Extracting Neural Materials from Multi-view Images

— Supplementary Material —

Acknowledgments. We thank Miloš Hašan, Tizian Zeltner, Saeed Hadadan, and Blaire (Yunchen) Yu for helpful discussions and sharing code. We also thank Aaron Lefohn and Chris Wyman for supporting this research.

A. Implementation Details

A.1. Large Material Reconstruction Model

Multi-view input encoding. We encode $F = 17$ views, evenly spaced in a 360 degree orbit around the object, alongside the six canonical views. The 17 views are encoded using the frozen Wan2.1 VAE encoder, and the canonical views are each encoded with the image VAE for additional quality [8]. The input views (F orbital + six canonical) are encoded into a latent tensor, $\mathbf{z}^{\mathbf{I}} \in \mathbb{R}^{(1+(F-1)/4+6)\times 16\times H/8\times W/8}$. We use $H = W = 512$ for the input image resolution. By using the Wan2.1's video VAE encoder, all the orbital input views are processed jointly, allowing the subsequent LMRM backbone to aggregate cross-view information.

Backbone. The backbone architecture of LMRM is a diffusion transformer (DiT) [5], initialized from the pre-trained text-to-video Wan2.1-1.3B [7], repurposed as a *single-step* model. The backbone performs a single deterministic forward pass at a fixed timestep t=0, rather than iterative denoising, so that the DiT acts as a feed-forward regressor that takes multi-view images and predicts triplane feature planes. As the reconstruction task is not text-conditioned, we bypass the text cross-attention by using a fixed null-text embedding to every layer. We keep the original Wan2.1-1.3B configuration: 30 transformer blocks with model width 1536, 12 attention heads, feed-forward inner dimension 8960, patch embedding (1×2×2), and 3D rotary position embeddings [6].

Triplane decoding head. After the encoded input latent tensor, $\mathbf{z}^{\mathbf{I}}$, get processed with the DiT backbone, the output tokens are mapped to triplane feature planes through a lightweight linear head followed by a convolutional decoder initialized from the pre-trained Wan2.1 VAE decoder (kept trainable). We decode the latent frames and obtain three triplane features: $\mathbf{T}_{\mathrm{XY}}, \mathbf{T}_{\mathrm{YZ}}, \mathbf{T}_{\mathrm{XZ}}$ representing the XY, YZ, and XZ planes respectively, each has 12 channels. We decode each feature plane independently to avoid the causal temporal coupling of the video VAE decoder.

Triplane MLPs. Given a 3D surface point we bilinearly sample the three axis-aligned planes and concatenate the resulting C=12-channel features (3C=36 total). We further append a NeRF-style positional encoding [4] (3 frequency bands) for 3D surface position and decode the material parameters with a triplane MLP, *i.e.*, 4-layer MLP of width 64 and intermediate ReLU activations, with the final sigmoid. For the PBR pre-training stage, the material MLP head predicts 5 channels = base color (3D) + roughness (1D) + metallicity (1D). For the neural material fine-tuning stage, the material MLP head predicts 9 channels = base color (3D) + neural specular latents (6D). A parallel uncertainty MLP head, implemented as a separate MLP of identical capacity, predicts a per-channel log-variance used in a β negative log-likelihood (β-NLL) objective (Eq. 9). The MLPs are evaluated only at surface points via a gather-scatter scheme.

Training. We train the LMRM for 200K steps on 32 NVIDIA A100 GPUs with a per-GPU batch size of 2 in `bfloat16`, using AdamW with a cosine learning rate scheduler. For learning rates, we use 5×10^{-5} for the backbone, and 5×10^{-4} for both the material and uncertainty triplane MLPs. The β-NLL loss is disabled for the first 30K steps and then linearly ramped over 10K steps. Losses are computed on 8 randomly sampled views per object.

