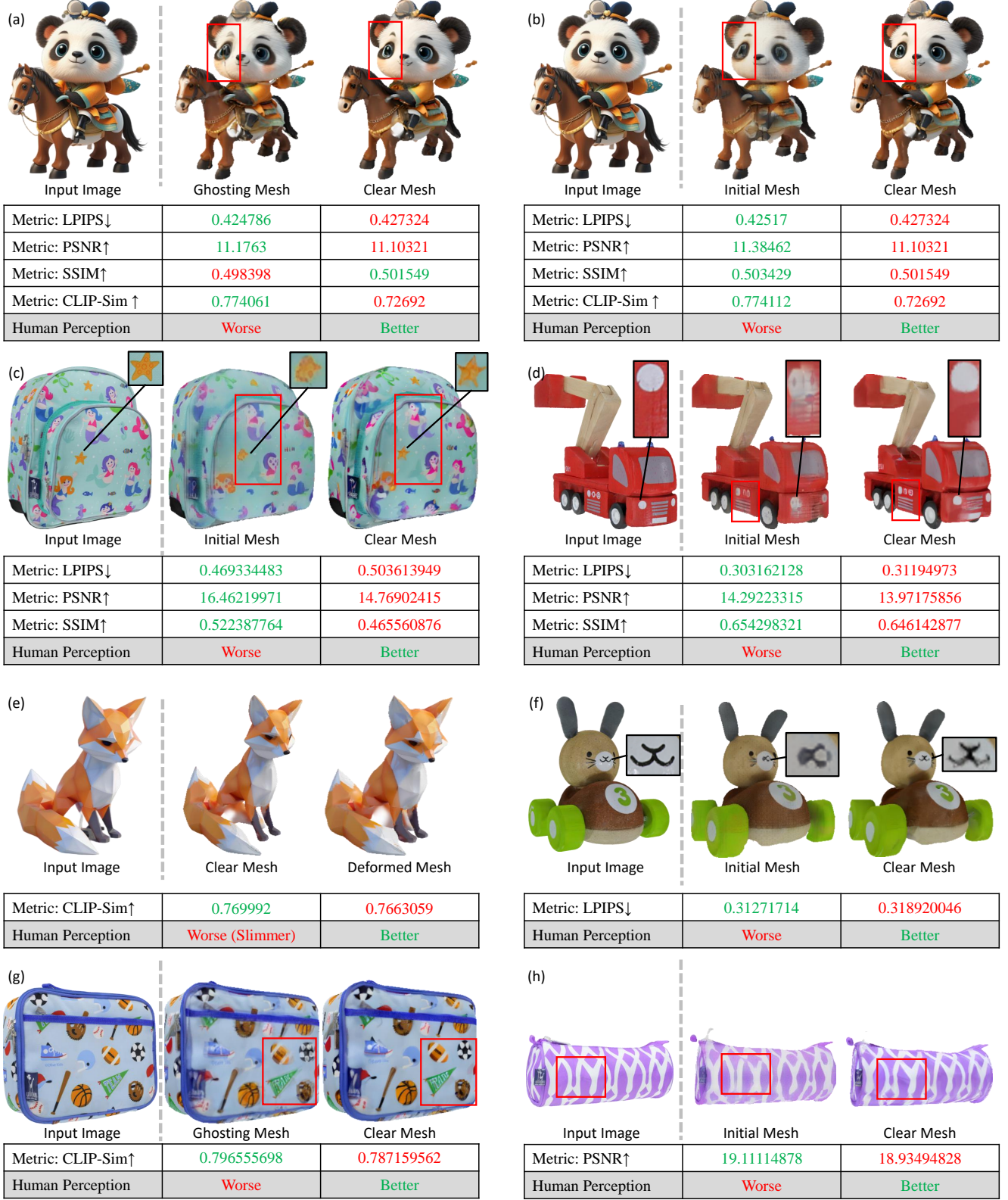


Figure 1. Illustration of the first limitation of existing quantitative metrics: mismatch with human perception. From (a) to (h), each sample shows the input image and rendering results of two different generated meshes. The mesh on the right is better by human perception, without blurring or ghosting. The tables below list common metrics calculated by comparing the rendered and input images, with the last row showing the comparison results of human perception. The metrics do not align with human perception.



