PBRSENSE: SINGLE-IMAGE PBR ESTIMATION WITH VLM-CONDITIONED IMAGE EDITOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Recovering the PBR (Physically based rendering) material properties of an object from a single image is a long-standing problem in computer vision and graphics. Traditional methods typically require multi-view inputs or controlled capture conditions and may take hours to solve the inverse rendering problem. Recent data-driven methods based on diffusion models can produce plausible material maps for in-distribution inputs but often struggle when the input falls outside the training distribution. VLM-conditioned image editing models provide a pretrained image-conditioned generation interface with language-controlled target specification. Whether such models can be adapted to predict PBR maps for general objects while addressing these limitations has not been systematically studied. In this paper, we propose PBRSense, which adapts a pretrained VLM-conditioned image editing model for pixel-aligned prediction of albedo, metallic, roughness, and surface normal maps from a single object image. Three attribute branches share the DiT backbone and use different text prompts to specify their prediction targets. We freeze the inherited VLM and VAE, selectively fine-tune DiT blocks, and use an endpoint velocity objective for single-step prediction. We compare trainable DiT ranges under a fixed training budget to inform the fine-tuning configuration. On the combined evaluation of two held-out benchmarks, the adapted model without communication already outperforms the evaluated baselines across the reported attributes; communication provides smaller additional gains. Qualitative results on real photographs illustrate its use beyond rendered inputs. Finally, using the predicted material maps as priors improves the material estimates of a frozen 3D generation model without additional training.

1 INTRODUCTION

PBR decomposes an object's appearance into a set of attribute maps, typically including albedo, metallic, and roughness (Burley, 2012). With only a small number of material parameters, PBR provides an expressive representation and has become a standard material representation for 3D assets, film, and game rendering. High-quality 2D PBR prediction can provide useful material references for creating 3D assets and help 3D generation pipelines produce more realistic materials.

Early methods posed the problem as inverse-rendering optimization (Barron & Malik, 2015; Zhang et al., 2021). They often require multiple views or controlled capture conditions, making them difficult to use for single-image prediction. Existing single-image PBR material estimation methods have widely leveraged pretrained image generation models. For example, IDArb (Li et al., 2025) is built upon the U-Net architecture of Stable Diffusion (Rombach et al., 2022), while Teamwork (Sartor & Peers, 2025) includes a DiT-based Stable Diffusion 3 (Esser et al., 2024) variant for multi-attribute decomposition. These methods improve visual detail and performance on real images, but predictions can become unreliable when the object or material is outside the coverage of the training data.

Recently, large-scale image editing models have emerged as a promising foundation for single-image PBR prediction. These models typically combine VLMs with DiTs, enabling them to condition on both image content and textual instructions while preserving rich semantic and spatial information (Wu et al., 2025a). A recent systematic evaluation of open-source image editing models (Liu et al., 2026) further showed that pretrained editors exhibit non-trivial capabilities on dense vision prediction tasks, suggesting that their learned representations may extend beyond natural im-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

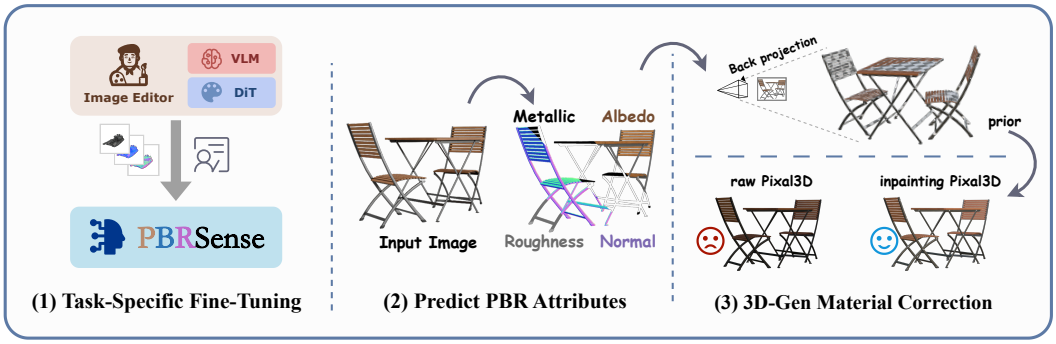

Figure 1: **Single-image PBR prediction and 3D material correction with PBRSense.** (1) We adapt a pretrained VLM-conditioned image editor to PBR estimation through task-specific fine-tuning. (2) Given a single object image, PBRSense predicts pixel-aligned albedo, metallic, roughness, and surface normals. (3) The predicted material maps are back-projected onto visible 3D surfaces and used as priors during material generation with a frozen Pixal3D model (Li et al., 2026a). In the example shown, the corrected appearance of the wood and metal more closely matches the input image.

age generation. This motivates adapting such models to single-image PBR attribute prediction task. However, unlike general-purpose image editing, which targets perceptually plausible natural images, PBR prediction requires the simultaneous generation of multiple pixel-aligned attribute maps following specific physical semantics and numerical conventions.

Motivated by these observations, we propose PBRSense, a single-image PBR estimator built on a pretrained VLM-conditioned image editing model. Our key idea is to rearrange PBR attribute prediction as a text-instructed image generation task, allowing the semantic understanding and spatial representations learned by pretrained image editing models to be transferred to PBR attribute prediction. Specifically, we organize the PBR attributes to be predicted into several three-channel RGB images and provide attribute-specific textual instructions, enabling the model to predict pixel-aligned attribute maps within a unified generation framework. Considering the differences in output distributions and numerical precision between general-purpose image editing and PBR prediction, we further investigate the parameter update range of the DiT and adopt a selective fine-tuning strategy to adapt the model to PBR prediction while preserving its pretrained representations as much as possible.

