Neural LightRig: Unlocking Accurate Object Normal and Material Estimation with Multi-Light Diffusion

Supplementary Material

A. Dataset Details

In the main paper, we provided an overview of the *LightProp* dataset, designed specifically to address the challenges of learning robust multi-light image generation and geometry-material estimation. Here, we detail the data curation and rendering configurations.

A.1. Data Curation

Objaverse [2] originally contains around $800,000$ synthetic objects across various categories and styles. To ensure high-quality content for *LightProp*, we implemented a rigorous curation process. First, we filtered out objects with extreme thinness or unbalanced proportions, such as objects with large surface areas but minimal thickness or depth, which often distort lighting interactions and hinder effective learning. Additionally, we excluded objects that originated from 3D scans or those representing entire scenes, as these typically contain irrelevant environmental details that are less suitable for our framework. Finally, objects lacking essential PBR material maps (albedo, roughness, and metallic maps) were removed to ensure comprehensive material data for training. This selection process resulted in a refined subset of around $80,000$ high-quality objects for *LightProp*.

A.2. Rendering Setup

The *LightProp* dataset is created using the Cycles rendering engine in Blender [1], with each image generated at 128 samples per pixel and accelerated using CUDA. To introduce diversity in object orientation and perspective, each object is rendered from five distinct viewpoints: a front view, a right view, a top view, and two random views sampled on a surrounding sphere. For each viewpoint, we apply five distinct lighting conditions, comprising a point light, an area light, and three HDR environment maps randomly selected from 25 high-quality maps. To set up our directional lighting, we position eight lights around the camera and place one additional light directly at the camera's position. The lighting orientations are parameterized by spherical coordinates θ and φ, specifically configured as:

$$\theta_i = i \cdot \frac{\pi}{4} \quad \text{for } i = 0, 1, \ldots, 8, \tag{1}$$

$$\varphi_i = \{1, 2, 1, 2, 1, 2, 1, 2, 0\} \cdot \frac{\pi}{6}. \tag{2}$$

This arrangement ensures diverse lighting directions to enhance shading and reflectance variations in multi-light images, which are essential for accurate geometry and material estimation. In addition to the multi-light images, each object view is paired with ground-truth G-buffer maps, including surface normals, albedo, roughness, and metallic maps. These G-buffers, rendered via Blender's physically-based pipeline, provide the necessary supervision for training in surface normal and PBR material prediction.

B. Implementation Details

B.1. Multi-Light Diffusion

We build our multi-light diffusion model on top of Stable Diffusion v2-1[1]. As discussed in the main paper, we adopt a two-phase training scheme to adapt this pre-trained model for multi-light image generation. In the initial phase, we tune the first convolution layer, all parameters in the self-attention layers, and only the key and value parameters in the cross-attention layers. This phase runs for $80,000$ steps with a peak learning rate of 1×10^{-4} and a total batch size of 128, following a cosine annealing schedule with $2,000$ warm-up steps. We use the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.01, and enable bf16 mixed precision to accelerate the training. Additionally, we apply gradient clipping with a maximum norm of 1.0 to stabilize training and incorporate classifier-free guidance, with a probability of dropping the conditioning set to 0.1. In the following phase, we further fine-tune the full model for another $80,000$ steps at a significantly lower peak learning rate of 5×10^{-6} with the same training particulars. Both of the two phases are trained with an input image resolution of 256×256, and a multi-light output of 768×768. In total, the complete training process of our multi-light diffusion model takes approximately 2.5 days on 32 NVIDIA A100 (80G) GPUs.

B.2. Large G-Buffer Prediction Model

Architecture. Our large G-buffer prediction model takes as input a single image with 4 channels (including alpha), combined with multi-light images comprising 9 lighting conditions, each with 3 channels, resulting in a total of $4+9\times3 = 31$ input channels. The output consists of 8 channels, representing the surface normals, albedo, roughness, and metallic maps (3, 3, 1, and 1 channel, respectively). The regression U-Net architecture comprises four down-sampling blocks with progressively increasing channels of 224, 448, 672, and 896, followed by a bottleneck block with 896 channels, and then four up-sampling blocks with

