

MaterialFusion: Enhancing Inverse Rendering with Material Diffusion Priors

Supplementary Material

001 1. Additional Visualizations

002 We show visualizations of StableMaterial and Material-
003 Fusion’s performance on additional examples from the
004 BlenderVault test dataset as well as the NeRFactor and
005 NeRF synthetic datasets. A webpage containing videos of
006 the material reconstruction and relighting results has also
007 been attached.

008 2. Additional Details

009 **BlenderVault.** To collect BlenderVault, we utilized
010 BlenderProc [2] to download objects from the BlenderKit
011 website. We use Blender in order to render out $30\ 512 \times 512$
012 multi-view images, specifically using Cycles engine with
013 64 SPP. At each rendering operation, we load in the light-
014 ing by randomly selecting between three options with equal
015 probability: StreetLearn environment maps that were re-
016 constructed into HDR maps [5], Laval indoor environment
017 maps, or a Blender light source. If a Blender light source
018 is picked, then either a point light is set up at the camera
019 location with 150W power, or a sun light points codirec-
020 tionally with the camera at the object, with power amount
021 randomly sampled between 10 and 20. To accurately render
022 the albedo and ORM parameters of objects, we tried to ap-
023 proximate them as a Disney principled BRDF model as best
024 as possible, corresponding to the “Base Color” and rough-
025 ness and metalness parameters of the principled BSDF. Due
026 to the diversity of shaders used to represent materials. Some
027 objects had material or albedo that couldn’t be rendered due
028 to either the complexity of the object ($>500\text{MB}$) or due to
029 interference from features such as procedural generation. In
030 such cases we skip the objects and continue to rendering the
031 next one. Overall, around 300 objects were skipped but are
032 still included in the final dataset.

033 **Training Details.** StableMaterial was trained in similar
034 fashion to Zero123, by using a batch size of 1536 with im-
035 ages resized to 256×256 and learning rate of 10^{-4} with an
036 AdamW optimizer for 25k steps. A fully connected layer
037 ($1028 \rightarrow 1024$) that converts the concatenation of the CLIP
038 embedding and pose to a compatible embedding for Stable
039 Diffusion’s UNet was trained with a learning rate of 10^{-3} ,
040 where the pose representation used was similar Zero123’s.
041 Stable Diffusion’s UNet and the fully connected layers were
042 trained and all other components were frozen. Finetuning
043 took 2 days with $8 \times \text{H100 GPUs}$. The filtering keys used
044 for including Objaverse data were [pbr, pbrtexture,
045 substance, substancepainter], while excluding
046 objects that included [style, stylized, cartoon,

lowpoly, poly]. These objects were filtered from the
dataset gathered by [6].

MaterialFusion utilizes the nvdiffrast [4] differentiable
renderer in order to render out the appearance. nvdiffrast
can also render other properties for the system, such as
the albedo, ORM, diffuse and specular lighting, and more.
In particular, we feed the rasterized estimated albedo and
ORM batches to StableMaterial. $\mathcal{L}_{\text{recon}}$ required sRGB
tonemapped image inputs, and we took care to make sure
that the inputs to StableMaterial were converted to RGB.
 \mathcal{L}_{reg} contains a normal, albedo, and ORM smoothness reg-
ularizers, as well as a normal perturbation regularizer that
encourages normal map perturbations. The last term in-
cluded is a demodulated lighting regularization term that
utilizes the rendered specular and diffuse lighting on the
object. These regularizations are kept as they are as part of
MaterialFusion. For more details we refer the reader to the
nvdiffrast paper [3]. We show relighting comparisons for
the rest of the objects used in our evaluation in Figs. 4–6
along with the environment illumination used for rendering
the training data images.

TensoIR Details. Given that TensoIR doesn’t model
metallicness, we added it as an additional parameter to
the appearance representation decoded from the appearance
tensor \mathcal{G}_a . The appearance features are interpolated from
 \mathcal{G}_a and are then decoded with a radiance network \mathcal{D}_c , shad-
ing normal network \mathcal{D}_n , and a material network \mathcal{D}_β to
produce the corresponding representations. Of interest is
the material decoder which we modify to decode the addi-
tional metallicness parameter in addition to the albedo and
roughness. All losses and hyperparameters are kept simi-
lar. Physically-based rendering is then used to render the
resulting image, where metallicness is applied to the diffuse
component of the Cook-Torrance reflectance model [1]:

$$f_r = (1 - m) \frac{\mathbf{a}}{\pi} + f_s \quad (1)$$

where f_s is the specular component of the Cook-Torrance
model.