Figure 2. Illustration of the second limitation of existing quantitative metrics: the contradiction between the ambiguous task and the unique GT. For each sample, we display the input image, the ground-truth mesh’s backside (GT Backside), and the backside of generated meshes (Gen. backside) from different methods including ours and baseline methods. We can see that, while there are different possibilities for the backside, the GT is unique. Therefore, existing metrics are improper, since they only compare results with the unique GT.

C. Initialization method and enhancement module replacement experiment

To validate the universal applicability of the enhancement modules in Fancy123, we replace the mesh initialization method InstantMesh with other baseline methods, and apply Fancy123 enhancement modules as follows:

(1) TripoSR/SF3D + our enhancement: Since neither TripoSR nor SF3D generates multiview images as intermediate products, Fancy123’s appearance enhancement module can not be applied. Thus, only geometry and fidelity enhancements are applied.

(2) Unique3D + our enhancement: Since Unique3D already incorporates geometry enhancement identical to Fancy123, only appearance and fidelity enhancements are applied.

(3) LGM/CRM/InstantMesh + our enhancement: We apply out complete enhancement pipeline (geometry, appearance, and fidelity) to meshes generated by LGM, CRM, and InstantMesh. For LGM and CRM, we generate multiview normal maps by the normal estimation model provided by Unique3D. All these three methods use the same four viewpoints (front, back, left, right), ensuring good model compatibility. As shown in Fig. 3 (a) and (b), our enhancement modules improved the generation quality for all initialization methods. Even for low-quality initial meshes in Fig. 3 (b) with geometric flaws (e.g., marked by red boxes), Fancy123 enhances coherence, smoothness, and overall plausibility.

Additionally, we compare Fancy123 with two typical enhancement techniques for single-image-to-3D:

(1) Input-MSE Enhancement: This intuitive approach optimizes mesh parameters by minimizing the mean squared error (MSE) loss between input images and mesh renderings. It is widely adopted in image-conditioned 3D generation (e.g., Make-It-3D [7], DreamGaussian [9]), and it only enhances regions visible from the input viewpoint. However, it produces significant artifacts when geometric misalignment exists between the mesh and input image, as shown in Fig. 3 (c) “Input-MSE refine”. In contrast, our Fancy123 achieves precise alignment through 3D deformation, avoiding such artifacts.

(2) SDS/PDM (Perturb-Denoise-MSE) Enhancement: SDS [4] is a foundational method for “2D diffusion for 3D generation”. It’s also commonly used in refinement stages [?]. It works by manually adding noise to mesh renderings and optimizing mesh parameters via SDS loss. DreamGaussian [9] notes that SDS-based UV color optimization introduces unwanted artifacts and proposes to replace SDS loss with MSE after perturbation, which we term “PDM”. While both SDS and PDM enhance the whole meshes across all viewpoints, they suffer from texture blurring due to inconsistent 3D guidance signals inherent in dif-