[1] https://huggingface.co/stabilityai/stable-diffusion-2-1

correspondingly decreasing channels of 896, 672, 448, and 224. Each block contains two residual layers with Group Normalization (using 32 groups), and SiLU activation. Attention mechanisms, implemented in a pre-norm style , are applied in all but the first down-sampling block and the last up-sampling block, using an attention head dimension of 8. Within each block, up-sampling and down-sampling are performed via a convolutional layer placed after the two residual layers. To encode the spherical coordinates $\{\theta^i, \varphi^i\}$ associated with each lighting condition, we employ sinusoidal embeddings. Each scalar θ or φ is projected to a higher dimension of $d_{scalar} = 224$ and we concatenate these projected vectors into a single $9 \times 2 \times 224 = 4032$ dimensional vector, which is subsequently embedded by a 2-layer MLP, producing an illumination embedding with a final dimensionality of $d_{emb} = 896$. This embedding is modulated to each block in the U-Net with adaptive group normalization. For the smaller models in our ablation study, we use a U-Net with down-sampling blocks at 128, 256, 384, and 512 channels, mirrored in the up-sampling blocks, along with a 512-channel bottleneck block.

Training Details. We apply weighted loss contributions to balance $\mathcal{L}_{\text{normal}}$ and $\mathcal{L}_{\text{PBR}}$. Specifically, we set a $4:1$ ratio for surface normals relative to PBR materials. Additionally, we apply a stabilization factor of $\lambda_1 = 0.25$ for the MSE term in $\mathcal{L}_{\text{normal}}$, as outlined in the main paper. Given the computational demands of high-resolution feature maps, especially with attention layers, we employ a two-phase training strategy, gradually transitioning from low to high resolutions. In the initial phase, we train at a resolution of 256×256 to establish core feature representations, running for $60,000$ steps with a batch size of 128. This phase includes $1,500$ warm-up steps, a peak learning rate of 1×10^{-4}, and a weight decay of 0.01, using a cosine annealing schedule and the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Training on 32 NVIDIA A100 (80G) GPUs, this phase completes in approximately 20 hours. Following this foundational phase, we move to a higher resolution of 512×512, allowing the model to capture finer details essential for precise geometry and material predictions. This fine-tuning phase involves a reduced learning rate of 2×10^{-5} and runs for an additional $30,000$ steps on the same setup of 32 NVIDIA A100 (80G) GPUs, completing in approximately 7 days. All other training parameters are kept consistent with the initial phase.

Augmentation Details. In the main paper, we introduced the augmentations to bridge the gap between our multi-light diffusion model and the large G-buffer prediction model. For *Random Degradation*, we down-sample each multi-light image to a lower resolution uniformly sampled from $\mathcal{U}(128, 256)$ and then up-sample it back to the original resolution of 256. Following this, we apply grid distortion with a perturbation strength sampled from $\mathcal{U}(0.15, 0.3)$ to simulate geometrical misalignments. For *Random Intensity*, we convert the multi-light images to HSV format and adjust the brightness channel using an image-level scaling factor from $\mathcal{U}(0.9, 1.3)$. Additionally, we apply pixel-level noise by scaling each pixel independently with a factor sampled from $\mathcal{N}(1, 0.05)$. The input image receives a separate brightness adjustment factor sampled from $\mathcal{U}(0.9, 1.1)$. For *Random Orientation*, all spherical coordinates are perturbed by an angular gaussian noise in radians. θ^i receive a noise sampled from $\mathcal{N}(0, 0.1)$ and are wrapped with modulus 2π. φ^i are perturbed with noise from $\mathcal{N}(0, 0.02)$ and clamped within $[0, \frac{\pi}{2}]$. The above three augmentations are triggered independently with a probability of 0.6. For *Data Mixing*, this augmentation is applied with a probability of 0.3. We generate multi-light images from our diffusion model with a classifier-free guidance scale of 2.0 over 75 inference steps. Additionally, inspired by prior work on multi-view reconstruction [3], we shuffle the order of the multi-light images during training with a probability of 0.5 to encourage robustness in learning features across varied lighting sequences, thereby reducing dependency on any specific lighting arrangement.

Figure 1. Failure case.

C. Limitations

While our approach demonstrates strong performance, several limitations remain. First, for input images with extreme highlights or shadow areas, our method struggles to fully remove illumination effects in the predicted albedo maps, as shown in Fig. 1. Additionally, the resolution of the backbone multi-light diffusion model (256×256) limits the level of detail achievable in the generated multi-light images, subsequently constraining the final normal and material predictions. Increasing the model's resolution could enhance the quality of the predicted surface properties. Finally, our method is currently designed for objects rather than full scenes, limiting its applicability in complex, multi-object environments.