A.2. Test-time Optimization

In this stage, we optimize only the LMRM-initialized triplanes (the triplane MLPs are kept frozen) using Adam with a learning rate of 5×10^{-3}. For the uncertainty-guided material regularization, we take a single snapshot before optimization: we evaluate the frozen triplane MLPs to cache the initial material prediction and the per-channel aleatoric uncertainties. We keep the cache frozen throughout the optimization. We set $\lambda_{\mathrm{reg}} = 2.0$, chosen from a sweep over $\{0.1, 0.25, 0.5, 1, 2\}$ that best balanced rendered-image and material-decomposition quality. Test-time optimization takes roughly 15 minutes per object (1000 steps at 512×512) on a single NVIDIA A100 GPU, excluding final result rendering. We use the Blender AgX tonemapper for optimization and for all rendering results in the paper (including additional Figs. S1, S2, & S3), unless mentioned otherwise.

A.3. NeuMatEx for Real-world Photos

We provide additional real-world results from the DTC [1] dataset in Figure S4 and Figure S5. For all of these experiments, we use the captured mesh and HDR environment map, together with the posed LDR images. A constant tonemapping operator for the rendered images during our test-time optimization can introduce a domain gap with the real-world observations. To avoid this, we follow IRIS [3] and use a learnable camera response function (CRF) for this real-world experiments. We use the parametric CRF space of IRIS [3], based on Grossberg and Nayar's empirical model of response (EMoR) [2], which fits a PCA basis to real-world measured CRFs. The optimized CRFs are then used to tonemap the rendered images for the comparisons as well as for the relightings.

