Supplementary Material: Chord: **Ch**ain **o**f **R**endering **D**ecomposition for PBR Material Estimation from Generated Texture Images

ZHI YING*, Ubisoft La Forge, China
BOXIANG RONG*, ETH Zürich, Switzerland
JINGYU WANG, ETH Zürich, Switzerland
MAOYUAN XU, Ubisoft La Forge, China

A Appendix

A.1 Training Details

A.1.1 Texture RGB Generation. We trained our Texture RGB Generation model using SDXL [Podell et al. 2024] pre-trained weights, fine-tuning it on 1,000 high-quality texture rendering images rendered by $\mathcal{R}$ with fixed top-down view camera and top-left directional lighting settings. Each image was manually captioned to describe material type, color, patterns, and visual effects, with captions consistently starting with *"texture of"*. The model was trained on 1k-resolution with a batch size of 4, using circular padding for all convolutional layers. Optimization was performed using the Lion optimizer [Chen et al. 2024] with a learning rate of 4×10^{-7}, while the text encoders were trained with a learning rate of 2×10^{-7} and frozen after epoch 35 of the 100-epoch training. The entire process took approximately 18.5 hours on an RTX A6000 GPU.

A.1.2 Material Estimation. Our model builds on pre-trained SD 2.1 weights [Rombach et al. 2021]. The training and validation datasets include 5,789 public materials from MatSynth [Vecchio and Deschaintre 2024] and 1,297 private materials. After 4-way rotational augmentation, we obtain 28,344 samples at 1024×1024 resolution. During training, inputs are randomly downscaled by a factor of two with a probability of 0.2; otherwise, they are randomly cropped to 512×512 with a probability of 0.8. Texture RGB images are rendered as $I_{\text{RGB}} = \mathcal{R}(\text{MAT}; l)$, with l randomly rotated per iteration.

The training process consists of two phases. The optional Pre-training Phase uses standard v-prediction diffusion training to warm up model weights, with timestep t uniformly sampled from $[1, T]$. We adopt velocity parameterization [Salimans and Ho 2022] with zero-SNR [Lin et al. 2024], and freeze all VAE weights as training operates in latent space. The Single-step Phase is the main stage, using a fixed timestep $t = T$ and our chained prediction pipeline. During pretraining, we decay the learning rate exponentially from 1×10^{-4} to 1×10^{-5} over 5 epochs with batch size 8. In the single-step phase, we fix the U-Net learning rate at 3×10^{-5} and the VAE decoder at 1×10^{-6}, training for 20 epochs with batch size 3. Total training time was approximately 5.2 days on an RTX H100 GPU.

*Both authors contributed equally to the paper

Authors' Contact Information: Zhi Ying, zhi.ying@ubisoft.com, Ubisoft La Forge, Shanghai, China; Boxiang Rong, borong@student.ethz.ch, ETH Zürich, Zürich, Switzerland; Jingyu Wang, jingyuwang@student.ethz.ch, ETH Zürich, Zürich, Switzerland; Maoyuan Xu, mao-yuan.xu@ubisoft.com, Ubisoft La Forge, Chengdu, China.

A.2 Light Direction Search

In the Roughness & Metalness prediction step of our Chord pipeline, the renderer $\mathcal{R}$ requires an estimated lighting direction $\mathbf{l}^*$. Assuming a single directional light source, we adopt a simple yet effective strategy to estimate its direction based on observed irradiance. Empirically, light energy tends to decay along the direction of the light source. Thus, we estimate the lighting direction by identifying the axis along which irradiance decays most rapidly. See Alg. 1.

ALGORITHM 1: Light Direction Estimation from Irradiance Image

Input: RGB irradiance image $\mathbf{I}_{\text{IRR}}$, convergence threshold ϵ, iterative search sample count N
Output: Estimated direction $\mathbf{l}^*$
$\mathbf{I}_{\text{IRR}} \leftarrow$ `preprocess`($\mathbf{I}_{IRR}$); `// Convert to grayscale, apply noise reduction`
$[\theta_{\min}, \theta_{\max}] \leftarrow [0, 2\pi]$;
while $\theta_{\max} - \theta_{\min} > \epsilon$ **do**
  $\{\theta_1, \ldots, \theta_N\} \leftarrow$ `uniformly_sample`($[\theta_{\min}, \theta_{\max}], N$);
  $\{\text{diff}_1, \ldots, \text{diff}_N\} \leftarrow$ `compute_irradiance_difference`($\mathbf{I}_{IRR}$, $\{\theta_1, \ldots, \theta_N\}$);
  $k \leftarrow \underset{i \in [1,N]}{\text{argmax}}\ \text{diff}_i$;
  $\theta^* \leftarrow \theta_k$;
  $[\theta_{\min}, \theta_{\max}] \leftarrow [\theta_{k-1}, \theta_{k+1}]$; `// Update search interval`
end
$\mathbf{l}^* \leftarrow [\cos\theta^*, \sin\theta^*, \sqrt{0.5}]$;
return $\mathbf{l}^*$;