	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑	CLIP-Sim. ↑
LGM	86.24	0.395	13.254	0.612	0.710
+our refine	58.92	0.377	13.266	0.610	0.748
+input-MSE refine	69.44	0.407	12.745	0.575	0.696
+PDM refine	102.80	0.397	13.163	0.599	0.669
CRM	87.74	0.388	13.664	0.620	0.729
+our refine	57.65	0.368	13.756	0.626	0.784
+input-MSE refine	75.31	0.389	13.669	0.590	0.725
+PDM refine	102.18	0.391	13.570	0.594	0.686
TripoSR	68.52	0.367	13.876	0.629	0.739
+our refine	51.76	0.347	14.144	0.635	0.780
+input-MSE refine	66.96	0.369	13.880	0.619	0.726
+PDM refine	89.82	0.376	13.934	0.591	0.703
InstantMesh	46.16	0.349	13.735	0.634	0.800
+our refine	37.99	0.330	14.365	0.651	0.835
+input-MSE refine	40.82	0.354	13.848	0.619	0.803
+PDM refine	72.94	0.377	13.837	0.596	0.734
Unique3D	58.51	0.389	12.741	0.592	0.764
+our refine	53.69	0.379	13.094	0.597	0.774
+input-MSE refine	57.66	0.396	12.867	0.583	0.742
+PDM refine	97.35	0.414	12.918	0.558	0.662
SF3D	49.89	0.343	14.320	0.617	0.776
+our refine	50.66	0.332	15.091	0.648	0.790
+input-MSE refine	49.71	0.348	14.685	0.615	0.769
+PDM refine	79.77	0.359	14.220	0.621	0.728

Table 2. Results of different initialization methods without or with different refinement methods. Green and red indicate better and worse performance after refinement, respectively.

fusion denoising, as mentioned by DreamGaussian [9] authors. As shown in Fig. 3 (c) “PDM refine”, PDM may yield an even blurrier texture than the initial mesh.

Quantitative results in Tab. 2 show that our Fancy123 enhances all baselines, while Input-MSE and PDM degrade most metrics (green/red text indicating performance improvement/decline). We assume the key reasons are as mentioned above: (1) Input-MSE fails to handle geometric misalignment between 3D meshes and input images. (2) PDM reduces texture clarity due to inconsistent denoising guidance.

D. Can the order of enhancement modules be swapped?

No. (1) The initial mesh without appearance enhancement (AE) may be too blurry for fidelity enhancement (FE), since 3D deformation in fidelity enhancement is guided by RGB loss. (2) AE unprojects the deformed multiview images onto the mesh, and FE unprojects the input image. If we first apply FE and then AE, the result of FE will be overwritten by AE. So we recommend following the order in our paper to apply AM then FM, so as to first address multiview inconsistency (across all views) by AE then improve the input view quality by FE.