Extensive experiments show that PBRSense achieves substantial improvements on two held-out benchmarks and significantly outperforms IDArb, a recent strong baseline. PBRSense reduces metallic RMSE by 65.87% and roughness RMSE by 39.25%, improves albedo PSNR from 17.49 dB to 21.35 dB, and reduces normal angular error by 59.24%. On real photographs, PBRSense demonstrates stronger generalization and more accurately recovers the material and geometric details of real-world objects. Furthermore, we show that the PBR materials predicted by PBRSense are more accurate than those generated natively by the 3D generation model (Li et al., 2026a). Injecting our predicted materials into the 3D generation model in a training-free manner corrects color shifts and improves the fidelity of the generated materials as shown in Fig. 1.

Our contributions can be summarized as follows:

1. We introduce PBRSense, a single-image PBR estimator built on pretrained VLM-conditioned image editing model. The model predicts the attributes through a shared backbone and attribute-specific conditioning.

2. We investigate the effect of the DiT fine-tuning range on multi-attribute prediction. Ablation studies under a fixed training budget provide evidence for the choice.

3. We demonstrate improvements over existing methods on two held-out benchmarks and real photographs, and further show that PBRSense predictions can serve as effective material priors for improving single-image 3D material generation without additional training.

2 Related Work

2.1 Single-image material estimation

Optimization-based methods. Recovering material and illumination from images is a classical inverse rendering problem rooted in intrinsic image decomposition (Barrow & Tenenbaum, 1978). Optimization-based methods jointly infer scene geometry, reflectance, and lighting by minimizing rendering-based reconstruction objectives. Early single-image approaches relied on hand-crafted priors (Barron & Malik, 2015), while later methods introduced neural fields to disentangle reflectance and illumination (Zhang et al., 2021) and differentiable rendering to recover explicit geometry and BRDF parameters (Munkberg et al., 2022). Despite their physical interpretability, these methods often require multi-view observations, calibrated cameras, or other auxiliary information. Also, they typically involve costly per-scene optimization, limiting their applicability to unconstrained single-image material estimation.

Learning-based methods. Learning-based methods replace per-scene optimization with feed-forward prediction, typically trained on synthetic renderings, and have recently incorporated pretrained diffusion priors to improve robustness and output realism (Kocsis et al., 2024; Ke et al., 2025). Existing approaches span probabilistic intrinsic decomposition (Kocsis et al., 2024), unified decomposition and synthesis (Zeng et al., 2024), object-level inverse rendering under unknown illumination (Chen et al., 2024), multi-view or video settings (Li et al., 2025; Liang et al., 2025), and efficient one-step material prediction (Hong et al., 2025; Wu et al., 2025b). More recent methods further specialize model design for material attributes, including separate latent pathways for albedo and metallic/roughness estimation (Huang et al., 2025b) and DiT-based intrinsic decomposition that jointly predicts multiple physical attributes (Sartor & Peers, 2025; Zheng et al., 2025). Despite these advances, most diffusion-based approaches are adapted from generative backbones such as Stable Diffusion (Rombach et al., 2022) and remain sensitive to the training distribution. They may oversmooth spatially varying materials into nearly uniform metallic or roughness predictions in some cases (Li et al., 2025), while highly reflective or layered materials remain difficult to reconstruct (Hong et al., 2025). These limitations motivate studying whether newer pretrained image editing models can provide a stronger foundation for single-image material prediction in the input image coordinates.

2.2 VLM-conditioned image editors

Instruction-based image editing asks a model to understand an input image and an instruction and then produce a new image aligned with the input. This formulation is naturally close to dense prediction. DICEPTION adapts pretrained text-to-image diffusion models to multiple perception tasks, while Vision Banana instruction-tunes an image generator for tasks including segmentation, depth, and surface normal estimation (Zhao et al., 2025; Gabeur et al., 2026). Representative models include Qwen-Image, which uses a dual-stream MM-DiT backbone and Qwen2.5-VL conditioning, as well as FireRed-Image-Edit and LongCat-Image-Edit (Wu et al., 2025a; Bai et al., 2025; Super Intelligence Team, 2026a; Meituan LongCat Team et al., 2025). Recent work has found that such open-source editing models can already perform non-trivial dense visual prediction without task-specific fine-tuning (Liu et al., 2026). MARCUS-Avatar has adapted LongCat-Image-Edit for PBR estimation in UV space after face geometry reconstruction (Li et al., 2026b). Our PBRSense, instead, aims to extend the capability of those editing models above, resulting in a highly generalizable predictor of pixel-aligned 2D PBR attributes across diverse objects.