Our method estimates the dominant light direction by analyzing directional irradiance contrast in the preprocessed image. In the **preprocess** function, the input RGB irradiance image is first converted to grayscale and smoothed using mean and median filters to reduce noise and emphasize low-frequency lighting patterns. In the **compute_irradiance_difference** function, we evaluate directional contrast by drawing lines through the image center at various orientations, splitting the image into two half-planes. For each orientation, we compute the total irradiance in the positive and negative halves and take their difference as a measure of directional lighting strength. The orientation that yields the maximum contrast indicates the axis of strongest irradiance decay; the estimated light direction is taken to be orthogonal to this axis. To efficiently refine the estimate, we employ a coarse-to-fine strategy: starting from a uniform angular sampling over $[0, 2\pi]$, we iteratively narrow the search interval via binary refinement. The resulting 2D direction is lifted to 3D by appending a fixed z-axis component.

A.3 Evaluation Dataset Details

The detailed material category distribution is illustrated in Fig.1.

A.4 Additional Results

A.4.1 PBR Comparisons. In Fig.2 and Fig.3, we compare our method's material estimation results with state-of-the-art approaches, including RGB→X [Zeng et al. 2024], MatFusion [Sartor and Peers 2023], ControlMat [Vecchio et al. 2024], SurfaceNet [Vecchio et al. 2021], and MaterIA [Martin et al. 2022]. We directly use input images from the supplementary material of ControlMat that match our directional lighting assumptions to evaluate our method. Additionally, we take the results for the latter three comparison methods from the same supplementary material.

A.4.2 Highly Specular Materials. In Fig.4 and Fig.5, we show more estimation examples for highly specular metal materials comparing with baselines.

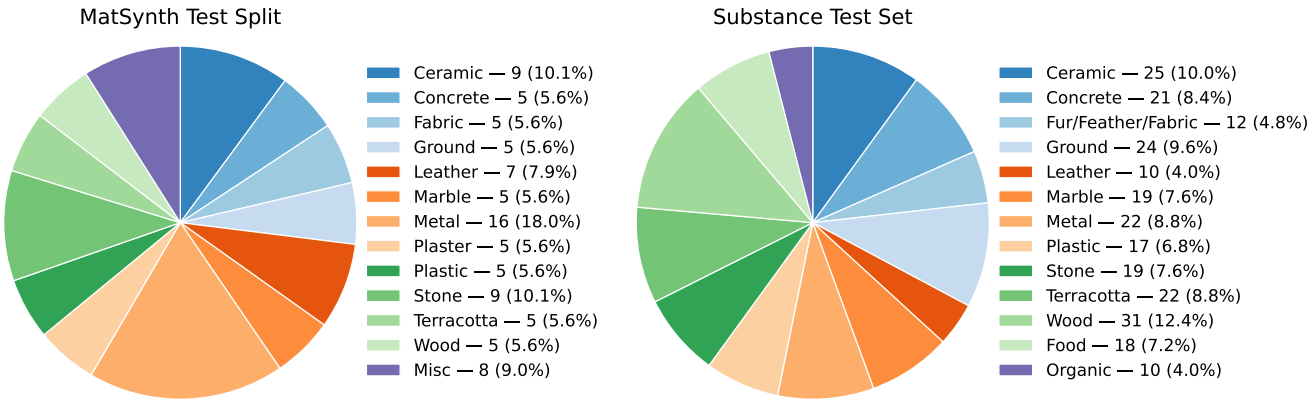

Fig. 1. Material category distributions for the two evaluation datasets.