D. Additional Results

D.1. Our Results

Figs. 2 and 3 present examples of our full pipeline output, including input images, generated multi-light images, estimated surface normals, PBR materials, and relit images under various environment maps. These results showcase the

robustness of our approach in generating consistent geometry and material estimates and realistic relighting effects across different lighting conditions. Additionally, Figs. 4 and 5 showcase extended single-image relighting results of our method under an even broader range of environment maps, further highlighting the model's ability to generate high-quality, adaptable relit images across diverse lighting setups. These results illustrate the robustness in managing various lighting conditions and further demonstrate the efficacy of our approach.

D.2. Comparison Results

In Fig. 6, Fig. 7, Fig. 8, and Fig. 9 we offer more comparison results for surface normal estimation, PBR material estimation, and single-image relighting. These comparisons further demonstrate the advantages of our method over baseline approaches in accurately capturing surface details, material properties, and producing realistic relit images under diverse lighting conditions.

References

[1] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 1

[2] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 1

[3] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *The Twelfth International Conference on Learning Representations*, 2024. 2

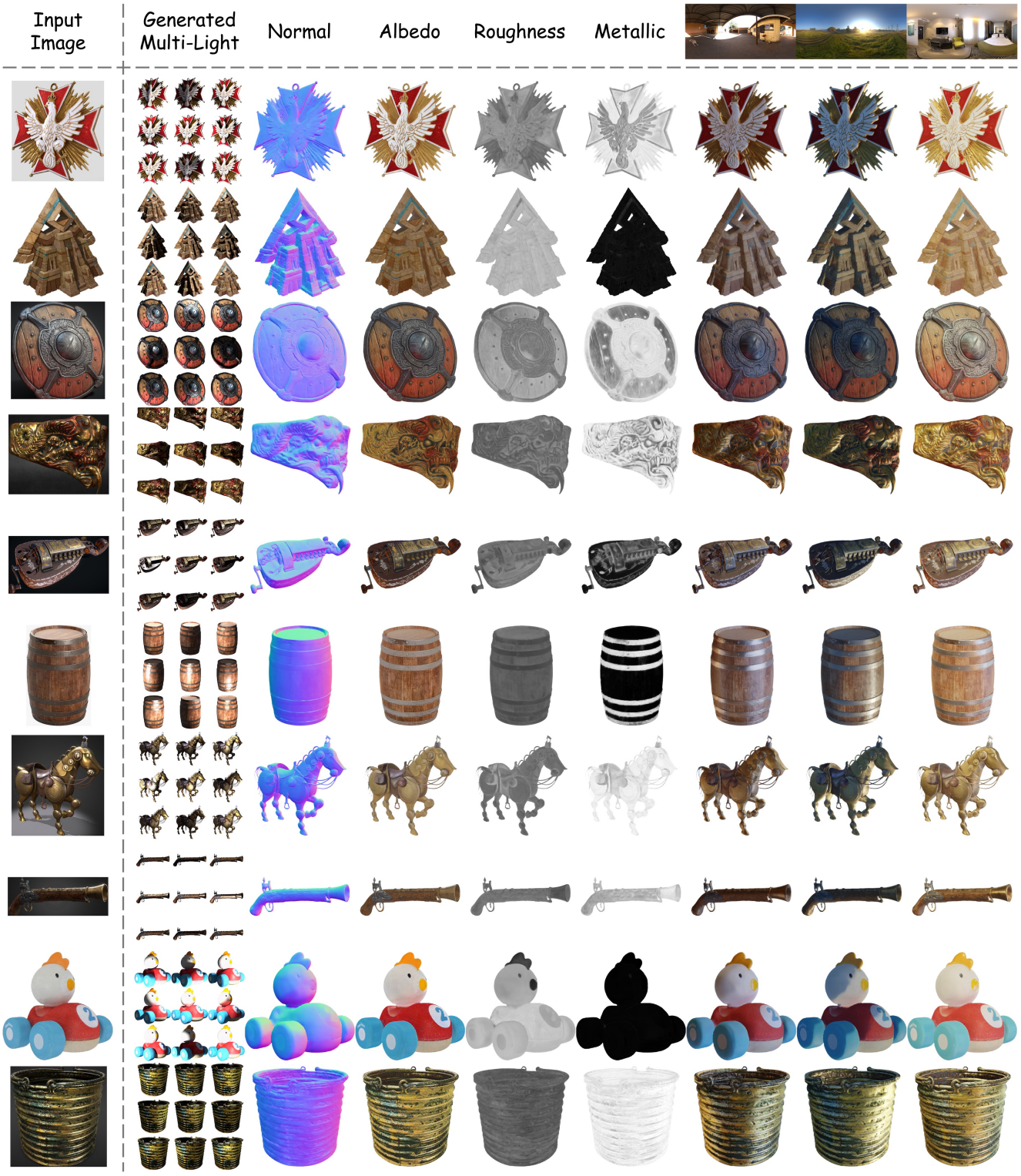

Figure 2. More results of our method.