2.3 3D material generation

TRELLIS introduces structured latents for 3D generation (Xiang et al., 2025). TRELLIS.2 extends this approach to native geometry and PBR materials (Xiang et al., 2026). Pixal3D uses ray back-projection to lift image features into a 3D volume, enforcing explicit pixel-to-3D alignment between the generated asset and the input view (Li et al., 2026a). When geometry is known, Material Anything directly generates materials for meshes (Huang et al., 2025a). Intrinsic Image Fusion fuses material predictions from multiple views (Kocsis et al., 2026). MatMart predicts materials from one or more input views, bakes them into UV space, and generates materials for unobserved

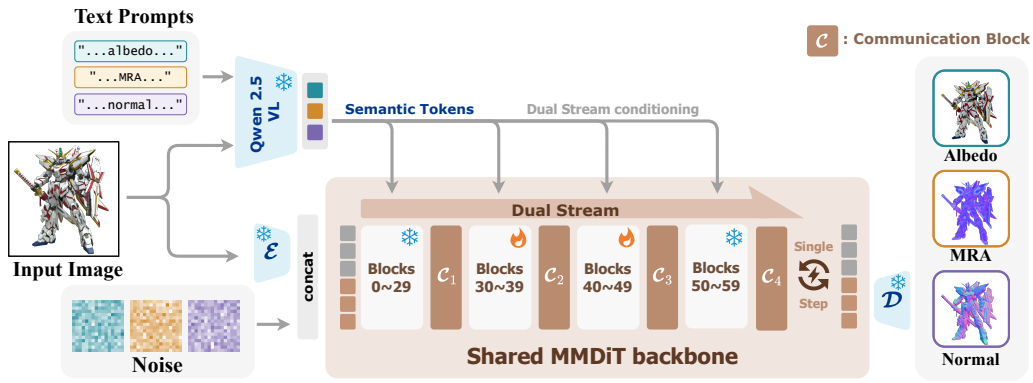

Figure 2: **Overview of PBRSense.** The attributes, Albedo, MRA (metallic, roughness, and auxiliary material alpha), and Normal, are predicted by parallel branches that share the same MM-DiT backbone. The model uses a single prediction step. We also attempt to insert lightweight cross-attribute communication modules between selected layers.

regions (Kocsis et al., 2026; Wu et al., 2026). DualMat also integrates material estimation into image-to-3D pipelines through multi-view prediction (Huang et al., 2025b). Our 3D experiment has a different goal: we do not retrain the 3D model. Instead, we project the single-image prediction onto the visible surface and preserve it as a known region during sampling (Lugmayr et al., 2022), allowing us to test whether the 2D prior itself can improve 3D material generation.

3 Method

PBRSense adapts a pretrained VLM-conditioned image editing model to estimate pixel-aligned PBR attributes and surface normals from a single image. We first formulate the task and our base model in Sec. 3.1, and then adapt the image editing backbone into three parallel attribute branches, as described in Sec. 3.2. Sec. 3.3 shows how we design our training process. Finally, we introduce lightweight cross-attribute communication to further refine the predictions in Sec. 3.4. An overview of the complete architecture is provided in Fig. 2.

3.1 Problem Setup

The input I is a single object image captured under unknown illumination, with no explicit assumption about multiple views, illumination, or geometry. We aim to recover pixel-aligned PBR attributes of the object: albedo $A \in \mathbb{R}^{H\times W\times 3}$, metallic $M \in \mathbb{R}^{H\times W\times 1}$, roughness $R \in \mathbb{R}^{H\times W\times 1}$, and camera-space surface normals $N \in \mathbb{R}^{H\times W\times 3}$. The material model follows a simplified Disney BRDF (Burley, 2012). We formulate the task as a joint conditional prediction problem as in equation 1.

$$p(A, M, R, N \mid I), \tag{1}$$

To align attribute map generation with the RGB image format, we represent the target attributes as three RGB three-channel images: An albedo map A, a material property map P, and a surface normal map N. Specifically, $P = (M, R, \alpha_{\text{mat}})$ packs metallic, roughness, and the auxiliary material alpha channel provided by the dataset into a three-channel image. Since most objects are opaque and material alpha is not a primary PBR attribute, we do not study or evaluate this attribute, though it is available in some of the datasets. Surface normals are also normalized and encoded as RGB images. All three targets are processed using the image representation and decoding interfaces of the pretrained VAE.

3.2 Model Adaptation Settings

We instantiate this adaptation framework using FireRed-Image-Edit (Super Intelligence Team, 2026a) as a representative pretrained VLM-conditioned editor. Its VLM, image VAE, dual-stream

MM-DiT, and image conditioning mechanism are inherited from the original model. The input image is encoded by the VAE to condition the target generation and is also processed by the VLM together with the text instruction to form a multimodal condition. The MM-DiT predicts the output latent under these conditions.

For the same input image, we construct three prediction inputs separately. Each branch uses the same image condition but is paired with independently initialized noise and a fixed attribute-specific prompt specifying whether to generate albedo, material properties, or surface normals. Rather than maintaining a separate generative backbone for each attribute, we process the three inputs in parallel along the batch dimension and share the same set of DiT weights, so that the weights can be jointly updated by training signals from different attributes.

3.3 Selective fine-tuning and Single-step learning

To limit the scope of task-specific updates, we freeze the pretrained VLM and VAE and fine-tune a selected range of DiT blocks. We use blocks 29–49 (zero-based index) in the final model. Sec. 4.3 compares this choice with equally sized ranges at other positions and with full DiT fine-tuning under a fixed training budget. We retain the backbone's flow parameterization (Esser et al., 2024) but fix the training endpoint to $\sigma = 1$, enabling single-step prediction at inference. Let $z_i = \mathcal{E}(Y_i)$ be the VAE latent of attribute map Y_i, and let ϵ_i be an independently sampled Gaussian noise vector for each branch. The model takes ϵ_i as the branch input and predicts a velocity $\hat{v}_i \approx \epsilon_i - z_i$. At inference time, one step gives $\hat{Y}_i = \mathcal{D}(\epsilon_i - \hat{v}_i)$, where $\mathcal{E}, \mathcal{D}$ are the shared and frozen VAE encoder and decoder. We train with a coverage-weighted squared error on the predicted latent velocity, excluding attribute targets without valid labels and filtering extreme residuals. The full loss, mask construction, and outlier filter are specified in Appendix A.2.

