Material Anything: Generating Materials for Any 3D Object via Diffusion (Supplementary Material)

Xin Huang[1*], Tengfei Wang[2†], Ziwei Liu[3], Qing Wang[1†]

[1]Northwestern Polytechnical University, [2]Shanghai AI Lab, [3]S-Lab, Nanyang Technological University

A. Material3D Dataset

A.1. Data Construction

To train Material Anything, we constructed a dataset *Material3D* that consists of 80K high-quality 3D objects curated from the Objaverse dataset, focusing specifically on models with comprehensive material maps. We further filtered objects using Blender, ensuring the presence of essential material maps: base color, roughness, metallic, and bump. Only models containing all these material properties were retained. Each object was imported into Blender, where Blender's Smart UV Project tool was used to generate UV mappings. For objects with multiple parts, all components were merged into a single mesh to ensure UV mapping could be projected onto a unified 2D map. After this filtering and preparation process, we rendered multi-view material images (albedo, roughness, metallic, and bump) from 10 fixed camera positions that are consistent with the setup used in our material generation phase, which served as training data for the material estimator. In addition, we rendered images under varying lighting conditions and included normal maps as input for model training, providing diverse lighting and surface information. UV material maps and CCM were also rendered to facilitate the training of the material refiner.

A.2. Lighting Conditions

Inspired by the image relighting method [5], we incorporated multiple lighting categories for rendering input images, enabling the model to handle diverse lighting scenarios.

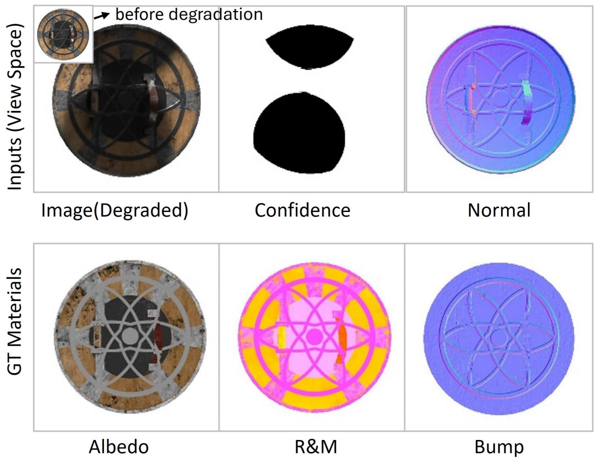
1. *Point Lighting.* Point light sources are uniformly sampled from a hemisphere (with $0° \leq \theta \leq 60°$) surrounding the object, with a radius sampled in the range [4m, 5m]. The number of point lights is randomly sampled between [1, 3]. The sum power of all lights is uniformly chosen within [900W, 2400W]. To ensure the visibility of highlighted regions, the hemisphere is rotated according to the camera position, while the camera itself remained fixed at the top of the hemisphere.
2. *Area Lighting.* Similar to point lighting, area light sources are sampled from a hemisphere (with $0° \leq \theta \leq 60°$) with a radius from 4m to 5m. The size of the area light ranges from 3m to 10m, and its power is uniformly selected within [1000W, 2000W]. Only one area light is utilized during rendering.
3. *Environment Lighting.* Environmental lighting broadly influences scene illumination beyond isolated light sources. To counter the white balance bias common in diffusion-generated images, we employ white environment lighting with strengths ranging from [0.5, 3], avoiding colored HDR environment maps.
4. *Without lighting.* To ensure the material estimator can accurately predict materials independent of lighting conditions, we render views of objects using only albedo textures, which is the same as rendering multi-view albedo maps.

For each camera position, we render 13 images (including albedo, roughness, metallic, bump, normal maps, and 8 RGB images under point, area, and environment lighting). For UV material map rendering, we utilize Blender's smart UV project to unwarp the mesh, producing five UV space maps (albedo, roughness, metallic, bump, and canonical coordinate maps).