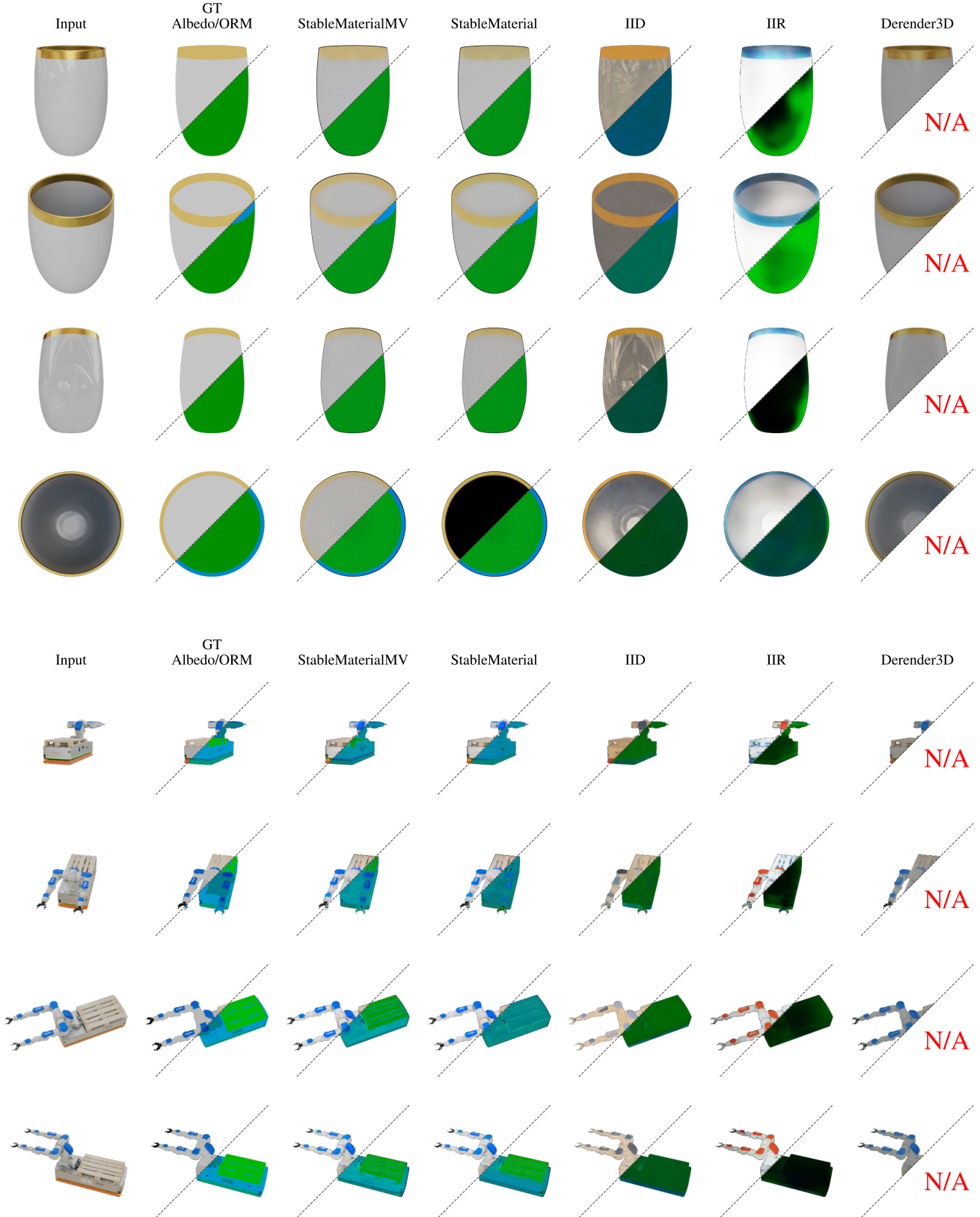
