

# ARM: Appearance Reconstruction Model for Relightable 3D Generation

## Supplementary Material

### 9. Detailed explanation of Eq. 1

ARM models the appearance of object by a spatially varying BRDF described in Eq. 1. For the microfacet normal distribution term  $D$ , we use isotropic GGX distribution [69]:

$$D(\mathbf{n}, \mathbf{h}, \alpha) = \frac{\alpha^2}{\pi((\mathbf{n} \cdot \mathbf{h})(\alpha^2 - 1) + 1)^2}, \alpha = \rho^2,$$

where  $\mathbf{n}$  is the half-way vector. The Geometry function  $G$  is based on the Schlick-GGX Geometry function:

$$G(\mathbf{n}, \mathbf{l}, \mathbf{v}, k) = G_{\text{sub}}(\mathbf{n}, \mathbf{l}, k)G_{\text{sub}}(\mathbf{n}, \mathbf{v}, k),$$

where

$$G_{\text{sub}}(\mathbf{n}, \mathbf{v}, k) = \frac{\mathbf{n} \cdot \mathbf{v}}{(\mathbf{n} \cdot \mathbf{v})(1 - k) + k}.$$

Here,  $k = (\rho^2 + 1)^2/8$ . Last, the Fresnel term  $F$  is

$$F(\mathbf{v}, \mathbf{h}) = F_0 + (1 - F_0)(1 - (\mathbf{h} \cdot \mathbf{v}))^5,$$

where

$$F_0 = m c_d + (1 - m)0.04.$$

### 10. Details on GeoRM and GlossyRM

GeoRM and GlossyRM are built on the LRM framework, with a super-resolution upsampler added to the triplane synthesizer, as shown in Fig. 8.

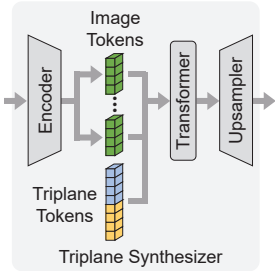


Figure 8. **Architecture of triplane synthesizer.**

A pretrained ViT image encoder [6] converts multi-view input images into image tokens. To make the network aware of camera pose, we add AdaLN camera pose modulation layers to the ViT encoder, following Instant3D [29], enabling pose-aware output tokens. The image encoder is jointly fine-tuned during

training. The super-resolution upsampler is based on SR-ResNet [28], using four Residual-in-Residual Dense Blocks with a filter size of 512. After these blocks, the upsampling steps consist of three convolutional layers, raising the triplane resolution to 256. Details of the remaining model components, including the encoder and transformer, are provided in Tab. 4.

While GeoRM and GlossyRM share the same architecture, they are trained as two distinct models. For GeoRM,

Input Views	6
Encoder Dim.	768
Transformer Dim.	1024
Transformer Layers	16
Transformer Heads	16
Triplane Resolution (Coarse)	32
Triplane Resolution (Fine)	256
MLP Hidden Layers	4
MLP Hidden Dim.	32

Table 4. **Specifications of GeoRM and GlossyRM.** Parameters for each component of the large reconstruction models used in our approach are listed.