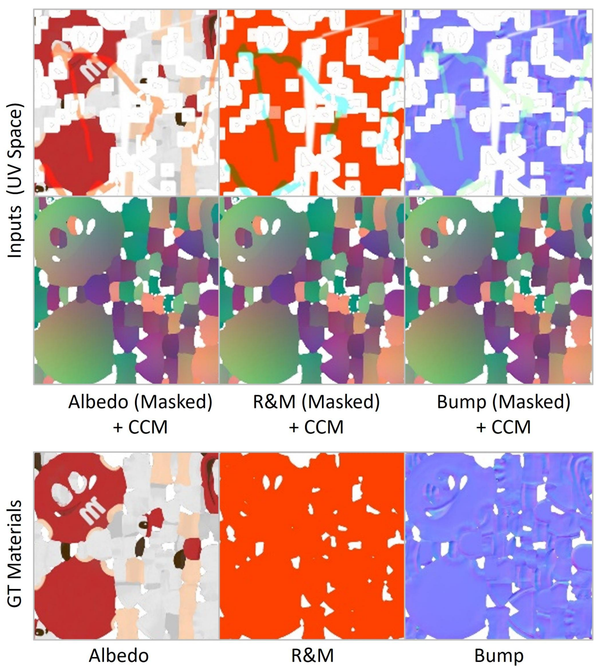
A.3. Simulating Inconsistent Lighting Effects

To improve the robustness of the material estimator, we randomly select two images under different lighting conditions for a camera view and stitch portions of each into a composite during training. This enables a single image to exhibit two distinct lighting types, simulating the inconsistency in multi-view materials. Furthermore, we introduce degradations to one of the images, applying effects such as blurring and color shifts. A confidence mask is used to delineate the regions that have undergone degradation. The final input to the material estimator comprises the stitched image, the confidence mask, and the normal map, as shown in Fig. 1 (a). To train the material refiner, we randomly

* Work was done during an internship at Shanghai AI Lab.
† Corresponding authors.



(a) A training case for Material Estimator



(b) A training case for Material Refiner

Figure 1. **The virtualization of our training data.** We apply various degradations and simulate inconsistent lighting effects in the inputs to enhance the robustness of our method.

mask regions of the UV material maps and apply degradations such as blurring and color shifts. These masked material maps are taken as input, as shown in Fig. 1 (b). The CCM, derived from the UV mapping of 3D point coordinates, is also included. These maps guide the areas requiring inpainting and facilitate the integration of 3D adjacency information during the diffusion process.

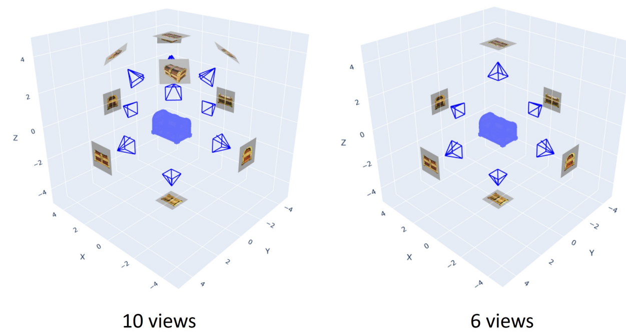


Figure 2. **The camera poses for progressive material generation and building training data.**

B. Implementation Details

B.1. Training Details

We implemented Material Anything using the Diffusers [2], with Stable Diffusion v2.1 [3] serving as the backbone diffusion model. The training process leverages the AdamW optimizer with a learning rate of 5×10^{-5}. Our material estimator was trained over 300K iterations on 8 NVIDIA A100 GPUs with a batch size of 32, requiring approximately 5 days to complete. In parallel, the material refiner was trained for 150K iterations under the same GPU configuration and batch size, with a training duration of about 2 days. Training data was rendered at a resolution of 512×512 using Blender's Cycles path tracer, ensuring high-quality reference materials for robust learning.

B.2. Material Generation Details

During material generation, each input object is centered within a normalized bounding box. To capture comprehensive material properties, 6 or 10 views are rendered, as illustrated in Fig. 2. The input image resolution for our material refiner is set to 768×768, while the resolution for UV material maps is 1024×1024. This setup ensures high-fidelity material maps that are detailed and adaptable across different viewing angles. For the input objects without UV mappings, xatlas [4] is used to unwrap them. All results, including those from our method and the baselines, are generated on a single NVIDIA A100 GPU.