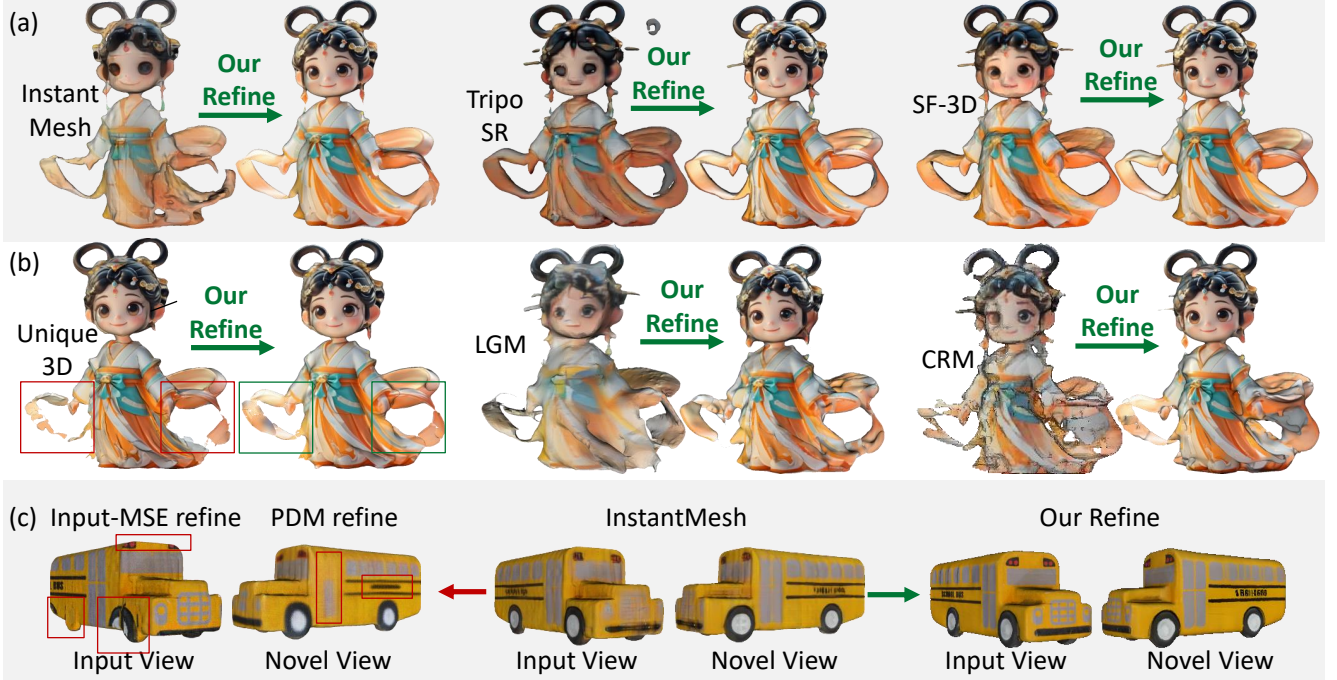


Figure 3. (a)-(b) Applying our Fancy123’s enhancement modules to different initialization methods. (b) Fancy123 improves the results even for relatively low-quality initial meshes. (c) Comparison of Fancy123’s refinement performance against Other refinement techniques.

E. Failure case

As mentioned in our manuscript, our 3D deformation module can sometimes lead to artifacts when different semantic parts of an object share similar colors. Fig. 4 shows an example. When deforming mesh (c) \mathcal{M}_c into (d) \mathcal{M}_d to match the input image (a) I^{in} , the bird’s brown back in (d) is mistakenly regarded as part of a branch and is therefore incorrectly raised. Our analysis is as follows:

- From Fig. 4 (e), we can see that the branch in the rendered image of \mathcal{M}_c , almost completely misaligns with the branch in I^{in} : the branch in I^{in} is on the right of the branch in \mathcal{M}_c .
- In Fig. 4 (e) and (c), immediately below the correct branch position (on the right side) is the bird’s brown back, which has a similar color to the branch.
- Therefore, the 3D deformation optimization loop chooses to raise the bird’s brown back to better match the input image, rather than moving the almost completely misaligned branch.
- As for the branch, the optimization loop adopts a locally optimal approach: minimizing or even eliminating the branch, so as to reduce its pixel count and thus lower the loss function value. To address such issues, we plan to further introduce semantic guidance in our 3D deformation module in the future. For now, we recommend skipping the 3D-deformation-based fidelity enhancement module, and only applying the 2D-deformation-based ap-



Figure 4. A failure case of our Fancy123 when different semantic parts share similar colors. When deforming mesh (c) \mathcal{M}_c into (d) \mathcal{M}_d , the bird’s brown back is mistakenly regarded as part of the brown branch, and therefore is incorrectly raised.

pearance enhancement module.

F. More qualitative results

F.1. Comparison with SoTA

We provide more visual comparisons between our Fancy123 and baseline methods in Figs. 5 and 6. InstantMesh, CRM, TripoSR, and LGM often produce blurry-looking meshes. SF3D and TripoSR exhibit lower plausibility in novel views. Unique3D often yields significant artifacts as shown in the red boxes. Unlike them, our Fancy123 archives high clarity and plausibility for various challenging input images.

F.2. Ablation study

2D deformation. Figs. 7 and 8 present more visual results on the effect of our 2D-deformation-based appearance enhancement module.

- (b): The original mesh generated by InstantMesh looks blurry.
- (c): Directly unprojecting the multiview images to the mesh without 2D deformation leads to ghosting.
- (d): After our appearance enhancement module: unprojecting the deformed multiview images to the mesh improves appearance quality, especially clarity.