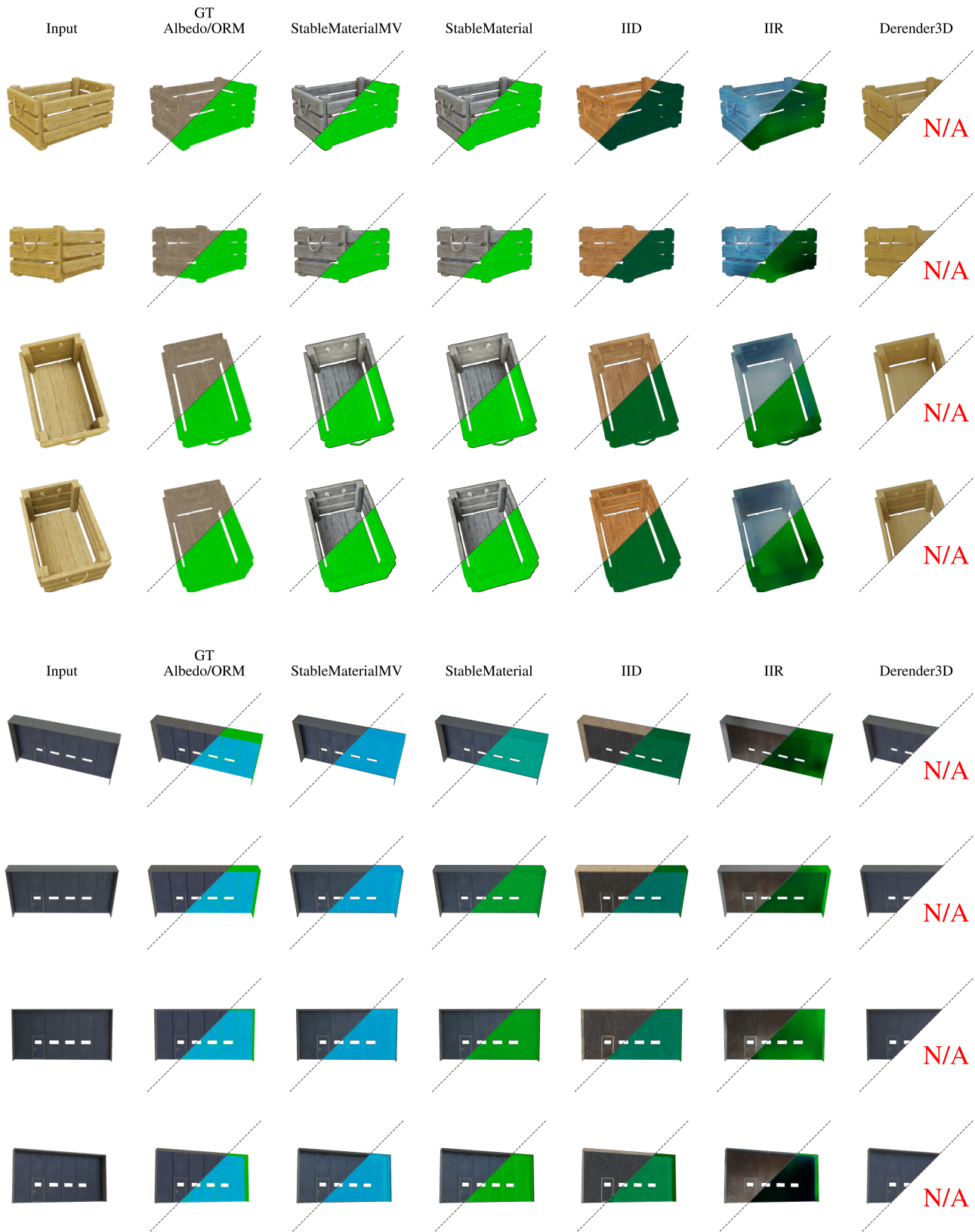