C. Applications

Material Anything offers robust capabilities to edit and customize materials of texture-less 3D objects by simply adjusting the input prompt, enabling flexible and intuitive material manipulation. As illustrated in Fig. 3, we demonstrate that a barrel's material can be transformed into realistic textures like wood, gold, and stone, showcasing the versatility of our approach across various material types. This application allows users to dynamically adapt 3D models to spe-

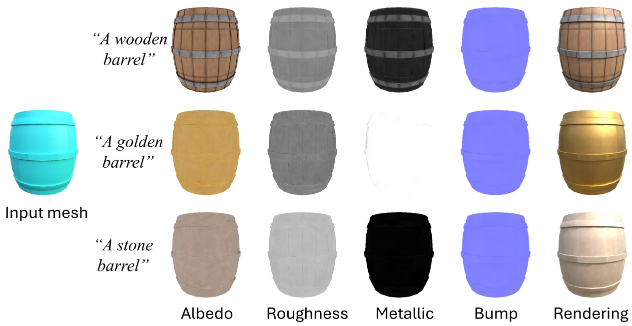


Figure 3. **Material editing with prompts.** Material Anything enables flexible editing and customization of materials for texture-less 3D objects by simply adjusting the input prompt.

cific aesthetic or functional requirements, enhancing asset adaptability for virtual environments, simulations, and design visualization.

Furthermore, our method supports relighting, enabling objects to be viewed under different lighting conditions, as shown in Fig. 6. Material Anything generates material properties for each object, ensuring physically consistent relighting and enhanced realism. This functionality allows for more accurate simulations in AR, VR, and digital content creation, where realistic lighting is essential for immersion. Collectively, these capabilities make the proposed method a versatile and efficient solution for content creators and researchers aiming to produce high-quality, relightable 3D objects with customized materials.

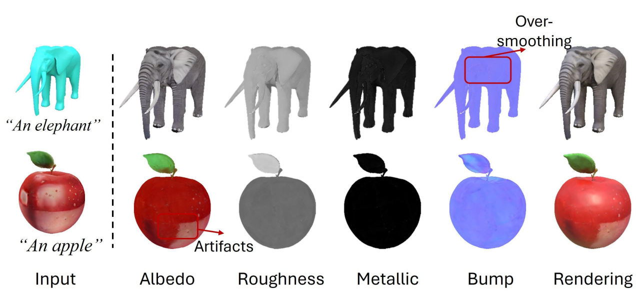


Figure 4. **Failure Cases by Material Anything.**

D. Limitations and Failure Cases

Material Anything is designed to address the complex task of generating materials for a diverse range of 3D objects. However, our approach has certain limitations. First, owing to the characteristics of the Objaverse, where many objects exhibit uniform roughness and metallic attributes with minimal surface details in bump maps, our method may produce materials with constrained surface details. This limitation is illustrated in the elephant example in Fig. 4, where the resulting bump maps lack details. Additionally, for objects with existing textures, our method struggles to remove prominent artifacts. For example, in the apple instance in Fig. 4, large white artifacts are misinterpreted as part of the texture, resulting in an inaccurate albedo.

E. Additional Results

E.1. More Visual Results

We present additional qualitative results to illustrate the effectiveness of Material Anything. **Video results are present in our supplementary video.** In Fig. 7, we display results generated by our material estimator on the Objaverse dataset, compared with their GT materials. As shown, our method effectively generates materials closely aligned with the ground truth, capturing essential details and textures to enhance realism. In Fig. 8, we show additional results on texture-less inputs, demonstrating our method's capability to handle complex UV mappings. Despite the complexity of certain UV layouts, our method consistently generates high-quality material maps in UV space, preserving material fidelity across the entire surface. Finally, we present additional results on various input types, including generated models, albedo-only inputs, and scanned 3D objects. These examples, shown in Fig. 9, highlight our method's robustness across varied lighting conditions and input characteristics, demonstrating its versatility in producing realistic materials adaptable to diverse lighting environments.

E.2. Comparisons with FlashTex

To further evaluate our method, we compare it with FlashTex [1], an SDS-based method, similar to DreamMat [6] compared in the main paper. Our method outperforms FlashTex in FID↓ (100.63 vs. 110.55), CLIP scores↑ (31.06 vs. 30.71), and Inference Time↓ (3 mins vs. 4 mins). As shown in Fig. 5, our results have clearer textures, while FlashTex often produces oversaturated artifacts, a common issue in SDS-based methods.