Figure 3. More results of our method.

Figure 4. More single-image relighting results of our method.

Figure 5. More single-image relighting results of our method.

Figure 6. More comparisons on surface normal estimation.

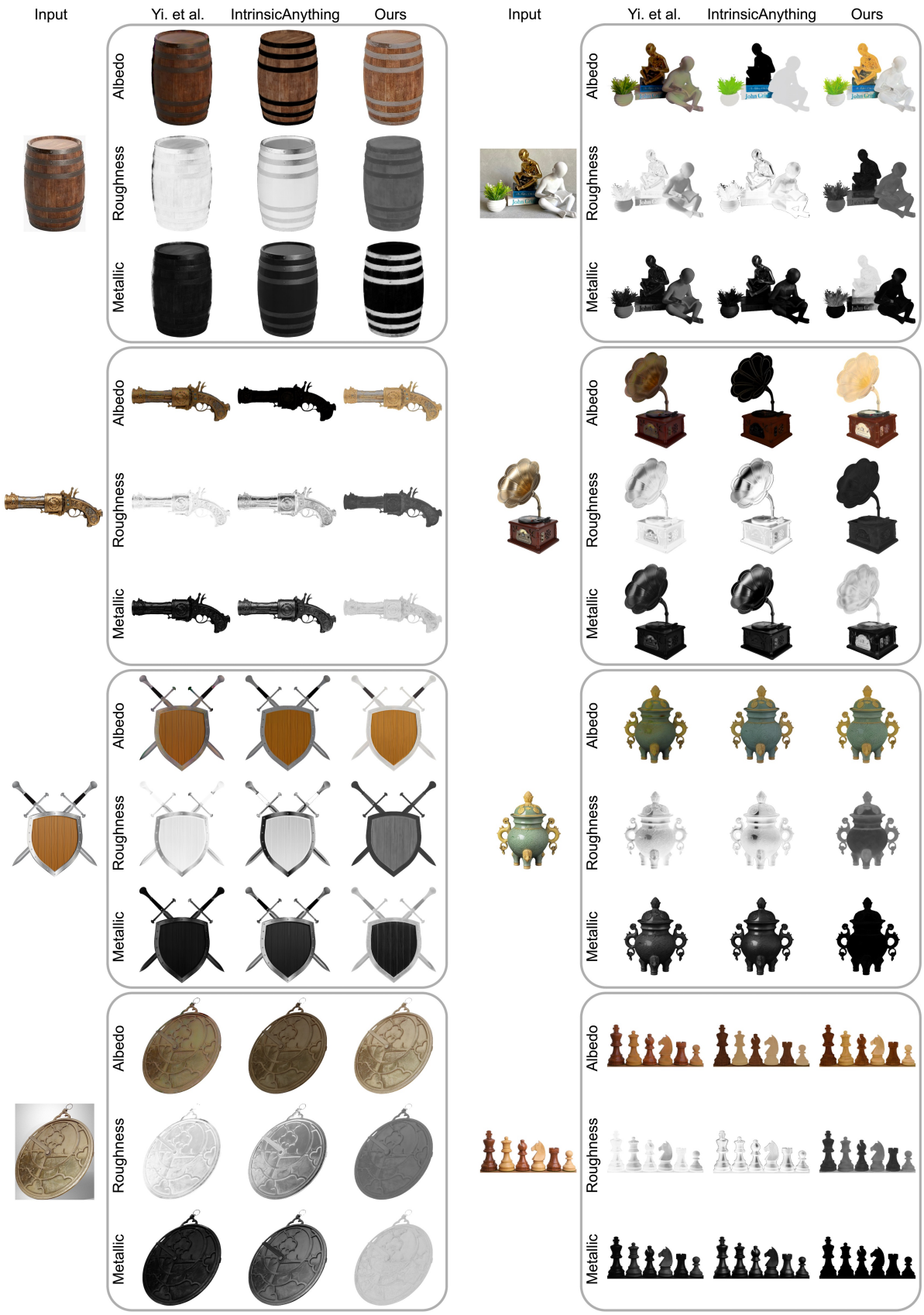

Figure 7. More comparisons on PBR material estimation.

Figure 8. More comparisons on PBR material estimation.

Figure 9. More comparisons on single-image relighting.