Figure 1. Full albedo and ORM comparison results for StableMaterial on the cup and armatures examples shown in Fig. ??.



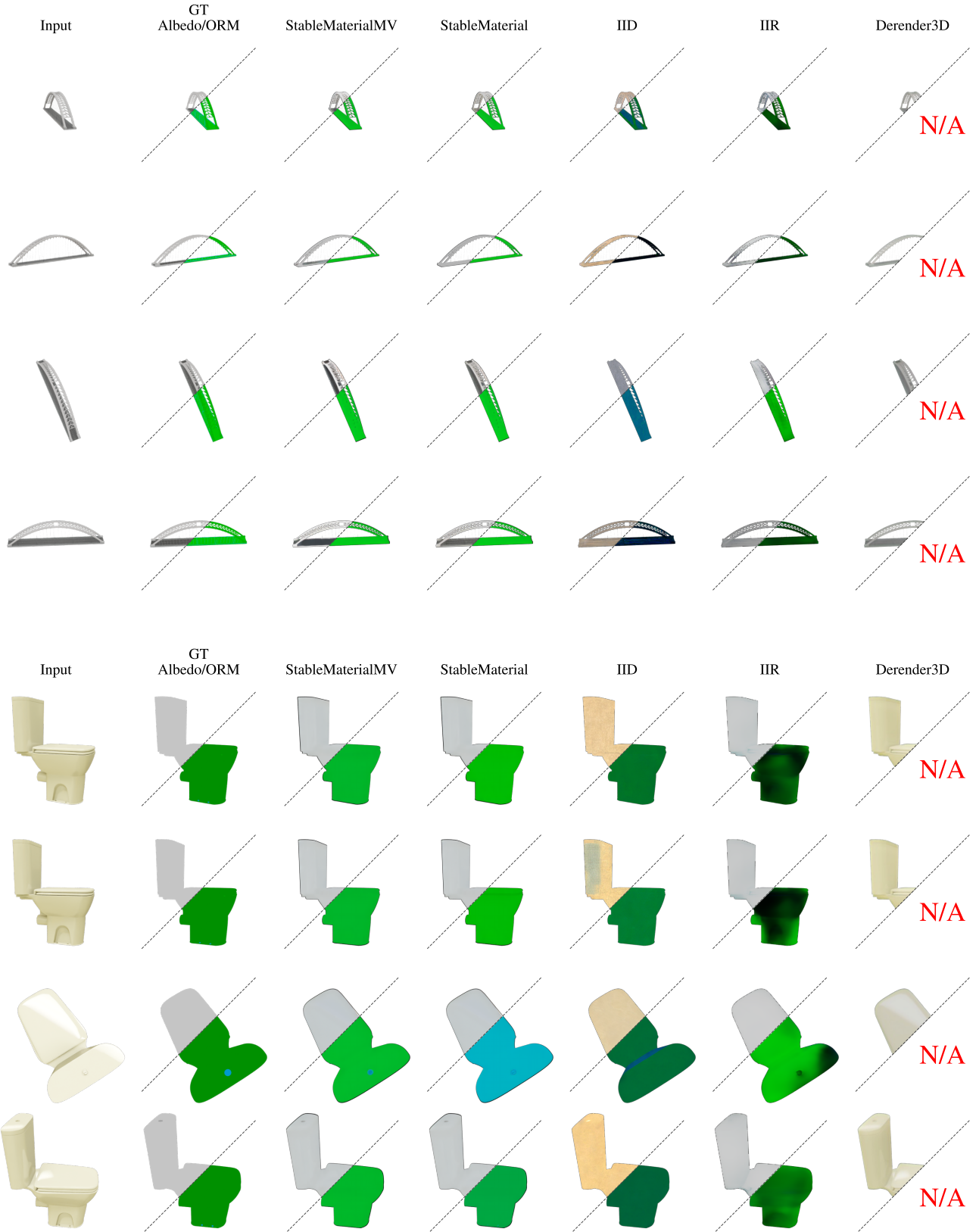


Figure 3. Additional albedo and ORM comparisons for randomly selected examples from the BlenderVault test dataset.