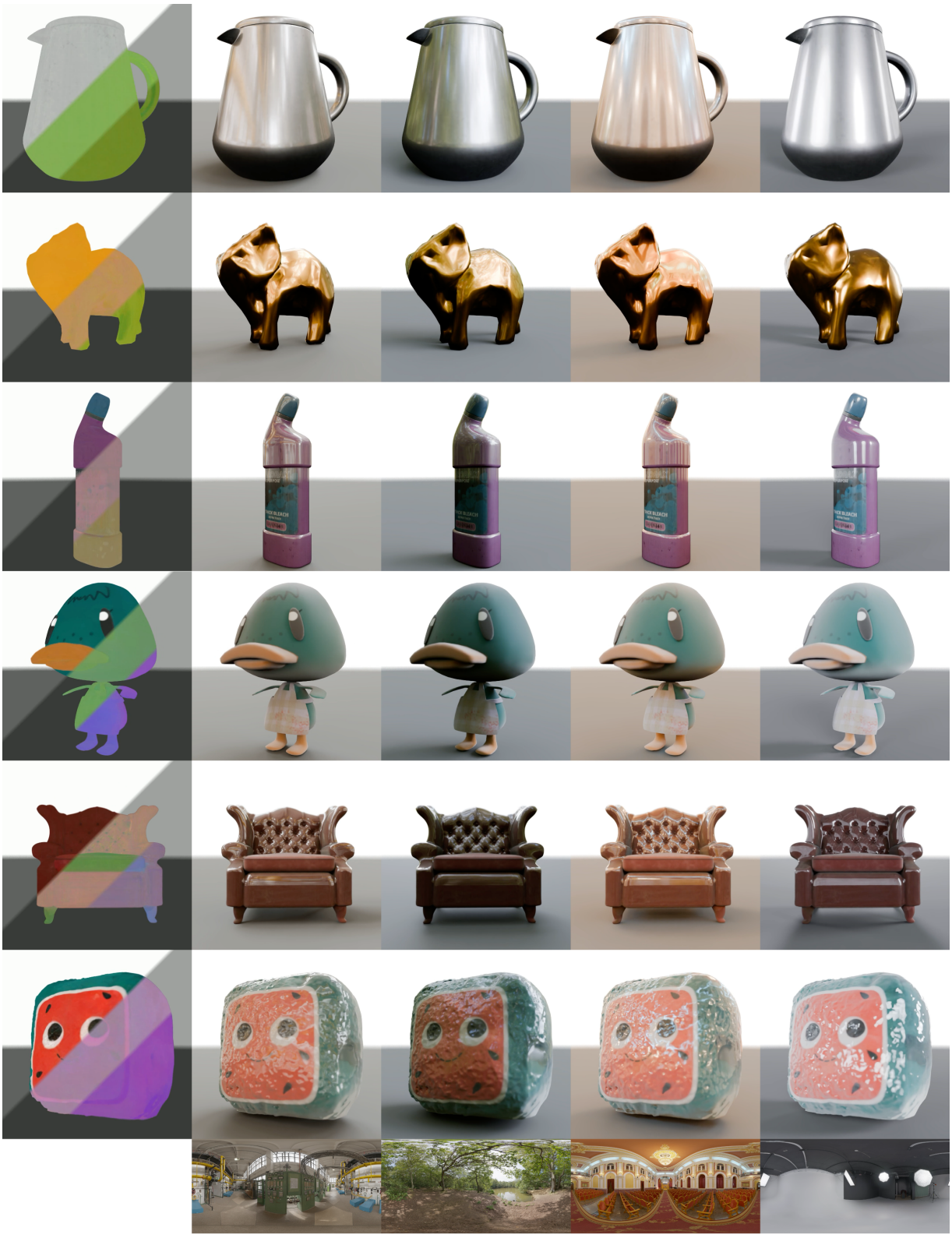

Figure S1. **Visualization of the extracted neural materials.** Column 1 shows the decomposed neural materials (diffuse base color + neural specular latents A & B), and columns 2-4 shows the 128spp relit renderings using four different HDR light probes.

Figure S2. **Visualization of the extracted neural materials.** Column 1 shows the decomposed neural materials (diffuse base color + neural specular latents A & B), and columns 2-4 shows the 128spp relit renderings using four different HDR light probes.

Figure S3. **PBR vs. Neural Material (128 spp, relit).** Insets: extracted diffuse base color (all columns, bottom), probes for relighting (last column, top). All methods use known fixed geometry. We include Hunyuan3D-2.1 and TRELLIS.2 to highlight the limitations of PBR, and do not represent a fair comparison to optimization methods (NVDiffRecMC++, NeuMatEx), which use known poses and lighting.

Figure S4. **Comparison for real-world neural material extraction.** Additional results on the DTC [1] dataset. Our method extracts more expressive neural materials than the PBR-based optimization method NVDiffRecMC++, capturing complex specular reflections.