3D deformation. Fig. 9 illustrates more examples on the effect of our 3D deformation operation. For simplicity, we directly use the mesh symbol \mathcal{M} to denote the rendered images of a mesh, omitting the rendering symbol \mathcal{R} .

- \mathcal{M}_c : the rendered image of the mesh before 3D deformation.
- \mathcal{M}_d : the rendered image of the mesh after 3D deformation.

To measure the matching degree between a mesh \mathcal{M} and the input image I^{in} , we perform two different forms of visualization: image overlay and subtraction:

- $(I^{\text{in}} + \mathcal{M})/2$: Overlay two images. The ghosting regions indicate mismatches.
- $|I^{\text{in}} - \mathcal{M}|$: Subtract two images. The non-black regions indicate mismatches, with brighter areas indicating greater mismatches.

From Fig. 9, we can see that, the undeformed mesh \mathcal{M}_c often fails to match the input image I^{in} , which prevents us from unprojecting I^{in} to \mathcal{M}_c . After 3D deformation, this mismatch issue is greatly alleviated. In other words, the mesh’s fidelity to the input image is greatly improved.

References

- [1] AiuniAI. Unique3d/app/custom_models/normal_prediction.py at main. https://github.com/AiuniAI/Unique3D/blob/main/app/custom_models/normal_prediction.py, 2024. 1
- [2] Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. SF3D: Stable Fast 3D Mesh Reconstruction with UV-unwrapping and Illumination Disentanglement. *arXiv preprint arXiv:2408.00653*, 2024. 2
- [3] William Gao, Noam Aigerman, Groueix Thibault, Vladimir Kim, and Rana Hanocka. TextDeformer: Geometry Manipulation using Text Guidance. In *ACM Transactions on Graphics (SIGGRAPH)*, 2023. 2
- [4] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 5
- [5] 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. 1
- [6] SUDO-AI-3D. zero123plus/examples/normal_gen.py at main. https://github.com/SUDO-AI-3D/zero123plus/blob/main/examples/normal_gen.py, 2024. 1
- [7] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In *ICCV*, pages 22819–22829, 2023. 5
- [8] 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. In *ECCV*, pages 1–18. Springer, 2024. 1, 2
- [9] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *ICLR*, 2024. 5
- [10] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. TripoSR: Fast 3D Object Reconstruction from a Single Image. *arXiv preprint arXiv:2403.02151*, 2024. 2
- [11] 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. In *ECCV*, pages 57–74. Springer, 2025. 2
- [12] Kailu Wu, Fangfu Liu, Zhihan Cai, Runjie Yan, Hanyang Wang, Yating Hu, Yueqi Duan, and Kaisheng Ma. Unique3D: High-Quality and Efficient 3D Mesh Generation from a Single Image. In *NeurIPS*, 2024. 1, 2
- [13] 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. 1, 2



Figure 5. More visual comparisons. Our method exhibits high clarity and plausibility compared to baseline methods. Better zoom in.



Figure 6. More visual comparisons. Our method exhibits high clarity and plausibility compared to baseline methods. Better zoom in.



Figure 7. Ablation experiments on the 2D-deformation-based appearance enhancement module: unprojecting the deformed multiview images to the mesh (d) archives clear-looking mesh without blurring (b) or ghosting (c).



Figure 8. Ablation experiments on the 2D-deformation-based appearance enhancement module: unprojecting the deformed multiview images to the mesh (d) archives clear-looking mesh without blurring (b) or ghosting (c).



Figure 9. Ablation experiments on 3D mesh deformation. For simplicity, \mathcal{M}_c and \mathcal{M}_d denote the rendered images of meshes before and after 3D deformation, respectively, omitting the rendering symbol \mathcal{R} . By comparing \mathcal{M}_c and \mathcal{M}_d with the input image I^{in} through addition or subtraction of them, we can see that \mathcal{M}_d matches I^{in} better than \mathcal{M}_c , thus paving the path for the unprojection of I^{in} to \mathcal{M}_d .