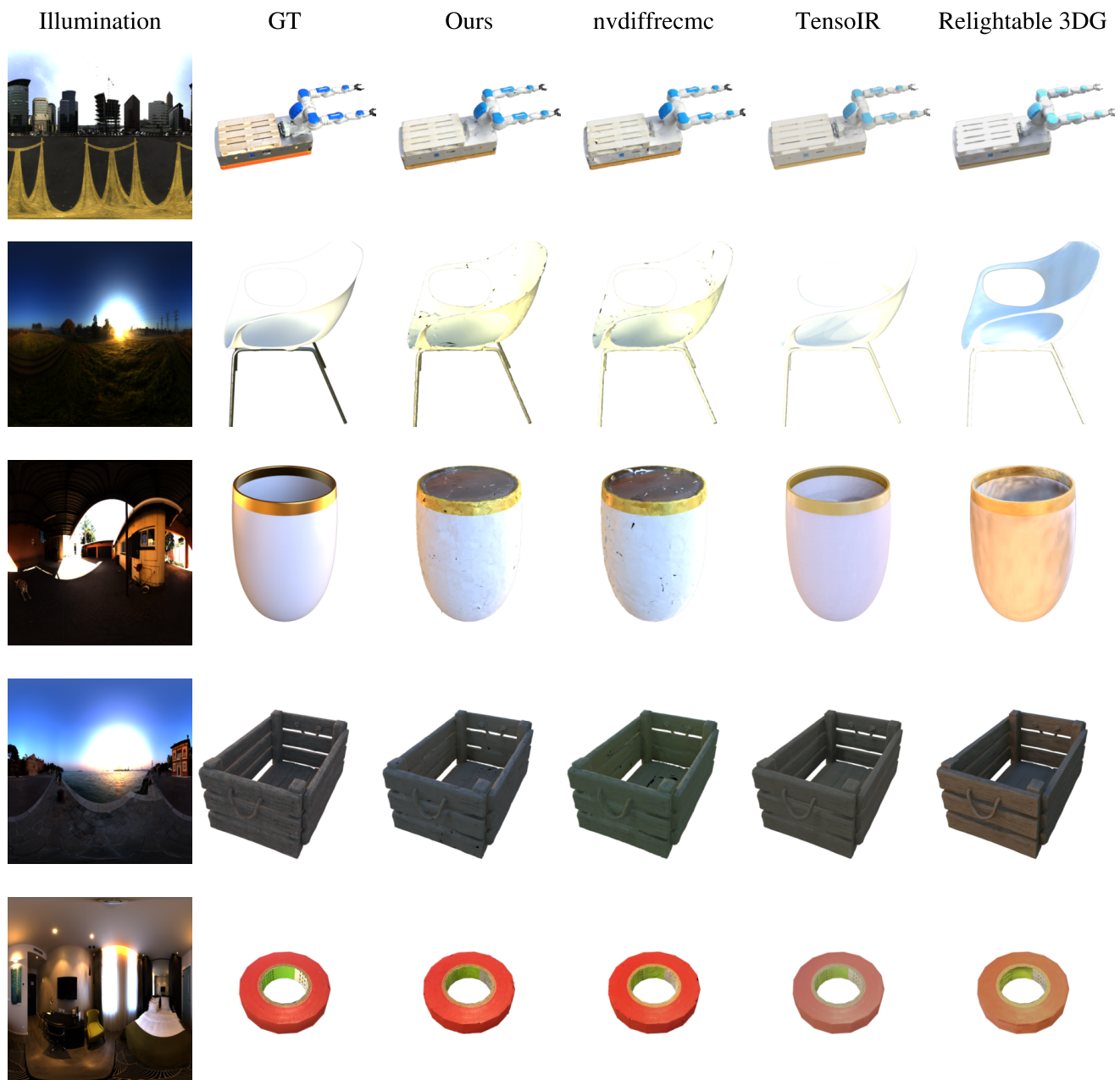


Figure 4. Additional comparisons for MaterialFusion vs. other methods on 3D physical properties reconstruction on more objects from the BlenderVault test dataset.