Figure 5. Comparisons with FlashTex

References

[1] Kangle Deng, Timothy Omernick, Alexander Weiss, Deva Ramanan, Jun-Yan Zhu, Tinghui Zhou, and Maneesh Agrawala. Flashtex: Fast relightable mesh texturing with lightcontrolnet. In *ECCV*, pages 90–107, 2024. 3

Figure 6. **Relighting results by Material Anything under various HDR environment maps.** The left column displays the input textureless meshes, while the top row presents the HDR environment maps used.

[2] Huggingface. Diffusers. https://github.com/huggingface/diffusers, 2024. 2

[3] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2

[4] Jonathan Young. Xatlas. https://github.com/jpcy/xatlas, 2024. 2

[5] Chong Zeng, Yue Dong, Pieter Peers, Youkang Kong, Hongzhi Wu, and Xin Tong. Dilightnet: Fine-grained lighting control for diffusion-based image generation. In *ACM SIGGRAPH Conference Proceedings*, pages 1–12, 2024. 1

[6] Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, et al. Dreammat: High-quality pbr material generation with geometry-and light-aware diffusion models. *ACM TOG*, 43(4):1–18, 2024. 3

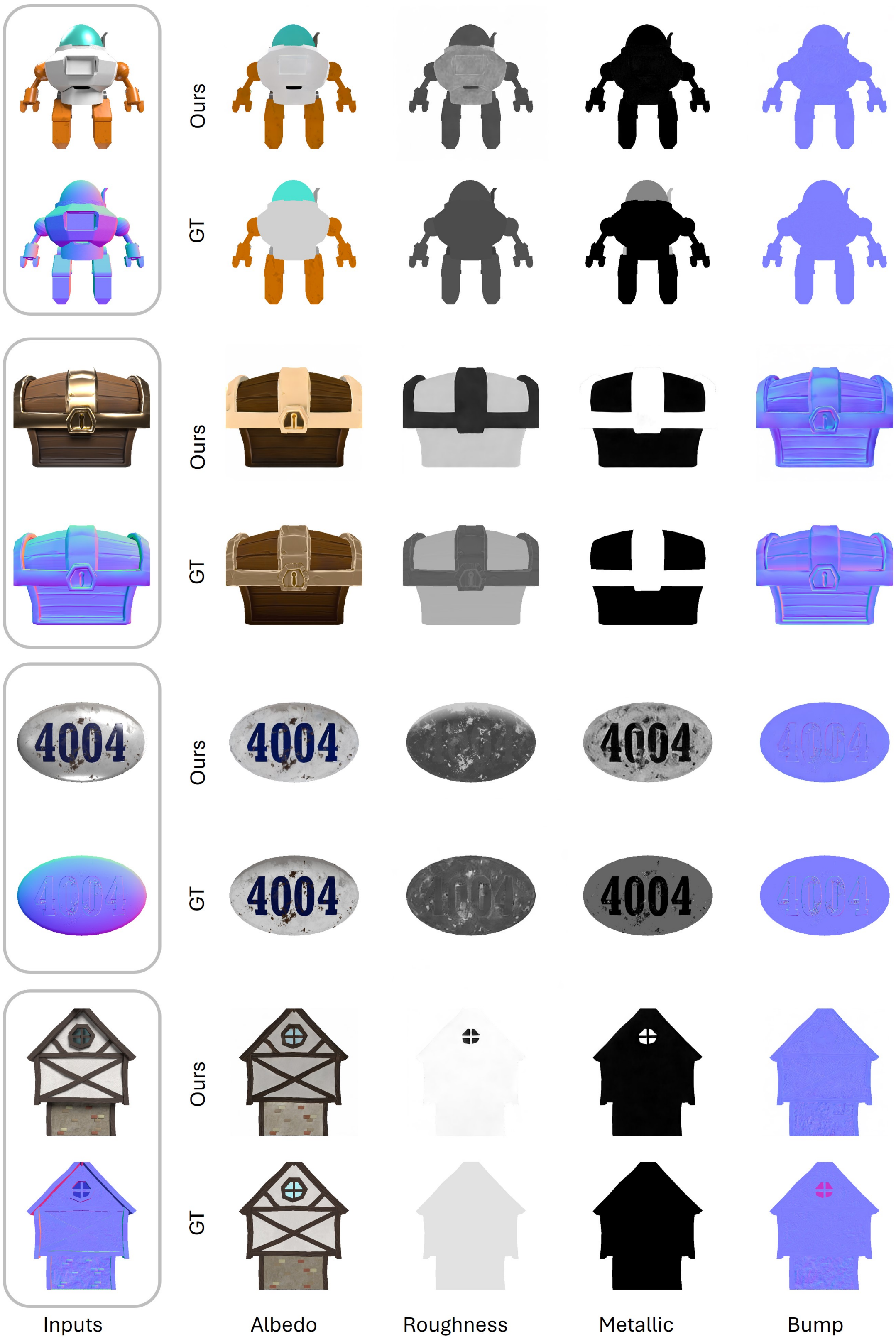


Figure 7. **Results by our material estimator on 2D renderings from Objaverse.**

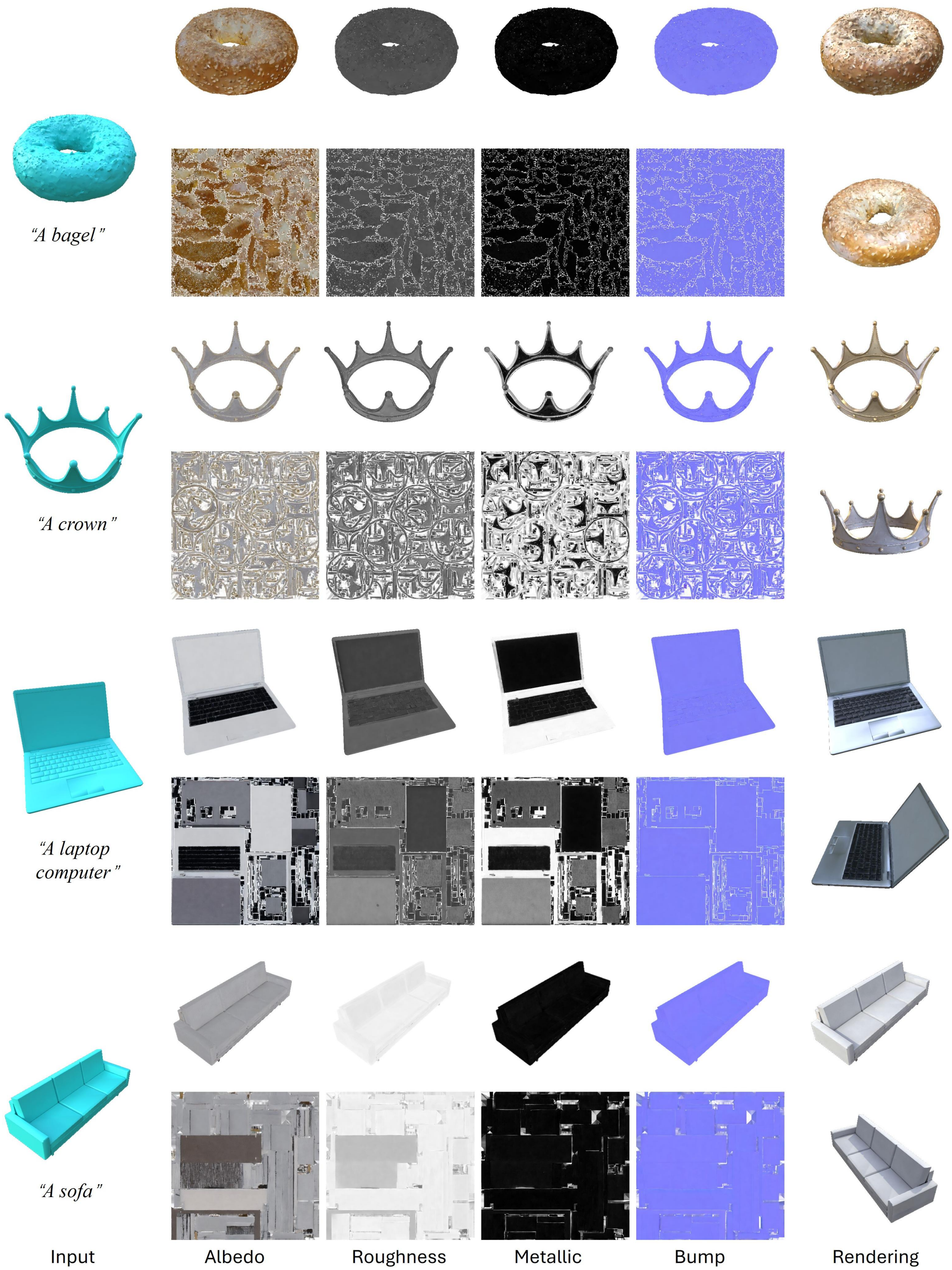


Figure 8. **Additional results by Material Anything on texture-less 3D objects.** The generated UV material maps are provided.