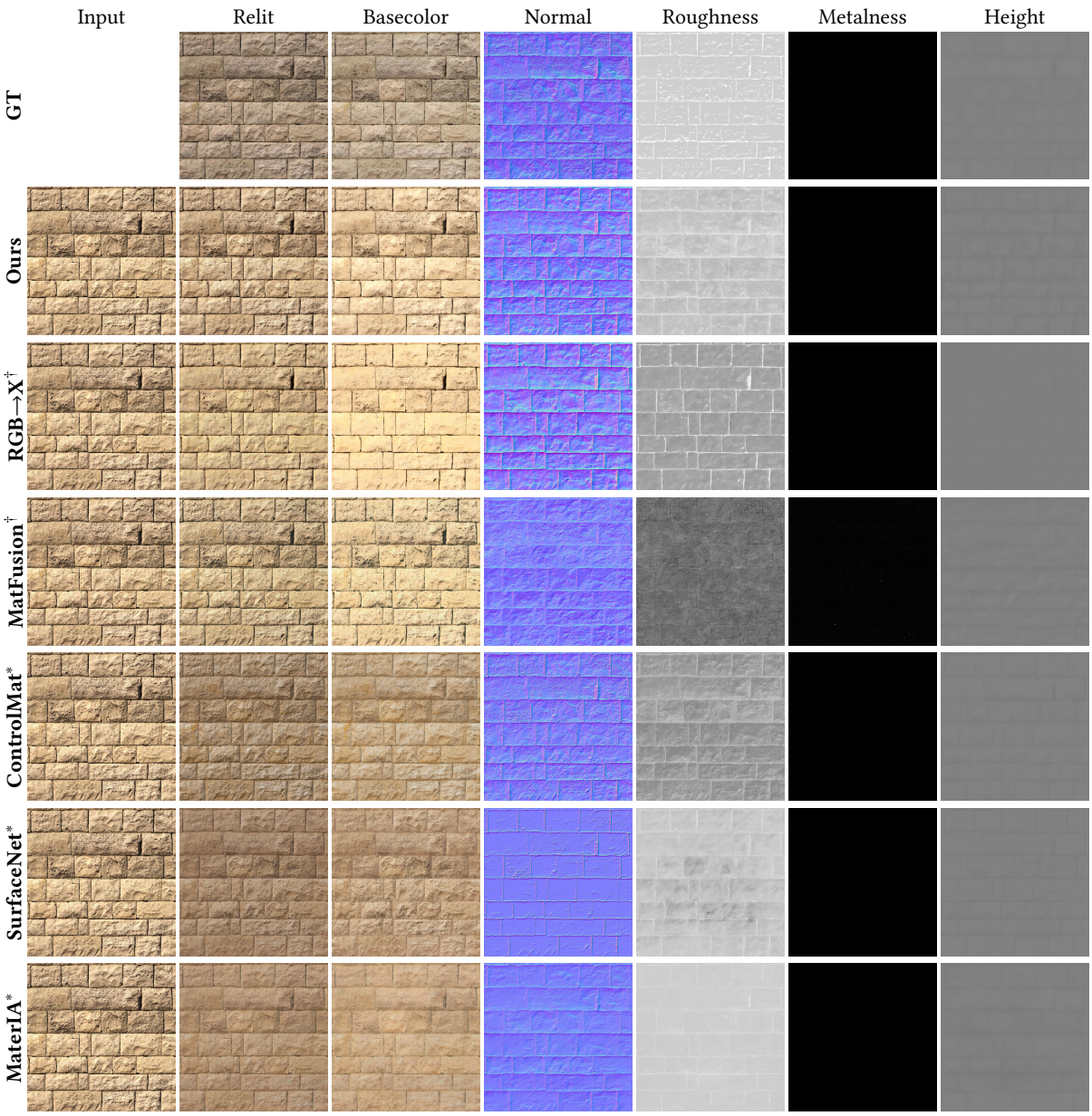

Fig. 2. **Material estimation comparison: brick wall.** †: trained on our dataset, ∗: results obtained from ControlMat supplementary material.