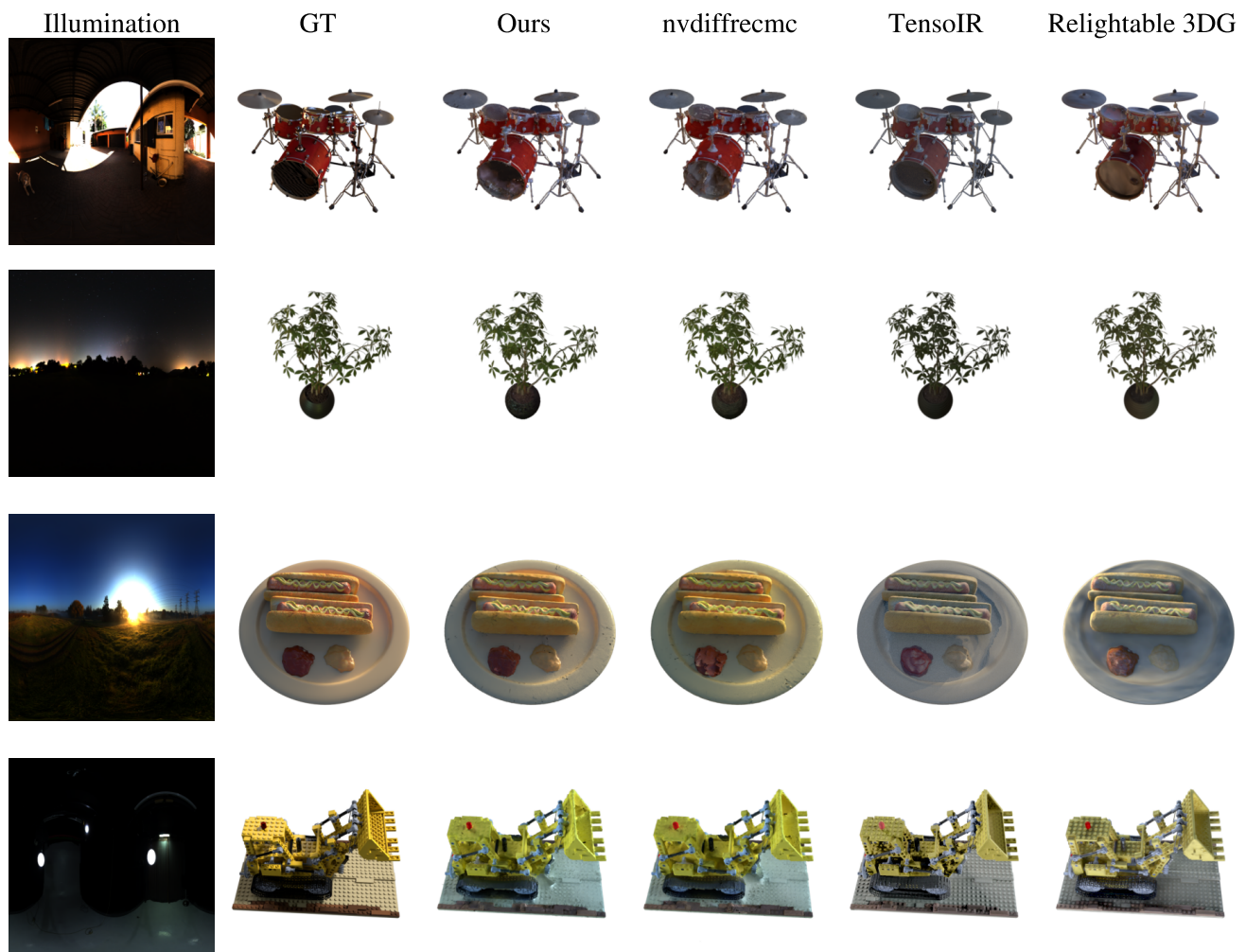


Figure 5. Additional comparisons for MaterialFusion vs. other methods on 3D physical properties reconstruction on the NeRFactor dataset.