3.4 Lightweight cross-attribute refinement

To allow information exchange across attribute branches, we add lightweight communication modules that read and update only the target-image tokens. Each receiving branch attends to the other attributes through a local neighborhood path and a global pooled path. The resulting messages are normalized and added through independently learned scalar gates with bounded magnitude. All gates are initialized to zero to avoid disrupting the pretrained backbone. The module design, update equations, and implementation settings are provided in Appendix A.3; additional structure ablations are reported in Appendix C.

4 Experiments

4.1 Experimental setup

Implementation details. For the representative implementation, we fine-tune the public FireRed-Image-Edit-1.1 checkpoint (Super Intelligence Team, 2026a;b). Its backbone is a 60-layer dual-stream MM-DiT, and its semantic condition encoder is Qwen2.5-VL-7B (Bai et al., 2025). We update backbone blocks 29–49 (zero-based, inclusive) with a learning rate of 5×10^{-7} and all communication modules with a learning rate of 2×10^{-4}. The remaining backbone parameters, VLM, and VAE are frozen. PBRSense is trained at its native 1024×1024 resolution for 20,000 steps with a global batch size of 64 training samples on 32 NVIDIA H800 GPUs. Inference uses a single prediction step without CFG.

Baselines. We compare against eight publicly available baselines evaluated under the single-image setting : Intrinsic Image Diffusion (Kocsis et al., 2024), RGB↔X (Zeng et al., 2024), IntrinsicAnything (Chen et al., 2024), Marigold (Ke et al., 2025), IDArb (Li et al., 2025), SuperMat (Hong et al., 2025), Teamwork (Sartor & Peers, 2025), and DiffusionRenderer (Liang et al., 2025). We use the official code and weights for all baselines. We also evaluate FireRed-Image-Edit-1.1 without task-specific fine-tuning, using single-step inference and attribute prompts (Raw FireRed-1.1).

Evaluation metrics. We report PSNR for albedo. To remove ambiguity caused by brightness, we also report scale-invariant PSNR (si-PSNR), computed after fitting one least-squares scalar for

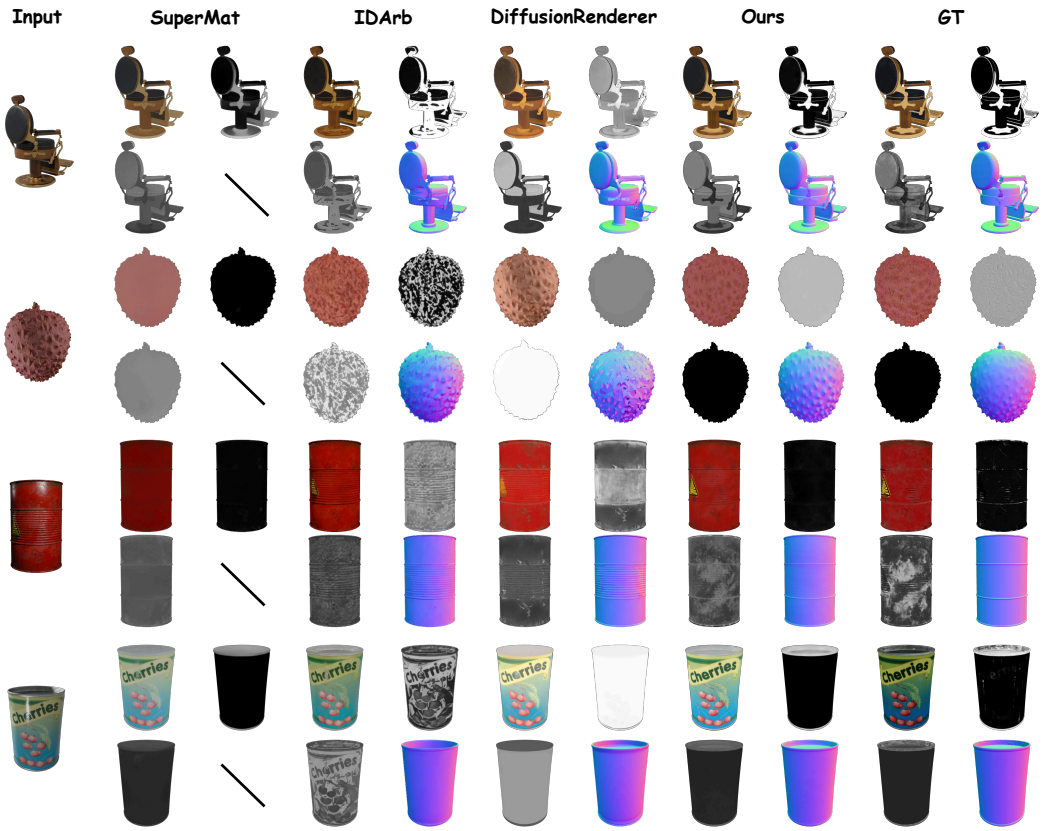

Figure 3: **Qualitative results.** In each subfigure, the top-left shows **albedo**, the top-right shows **metallic**, the bottom-left shows **roughness**, and the bottom-right shows **surface normals**. A backslash (\) indicates that the corresponding method does not provide that output.

each object. We report RMSE for metallic and roughness, and mean angular error for normals. Appendix B details the baseline settings, prediction alignment, valid regions, and metric aggregation.