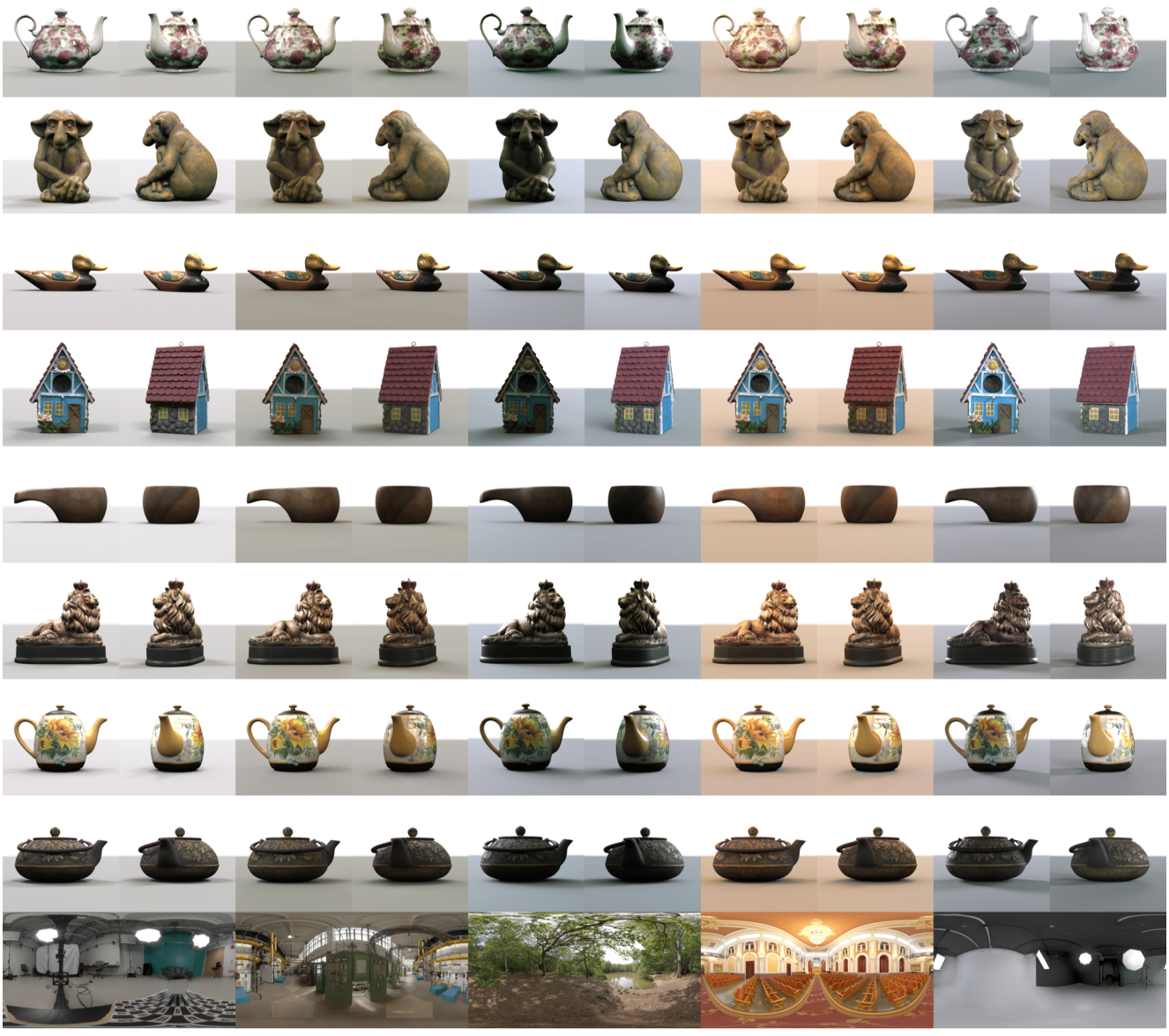

Figure S5. **Relighting real-world neural materials.** Relighting results showing two views of multiple objects from the DTC [1] dataset. Our extracted neural materials generalize to novel lighting conditions producing complex specular effects.

References

[1] Zhao Dong, Ka Chen, Zhaoyang Lv, Hong-Xing Yu, Yunzhi Zhang, Cheng Zhang, Yufeng Zhu, Stephen Tian, Zhengqin Li, Geordie Moffatt, Sean Christofferson, James Fort, Xiaqing Pan, Mingfei Yan, Jiajun Wu, Carl Yuheng Ren, and Richard Newcombe. Digital Twin Catalog: A Large-Scale Photorealistic 3D Object Digital Twin Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 5, 6

[2] Michael D. Grossberg and Shree K. Nayar. Modeling the space of camera response functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(10): 1272–1282, 2004. 1

[3] Chih-Hao Lin, Jia-Bin Huang, Zhengqin Li, Zhao Dong, Christian Richardt, Tuotuo Li, Michael Zollhöfer, Johannes Kopf, Shenlong Wang, and Changil Kim. IRIS: inverse rendering of indoor scenes from low dynamic range images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1

[4] 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. In *European Conference on Computer Vision (ECCV)*, 2020. 1

[5] William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 1

[6] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint, 2104.09864*, 2021. 1

[7] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint, 2503.20314*, 2025. 1

[8] 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 *European Conference on Computer Vision (ECCV)*, 2024. 1