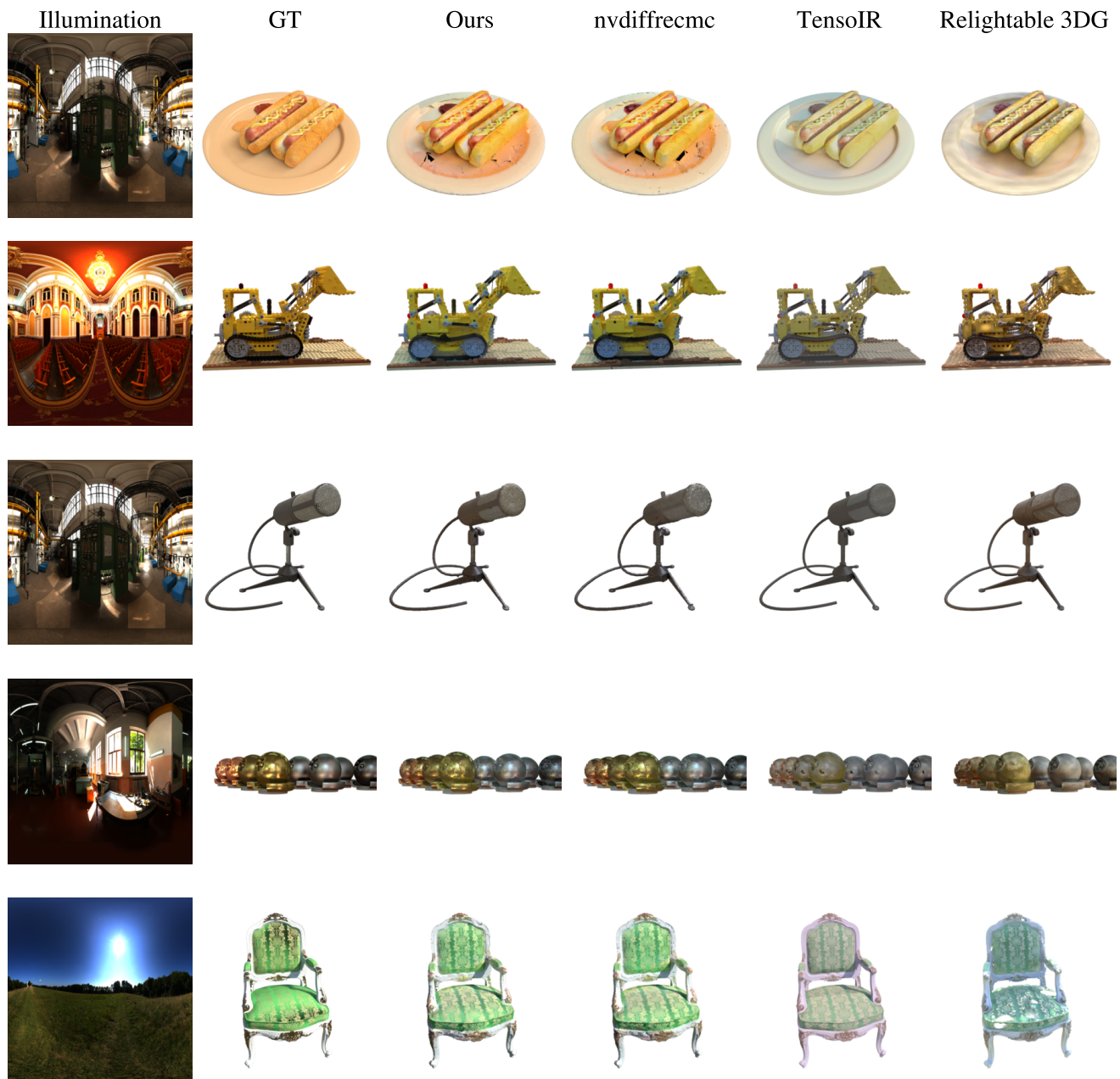


Figure 6. Additional comparisons for MaterialFusion vs. other methods on 3D physical properties reconstruction on the NeRF dataset.

References

- [1] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1): 7–24, 1982. [1](#)
- [2] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. [1](#)
- [3] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. *arXiv:2206.03380*, 2022. [1](#)
- [4] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. [1](#)
- [5] Marcel Santana Santos, Tsang Ing Ren, and Nima Khademi Kalantari. Single image hdr reconstruction using a cnn with masked features and perceptual loss. *ACM Transactions on Graphics*, 39(4), 2020. [1](#)
- [6] Jiayang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation, 2024. [1](#)