Datasets. For training, we use assets from TexVerse (Zhang et al., 2025), Objaverse-XL (Deitke et al., 2023) (GitHub and Sketchfab), the synthetic subset of HY3D-Bench (Team Hunyuan3D et al., 2026), ABO (Collins et al., 2022), and Toys4K (Stojanov et al., 2021). The above datasets cover a diverse range of assets, including manually modeled assets, real-world scans, and 3D assets created by generation model. We render these assets to obtain paired RGB images and albedo, metallic, roughness, and normal maps for training. We apply quality filtering and use validity masks to handle missing or invalid attribute labels. The rendered RGB image serves as the input, and the corresponding attribute maps provide supervision. The test datasets are excluded from training.

For evaluation, we use DTC (Dong et al., 2025) and the PolyHaven-315 test set introduced by SViM3D (Engelhardt et al., 2025). DTC contains scanned objects while PolyHaven-315 is collected from the 3D model platform PolyHaven (Poly Haven, n.d.). Both datasets are re-rendered at 1024×1024 resolution. We also use real photographs for qualitative evaluation. The 3D application study uses assets with ground-truth meshes from DTC and PolyHaven-315.

4.2 Comparison with prior methods

Benchmarks. Table 1 compares PBRSense with baselines on the test sets. PBRSense achieves higher albedo PSNR and si-PSNR, lower metallic and roughness RMSE, and lower normal angular error than the listed baselines that predict each attribute.

Table 1: **Quantitative comparison on DTC and PolyHaven-315**. Results are averaged across datasets, weighted by the number of objects. A, M, R, and N denote albedo, metallic, roughness, and normals. A dash indicates an unreported result. **Red** and **orange** cell backgrounds mark the best and second-best values in each column among the listed variants. * denotes that Marigold employs a different model for predicting surface normals, separate from the predictions of other PBR attributes.

Method	A-PSNR ↑	A-siPSNR ↑	M-RMSE ↓	R-RMSE ↓	N-angle (°) ↓
Intrinsic Image Diffusion	13.251	19.037	0.252	0.226	–
RGB↔X	13.187	20.711	0.487	0.218	26.715
IntrinsicAnything	16.665	20.483	–	–	–
Marigold	8.993	19.302	0.272	0.277	17.280*
IDArb	17.491	21.091	0.206	0.207	18.501
SuperMat	17.829	21.987	0.089	0.141	–
Teamwork	15.261	18.679	–	0.230	26.098
DiffusionRenderer	13.802	20.940	0.434	0.297	15.485
Raw FireRed-1.1	14.468	18.367	0.363	0.294	–
Ours w/o communication	21.162	24.749	0.076	0.128	7.739
Ours	21.354	24.954	0.070	0.126	7.541

The adapted backbone without communication already achieves the best score among the listed methods for each reported metric. This shows that task adaptation accounts for most of the improvement in the main comparison.

Fig. 3 shows qualitative comparisons of PBRSense with several baseline methods across the PBR attributes and surface normals. Our method produces cleaner and more spatially consistent attribute maps. For albedo prediction, our model effectively removes illumination effects while preserving the original texture of the object with high fidelity. For roughness and metallic prediction, our model produces clearer boundaries between components made of different materials. For surface normal prediction, our model better preserves the details of small components without introducing additional noise, while maintaining smooth and visually consistent transitions.

Real photographs. To compare the generalizability of the methods, we additionally report qualitative results on a set of in-the-wild object photographs collected from the Internet. We present several representative examples here in Fig. 4. In the first example, our model shows a clear advantage in normal estimation. It produces surface normals that better shows the actual geometric structure of the object and are less affected by visual interference from transparent components. Although IDArb predicts more fine-grained details in some attributes, some of these details are not consistent with the apparent material structure of the object. In the fourth example, our model recovers the textual details on the object with higher fidelity in the albedo prediction. In the other examples, our model can distinguish among different materials, such as leather, metal, and wood, and produces corresponding attribute maps with plausible properties. In contrast, IDArb shows weaker generalization to real-world objects and produces unreasonable predictions for several attributes.

4.3 Ablation Study

A key question in our study is how to achieve the best fine-tuning performance under a limited training budget. We investigate which layers of the 60-layer MM-DiT backbone are more suitable for task adaptation. All experiments start from the same FireRed-Image-Edit-1.1 weights and fine-tune three equal-length ranges: layers 0–20, 29–49, and 39–59, along with a full 60-layer fine-tuning baseline. All variants use the same training protocol, including the training set, resolution, cross-attribute communication modules, and single-step inference. To control the computational cost, we compare the results after 4,000 training steps, as shown in Table 2.

Among the three variants that fine-tune 21 layers, fine-tuning layers 29–49 consistently outperforms fine-tuning the early or late layers across all reported metrics. Compared with partial unfreezing, fully unfreezing the DiT further improves the PSNR and N-angle metrics. However, full unfreezing



Figure 4: Results on in-the-wild photographs

Table 2: Backbone adaptation at 4,000 steps. PBRSense updates blocks 29–49; Front, Tail update blocks 0–20, 39–59. All updates all the 60 blocks. Red and orange cell backgrounds mark the best and second-best values in each column among the listed variants. All models include the cross-attribute communication blocks.