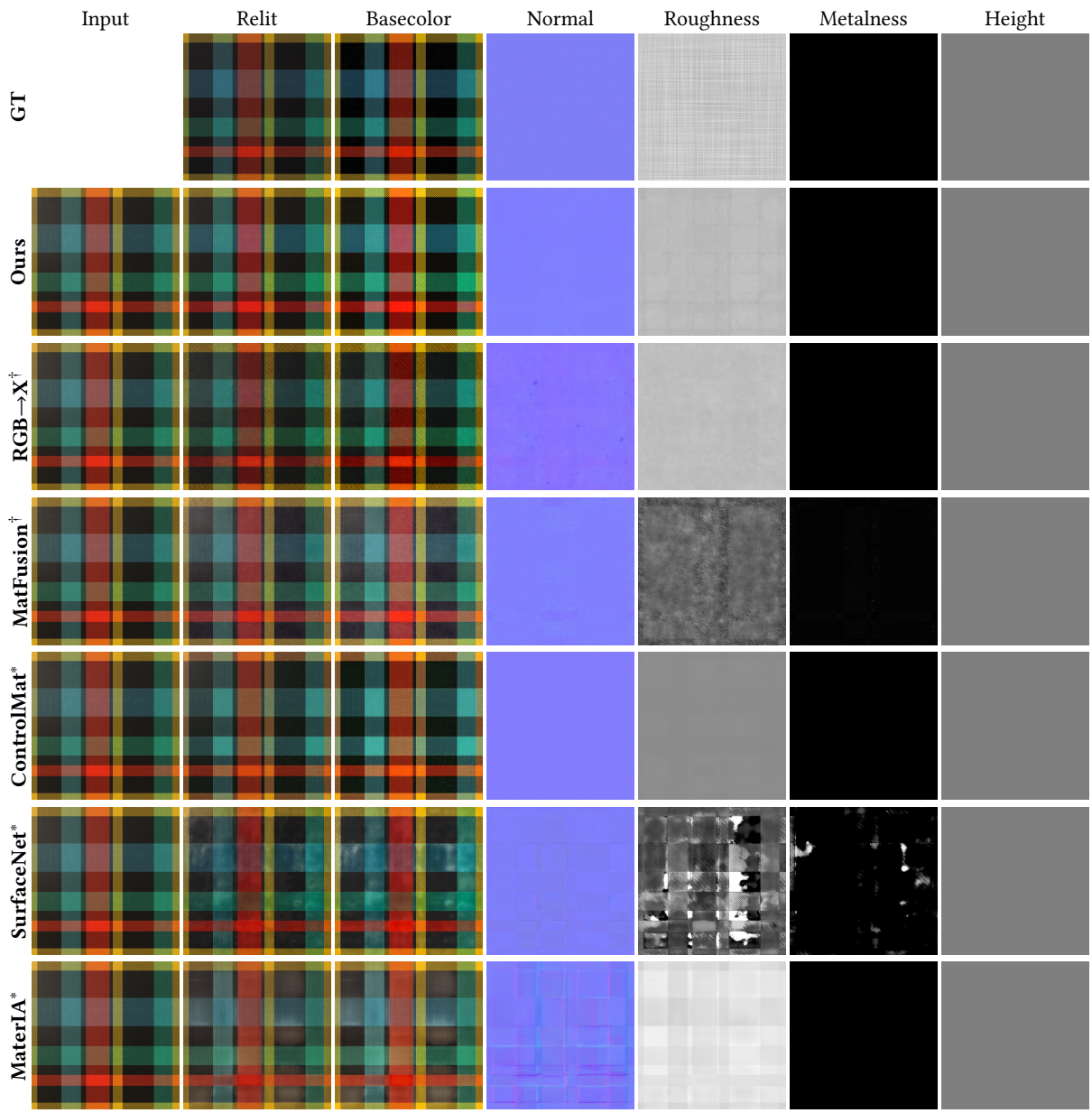

Fig. 3. **Material estimation comparison: fabric.** †: trained on our dataset, ∗: results obtained from ControlMat supplementary material.