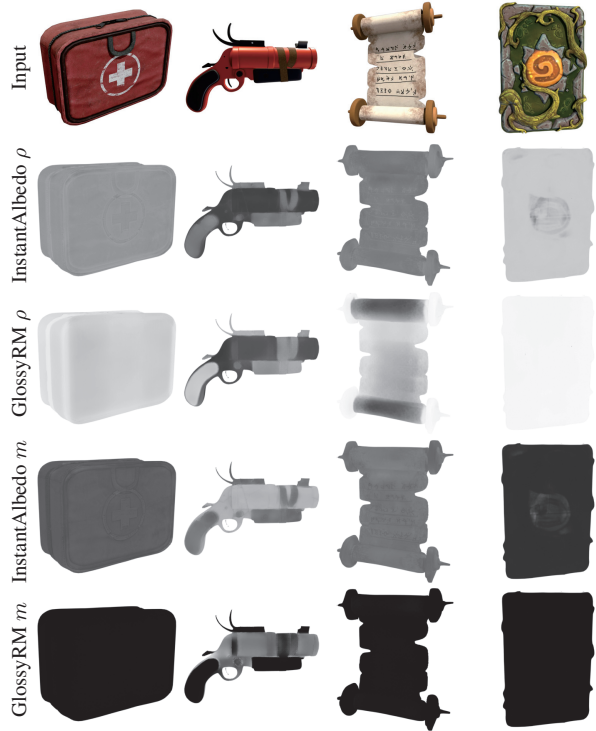


Figure 9. **Comparison with unified material prediction.** ARM separates the prediction of roughness and metalness by using GlossyRM, rather than predicting all material parameters within InstantAlbedo. We compare the differences between two approaches. InstantAlbedo tends to predict only intermediate values for roughness and metalness, making it difficult to produce extreme values close to 0 or 1, particularly for non-metallic objects.

Method	PSNR-D $\uparrow$	SSIM-D $\uparrow$	LPIPS-D $\downarrow$	PSNR- $\rho$ $\uparrow$	SSIM- $\rho$ $\uparrow$	LPIPS- $\rho$ $\downarrow$	PSNR- $m$ $\uparrow$	SSIM- $m$ $\uparrow$	LPIPS- $m$ $\downarrow$
SF3D [4]	16.937	0.834	0.205	18.012	0.873	0.202	20.433	0.862	0.153
Ours	<b>21.108</b>	<b>0.844</b>	<b>0.178</b>	<b>19.565</b>	<b>0.883</b>	<b>0.165</b>	<b>21.866</b>	<b>0.883</b>	<b>0.145</b>

Table 5. **Quantitative results of reconstructed PBR maps.** We report metrics comparing the predicted PBR maps with ground truth. Due to the high ambiguity in appearance decomposition, where multiple valid decompositions can explain the same shaded image, we only provide indicative scores in the supplementary material. Here, -D represents diffuse albedo,  $\rho$  denotes roughness, and  $m$  denotes metalness.

we adopt a two-stage training strategy similar to [81]. In the first stage, we load pretrained weights for all components except the newly introduced super-resolution module and train using a volume rendering loss. In the second stage, we employ differentiable marching cubes to extract iso-surface from the queried density grid, followed by rendering with a differentiable rasterizer [27].

After training GeoRM, we proceed to train GlossyRM while keeping GeoRM fixed. Specifically, we first use GeoRM to generate the 3D shape from the multi-view input. Then, for each vertex on this generated shape, we retrieve features from GlossyRM’s triplane and feed them into the decoding MLP to predict roughness and metalness. These per-vertex properties are then used to render multi-view images, with a loss computed against ground-truth images to guide GlossyRM’s training. For faster convergence, GlossyRM is initialized with GeoRM’s weights at the start of training.

## 11. Unified material prediction

ARM separates PBR parameter prediction into two networks: InstantAlbedo for diffuse albedo and GlossyRM for roughness and metalness. Although predicting all material properties within InstantAlbedo might seem more straightforward, our experiments indicate that this approach results in inaccurate material decomposition, as shown in Fig. 9. InstantAlbedo tends to predict only intermediate values for roughness and metalness, making it difficult to produce extreme values close to 0 or 1, particularly for non-metallic objects. Notably, for SVBRDF, human perception is generally more sensitive to spatial variations (subtle pixel changes within textures) than to angular variations (subtle changes of lighting and view direction in BRDF). By leveraging GlossyRM, which has ample network capacity, our method effectively produces realistic appearances, with InstantAlbedo capturing the fine-grained details in diffuse albedo.

## 12. Failure cases of material prior & FFC-Net

Additional examples of our material prior and FFC-Net are shown in Fig. 10. In the top part, the material prior struggles with highly ambiguous input images. For example, the metallic ring on the bottle’s neck is incorrectly assigned the