Method	A-PSNR↑	A-si-PSNR↑	M-RMSE↓	R-RMSE↓	N-angle (°)↓
PBRSense	20.318	24.051	0.1171	0.1580	9.365
Front	19.300	23.186	0.1916	0.1714	11.861
Tail	19.882	23.515	0.1455	0.1631	10.565
All	20.534	24.264	0.1223	0.1597	8.930

does not improve all attributes simultaneously, it also leads to a certain degradation in roughness and metallic prediction. These results suggest that expanding the range of trainable parameters does not necessarily lead to corresponding performance gains. To balance the training budget and model performance, we therefore adopt blocks 29–49 as our default configuration.

4.4 Application: Training-free transfer to 3D materials

PBRSense provides 2D material maps that can guide a frozen 3D material generator without additional training. We project its albedo, metallic, and roughness predictions onto visible surfaces and use them to guide Pixal3D (Li et al., 2026a) during material sampling. Fig. 1 (right) illustrates this process: the material priors correct the overly dark appearance of the metal frames in the raw Pixal3D output, bringing the result closer to the input image. We compare it with the raw RGB-conditioned Pixal3D pipeline, using the same ground-truth geometry, cameras, sampling steps, and random seed. Appendix E.1 describes the visibility test and the material-prior correction during sampling. To examine the effect of the 2D material prior, we compare raw RGB-conditioned Pixal3D with variants guided by IDArb (Li et al., 2025) and PBRSense. Both guided variants use the same projection and material-correction procedure. We randomly select 100 objects from DTC and 100 from PolyHaven, with one input view and one novel view per object. For each attribute and view, we compute RMSE per object and then average equally across objects.

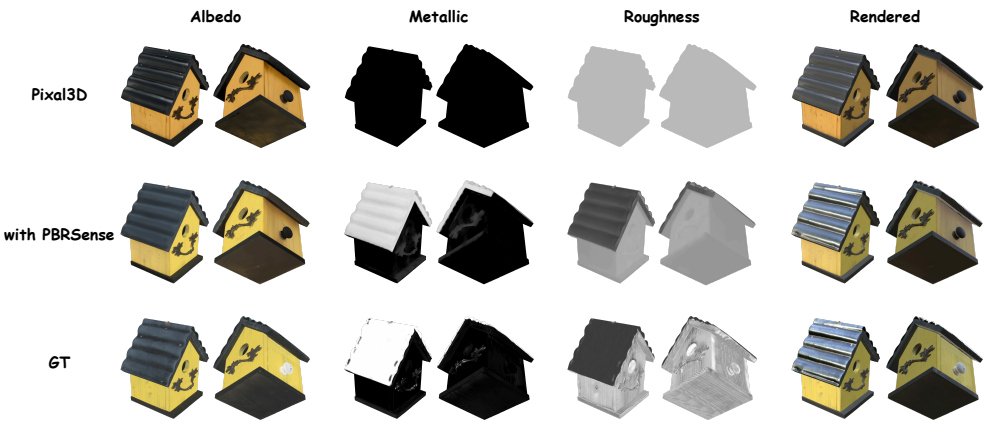

Figure 5: Qualitative multiview PBR results for 3D material generation.

Table 3: Training-free material reconstruction on the selected DTC and PolyHaven objects. We report mean per-object RMSE (↓). Red and orange cell backgrounds mark the best and second-best values in each column among the listed variants.

	Input view			Novel view		
Method	Albedo	Metallic	Roughness	Albedo	Metallic	Roughness
Raw Pixal3D	0.126	0.227	0.146	0.149	0.243	0.153
w/ IDArb	0.148	0.439	0.178	0.141	0.304	0.145
w/ PBRSense	0.094	0.184	0.110	0.124	0.217	0.121
Error reduction w/ PBRSense (%)	25.57	18.85	24.40	16.84	10.84	20.47

Table 3 shows that PBRSense guidance achieves the lowest mean RMSE for all three material attributes at both the input and novel views. Relative to raw Pixal3D, IDArb guidance increases the input-view errors for all three attributes; at the novel view, it slightly reduces albedo and roughness errors but increases metallic error. PBRSense guidance, in comparison, reduces all six errors. In particular, it reduces metallic and roughness RMSE by 18.85% and 24.40% at the input view, and by 10.84% and 20.47% at the novel view. These results suggest that the effect of training-free material correction depends on the injected 2D maps, and show that PBRSense provides useful material priors on this selected subset.

We present the PBR attributes of the same object from multiple views (Fig. 5). With PBRSense as a prior, the predicted PBR attributes are substantially improved and become more consistent with the ground truth across views, demonstrating the effectiveness of our approach. Results of more objects are presented in Appendix E.

5 Conclusion

In this paper, we presented PBRSense, a single-image PBR estimator built on a VLM-conditioned image editing model. Through shared-backbone multi-attribute prediction, one-step learning, and selective fine-tuning, PBRSense generates albedo, metallicity, roughness, and surface normal maps from the input single image. Ablation studies under a fixed training budget show that the choice of fine-tuning layers affects the prediction of different attributes, while cross-attribute communication brings further improvements beyond backbone adaptation. Our method performs well on both test datasets and in-the-wild real photographs. Furthermore, PBRSense can also guide a frozen image-to-3D material generator and improve its material estimates without any additional training. Together, these results show that a pretrained VLM-conditioned image editor can be adapted for single-image PBR prediction, and that its material maps can guide a frozen 3D material generator.