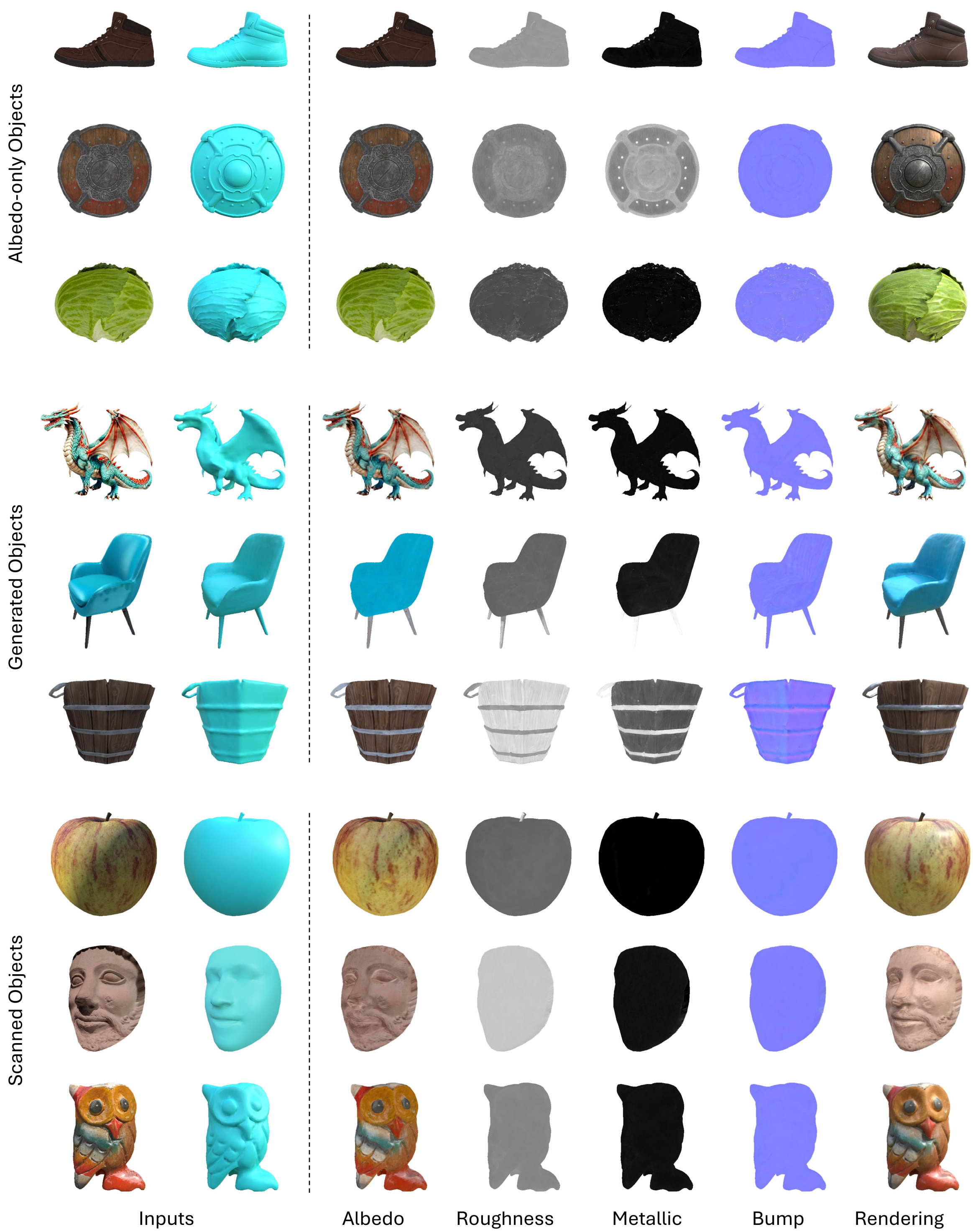


Figure 9. **Additional results by Material Anything on albedo-only, generated, scanned 3D objects.**