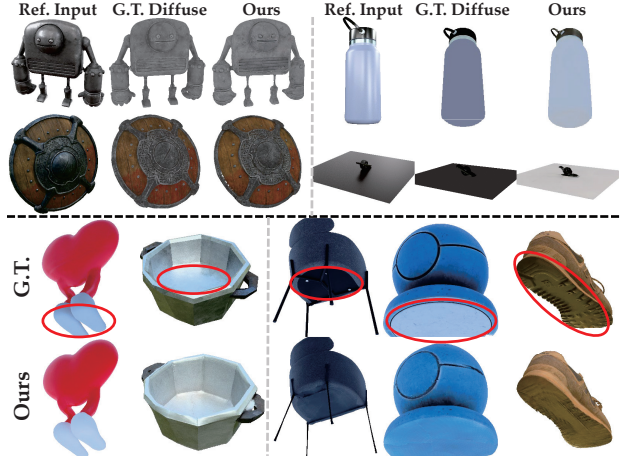


Figure 10. **More examples on albedo decomposition and texture inpainting.** The material prior fails to accurately decompose the diffuse albedo for the bottle and plane objects (**top**), while FFC-Net’s inpainting results do not align well with the ground truth for the rightmost three objects (**bottom**). Unseen inpainted areas are highlighted in red.

same color as the body. Similarly, the plane object is misclassified as a white metallic object instead of a black diffuse one. In the bottom part, the FFC-Net fails to inpaint patterns or colors missing from the input image, such as the specific pattern on the shoe’s bottom or the light blue at the bottom of the material sphere when only dark blue is present in the input.

## 13. Details on InstantAlbedo

The InstantAlbedo framework comprises three main networks: a material-aware image encoder, a U-Net, and an FFC-Net. The material-aware image encoder is based on [58], excluding the user reference injection and cross-attention layers. The intermediate features of different resolutions output by DINO, are fused using convolutional neural networks to generate a feature map matching the input image resolution. The total parameter number of InstantAlbedo is about 300M. For the FFC-Net, we use a ResNet-like architecture [19] with 3 downsampling blocks, 4 residual blocks, and 3 upsampling blocks. In our model, the residual blocks utilize FFC with a filter size of 512.



Figure 11. **Qualitative comparison.** We present examples of single-image 3D generation across different methods. While other methods exhibit blurriness, ARM reconstructs complex patterns with sharp details. Please zoom in to examine the texture quality.

## 14. Dataset selection

GeoRM and GlossyRM are trained on a 150K subset of the Objaverse dataset [12]. This subset is carefully curated based on the following criteria to ensure high-quality training data:

1. Each selected object must include a roughness map or a metalness map. This requirement ensures that the objects have sufficient material data for training

GlossyRM.

2. The object must not be a point cloud, nor a sparse or small object with low occupancy (fewer than 10 pixels per rendered view).
3. Low-quality objects, such as scanned indoor data or large scenes with multiple objects, are excluded.