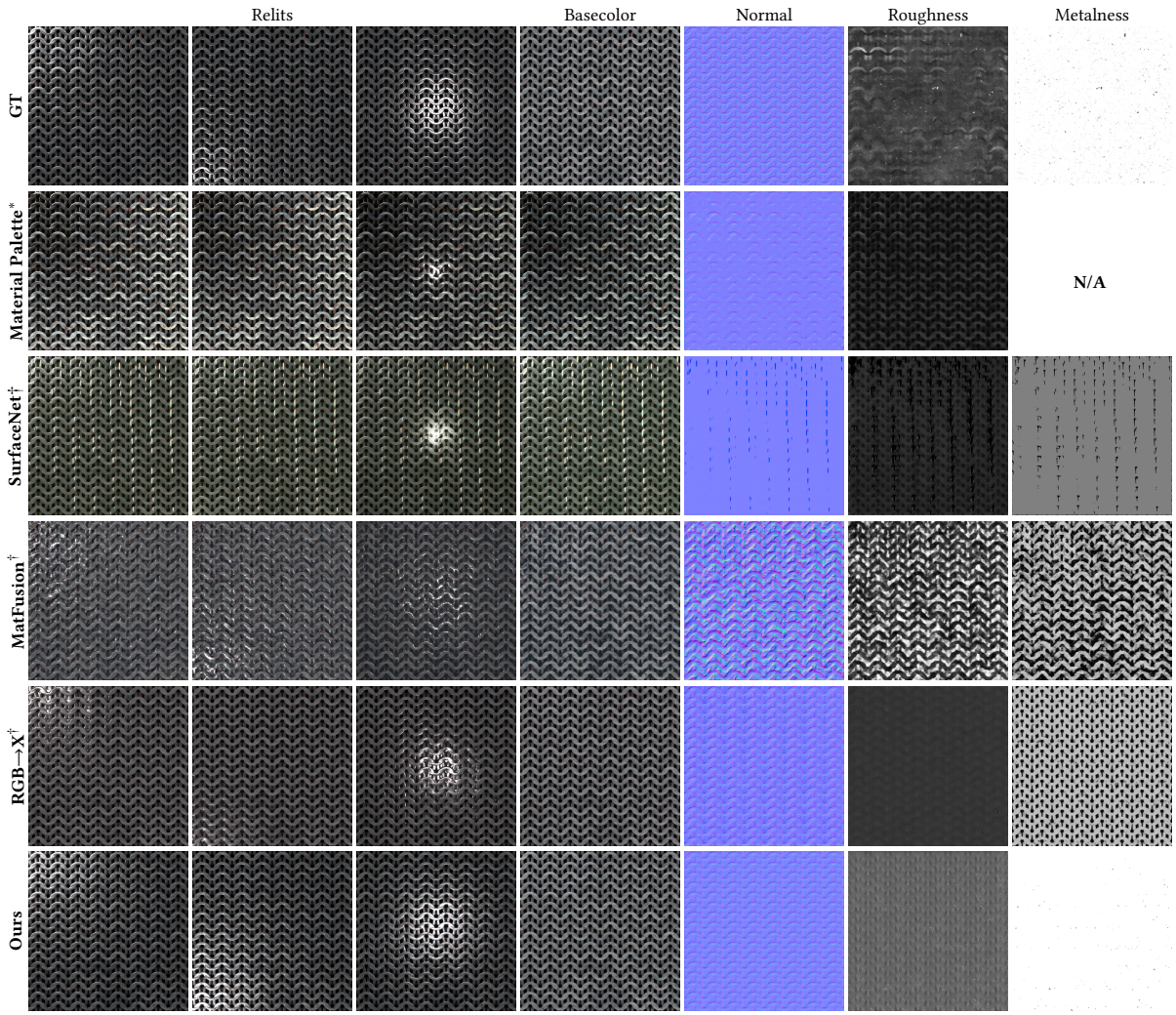

Fig. 4. **Additional estimation comparisons for chained metal.**

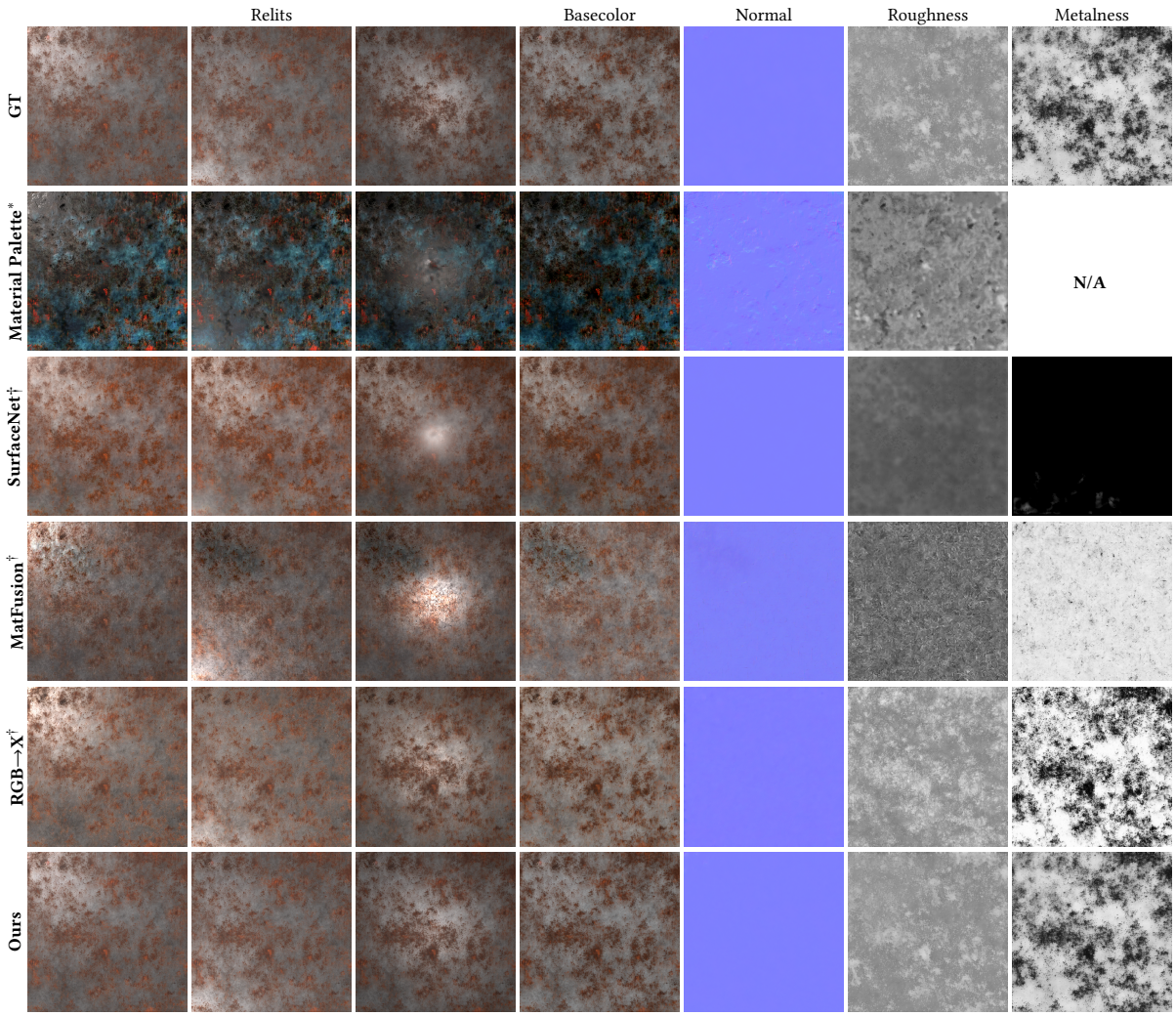

Fig. 5. **Additional estimation comparisons for metal surface.**

References

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. 2024. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems* 36 (2024).

Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. 2024. Common Diffusion Noise Schedules and Sample Steps are Flawed.

Rosalie Martin, Arthur Roullier, Romain Rouffet, Adrien Kaiser, and Tamy Boubekeur. 2022. MaterIA: Single Image High-Resolution Material Capture in the Wild. *Comput. Graph. Forum* (2022).

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2024. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=di52zR8xgf

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*.

Tim Salimans and Jonathan Ho. 2022. Progressive Distillation for Fast Sampling of Diffusion Models. In *Int. Conf. Learn. Represent. (ICLR)*. https://openreview.net/forum?id=TIdIXIpzhoI

Sam Sartor and Pieter Peers. 2023. Matfusion: a generative diffusion model for svbrdf capture. In *SIGGRAPH Asia 2023 conference papers*. 1–10.

Giuseppe Vecchio and Valentin Deschaintre. 2024. MatSynth: A Modern PBR Materials Dataset. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*.

Giuseppe Vecchio, Rosalie Martin, Arthur Roullier, Adrien Kaiser, Romain Rouffet, Valentin Deschaintre, and Tamy Boubekeur. 2024. Controlmat: a controlled generative approach to material capture. *ACM Transactions on Graphics* 43, 5 (2024), 1–17.

Giuseppe Vecchio, Simone Palazzo, and Concetto Spampinato. 2021. SurfaceNet: Adversarial SVBRDF Estimation from a Single Image. In *Int. Conf. Comput. Vis. (ICCV)*.

Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. 2024. RGB↔X: Image decomposition and synthesis using material- and lighting-aware diffusion models. *ACM Trans. on Graphics* (2024).