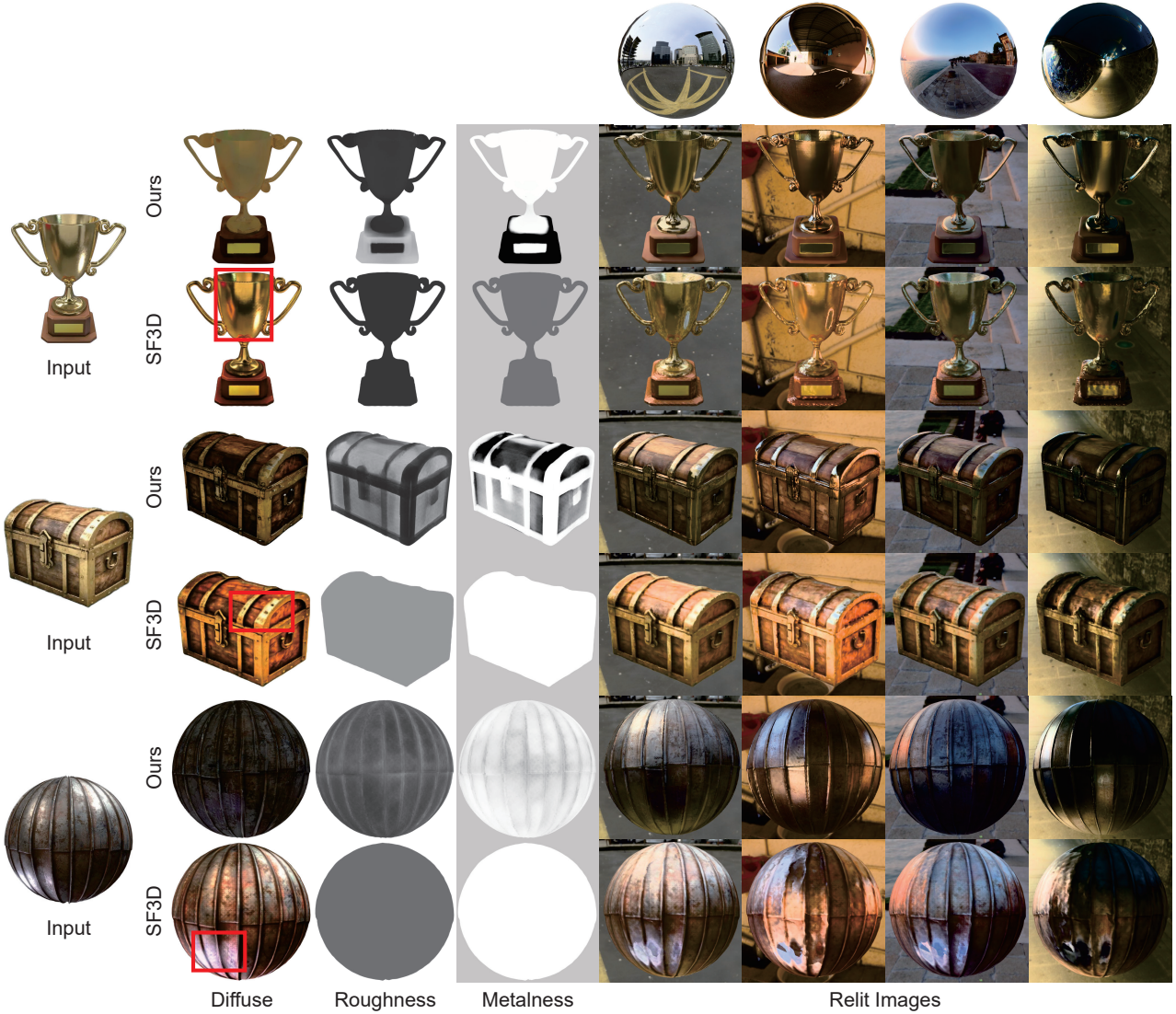


Figure 12. **PBR comparison.** We compare reconstructed PBR maps and relit images under novel lighting to SF3D [4]. While SF3D produces constant roughness and material with lighting baked into the diffuse color (highlighted in the figure), our method generates spatially-varying appearance, with well-separated illumination and materials.

## 15. Shape alignment

During evaluation, we align each method’s predicted meshes to the ground truth meshes before calculating metrics, as coordinate frames may differ across methods. Following MeshFormer [35], we use a two-step alignment based on the evaluation metric. First, we normalize both ground truth and predicted meshes to fit within a bounding box in the range  $[-1, 1]^3$ . Then, we uniformly sample rotations in  $[0, 2\pi)$  and scales in  $[0.7, 1.4]$  for initialization, refining the alignment using the Iterative Closest Point (ICP) algorithm. We select the alignment with the highest evaluation score.

Once aligned, we compute metrics for each method. For 3D metrics, we sample 100,000 points on both the ground truth and predicted meshes to calculate the F-score and Chamfer Distance, setting a threshold of 0.1 for the F-score. To evaluate texture quality, we compute PSNR, SSIM, and LPIPS between images rendered from the reconstructed mesh and ground truth. We sample 32 camera poses in a full 360-degree view around the object, rendering RGB images at a resolution of  $320 \times 320$ . Since we use the VGG model for LPIPS loss during training, we use the Alex model for LPIPS evaluation.



## 16. Additional results

In Tab. 5, We report quantitative metrics comparing the predicted PBR maps with ground truth, using SF3D and our method. Due to the high ambiguity in appearance decomposition, where multiple valid decompositions can explain the same shaded image, we only provide indicative scores in the supplementary material.

Fig. 11 presents complete qualitative examples, including comparisons with LGM and CRM. In Fig. 12, we provide further examples along with additional relighting results.