AI USE STATEMENT

We used generative AI tools to assist with literature search and synthesis, manuscript organization, translation, language editing, and code development. The authors reviewed the AI-assisted content and take responsibility for the final text, references, code, experimental claims, and other content of this work.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Jonathan T. Barron and Jitendra Malik. Shape, Illumination, and Reflectance from Shading. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 37(08):1670–1687, August 2015. ISSN 1939-3539. doi: 10.1109/TPAMI.2014.2377712. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2014.2377712.

Harry G. Barrow and J. Martin Tenenbaum. Recovering intrinsic scene characteristics from images. In Allen R. Hanson and Edward M. Riseman (eds.), *Computer Vision Systems*, pp. 3–26. Academic Press, New York, 1978.

Brent Burley. Physically-based shading at Disney. In *ACM SIGGRAPH 2012 Courses*, 2012. URL https://media.disneyanimation.com/uploads/production/publication_asset/48/asset/s2012_pbs_disney_brdf_notes_v3.pdf. Practical Physically Based Shading in Film and Game Production.

Xi Chen, Sida Peng, Dongchen Yang, Yuan Liu, Bowen Pan, Chengfei Lv, and Xiaowei Zhou. IntrinsicAnything: Learning diffusion priors for inverse rendering under unknown illumination. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LX*, 2024. doi: 10.1007/978-3-031-73027-6_26.

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F. Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. ABO: Dataset and benchmarks for real-world 3D object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21126–21136, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Collins_ABO_Dataset_and_Benchmarks_for_Real-World_3D_Object_Understanding_CVPR_2022_paper.html.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A universe of 10M+ 3D objects. In *Advances in Neural Information Processing Systems*, volume 36, pp. 35799–35813, 2023. URL https://arxiv.org/abs/2307.05663.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 753–763, 2025. doi: 10.1109/CVPR52734.2025.00079.

Andreas Engelhardt, Mark Boss, Vikram Voleti, Chun-Han Yao, Hendrik P. A. Lensch, and Varun Jampani. SViM3D: Stable video material diffusion for single image 3D generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28428–28439, 2025. URL https://arxiv.org/abs/2510.08271.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Valentin Gabeur, Shangbang Long, Songyou Peng, Paul Voigtlaender, Shuyang Sun, Yanan Bao, Karen Truong, Zhicheng Wang, Wenlei Zhou, Jonathan T. Barron, Kyle Genova, Nithish Kannen, Sherry Ben, Yandong Li, Mandy Guo, Suhas Yogin, Yiming Gu, Huizhong Chen, Oliver Wang, Saining Xie, Howard Zhou, Kaiming He, Thomas Funkhouser, Jean-Baptiste Alayrac, and Radu Soricut. Image generators are generalist vision learners. *arXiv preprint arXiv:2604.20329*, 2026. URL https://arxiv.org/abs/2604.20329.

Yijia Hong, Yuan-Chen Guo, Ran Yi, Yulong Chen, Yan-Pei Cao, and Lizhuang Ma. SuperMat: Physically consistent PBR material estimation at interactive rates. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25083–25093, 2025. URL https://arxiv.org/abs/2411.17515.

Xin Huang, Tengfei Wang, Ziwei Liu, and Qing Wang. Material anything: Generating materials for any 3D object via diffusion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26556–26565, 2025a. doi: 10.1109/CVPR52734.2025.02473. URL https://arxiv.org/abs/2411.15138.

Yifeng Huang, Zhang Chen, Yi Xu, Minh Hoai, and Zhong Li. DualMat: PBR material estimation via coherent dual-path diffusion. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025b. doi: 10.1145/3746027.3755237. URL https://arxiv.org/abs/2508.05060.

Bingxin Ke, Kevin Qu, Tianfu Wang, Nando Metzger, Shengyu Huang, Bo Li, Anton Obukhov, and Konrad Schindler. Marigold: Affordable adaptation of diffusion-based image generators for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2025. doi: 10.1109/TPAMI.2025.3591076.

Peter Kocsis, Vincent Sitzmann, and Matthias Nießner. Intrinsic image diffusion for indoor single-view material estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5198–5208, June 2024.

Peter Kocsis, Lukas Höllein, and Matthias Nießner. Intrinsic image fusion for multi-view 3D material reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22571–22580, 2026. URL https://arxiv.org/abs/2512.13157.

Dong-Yang Li, Wang Zhao, Yuxin Chen, Wenbo Hu, Meng-Hao Guo, Fang-Lue Zhang, Ying Shan, and Shi-Min Hu. Pixal3D: Pixel-aligned 3D generation from images. In *ACM SIGGRAPH 2026 Conference Papers*, 2026a. URL https://arxiv.org/abs/2605.10922.

Hong Li, Minqi Meng, Yanjun Liang, Chongjie Ye, Houyuan Chen, Weiqing Xiao, Xianda Guo, Guojun Lei, Xuhui Liu, Chaojie Yang, Yanlun Peng, Hao Zhao, and Baochang Zhang. Monocular avatar reconstruction via cascaded diffusion priors and UV-space differentiable shading. In *European Conference on Computer Vision (ECCV)*, 2026b. URL https://arxiv.org/abs/2606.28144.

Zhibing Li, Tong Wu, Jing Tan, Mengchen Zhang, Jiaqi Wang, and Dahua Lin. IDArb: Intrinsic decomposition for arbitrary number of input views and illuminations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.12083.

Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Chih-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, and Zian Wang. Diffusion renderer: Neural inverse and forward rendering with video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26069–26080, June 2025.

Wei Liu, Jiaxin Lin, and Rui Chen. Open-source image editing models are zero-shot vision learners, 2026. URL https://arxiv.org/abs/2605.04566.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11461–11471, 2022.

Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, Xunliang Cai, Yayong Guan, and Jie Hu. LongCat-Image technical report, 2025. URL https://arxiv.org/abs/2512.07584.

Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3D models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8280–8290, 2022.

Poly Haven. The public 3D asset library, n.d. URL https://polyhaven.com/. CC0 asset library. Accessed 2026-09-20.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Sam Sartor and Pieter Peers. Teamwork: Collaborative diffusion with low-rank coordination and adaptation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, 2025. doi: 10.1145/3757377.3763870. URL https://doi.org/10.1145/3757377.3763870.

Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1798–1808, 2021. URL https://arxiv.org/abs/2101.07296.

Super Intelligence Team. FireRed-Image-Edit-1.0 technical report, 2026a. URL https://arxiv.org/abs/2602.13344.

Super Intelligence Team. FireRed-Image-Edit-1.1, 2026b. URL https://huggingface.co/FireRedTeam/FireRed-Image-Edit-1.1. Hugging Face model card.

Team Hunyuan3D, Bowen Zhang, Chunchao Guo, Dongyuan Guo, Haolin Liu, Hongyu Yan, Huiwen Shi, Jiaao Yu, Jiachen Xu, Jingwei Huang, Kunhong Li, Lifu Wang, Linus, Penghao Wang, Qingxiang Lin, Ruining Tang, Xianghui Yang, Yang Li, Yirui Guan, Yunfei Zhao, Yunhan Yang, Zeqiang Lai, Zhihao Liang, and Zibo Zhao. HY3D-Bench: Generation of 3D assets. *arXiv preprint arXiv:2602.03907*, 2026. URL https://arxiv.org/abs/2602.03907.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image technical report, 2025a. URL https://arxiv.org/abs/2508.02324.

Xiuchao Wu, Pengfei Zhu, Jiangjing Lyu, Xinguo Liu, Jie Guo, Yanwen Guo, Weiwei Xu, and Chengfei Lyu. StableIntrinsic: Detail-preserving one-step diffusion model for multi-view material estimation. *arXiv preprint arXiv:2508.19789*, 2025b.

Xiuchao Wu, Pengfei Zhu, Jiangjing Lyu, Xinguo Liu, Jie Guo, Yanwen Guo, Weiwei Xu, and Chengfei Lyu. MatMart: Material reconstruction of 3D objects via diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2336–2345, 2026. URL https://arxiv.org/abs/2511.18900.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480, June 2025.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14419–14429, 2026. URL https://arxiv.org/abs/2512.14692.

Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. RGB↔X: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. doi: 10.1145/3641519.3657445. URL https://doi.org/10.1145/3641519.3657445.

Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.*, 40(6), December 2021. ISSN 0730-0301. doi: 10.1145/3478513.3480496. URL https://doi.org/10.1145/3478513.3480496.

Yibo Zhang, Li Zhang, Rui Ma, and Nan Cao. TexVerse: A universe of 3D objects with high-resolution textures. *arXiv preprint arXiv:2508.10868*, 2025. URL https://arxiv.org/abs/2508.10868.

Canyu Zhao, Yanlong Sun, Mingyu Liu, Huanyi Zheng, Muzhi Zhu, Zhiyue Zhao, Hao Chen, Tong He, and Chunhua Shen. DICEPTION: A generalist diffusion model for visual perceptual tasks. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.17157.

Rongjia Zheng, Qing Zhang, Chengjiang Long, and Wei-Shi Zheng. DNF-Intrinsic: Deterministic noise-free diffusion for indoor inverse rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10342–10352, 2025. URL https://arxiv.org/abs/2507.03924.

A Implementation Details

A.1 VLM text prompts

We use fixed attribute-specific text prompts to specify the prediction target for each branch. The same prompts are used during both training and inference. The exact prompts are listed below.

Albedo.

```
Convert this rendered image into its albedo (base
color) map:  remove all lighting, shading, shadows
and highlights, keep only the intrinsic surface colors
on a black background.
```

Metallic and Roughness.

```
Convert this rendered image into its material property
map:  encode metallic in the red channel, roughness
in the green channel, and material alpha in the blue
channel, on a black background.
```

Surface Normal.

```
Convert this rendered image into its camera-space
surface normal map:  encode the normal direction at
every pixel as RGB with x pointing right, y pointing
up and z toward the camera, on a flat gray background.
```

A.2 Training loss details

We use the endpoint velocity regression and single-step inference defined in Sec. 3.3. Below, $\hat{v}_{bi}$ and v^*_{bi} denote the predicted and target latent velocities for sample b and attribute i, respectively. Valid-region masks are pooled by area onto the token grid to give coverage $c_{bi}(p)$. The weight is $w_{bi}(p) = 0.25 + 0.75\, c_{bi}(p)$. Let $\mathcal{V}$ be the set of (sample, attribute) pairs that have a valid label. Then

$$\mathcal{L} = \frac{1}{|\mathcal{V}|} \sum_{(b,i)\in\mathcal{V}} \frac{\sum_p w_{bi}(p)\, \rho(\hat{v}_{bi}(p) - v^*_{bi}(p))}{\sum_p w_{bi}(p)}, \qquad \rho(e) = \frac{1}{C} \sum_{k=1}^{C} e_k^2\, \mathbf{1}(|e_k| \le 50). \tag{2}$$

C is the channel count of the packed target latent. The outlier filter applies only to the numerator. Albedo and material coverage is derived from the target alpha masks; normals use a separate validity mask. Spatially invalid regions retain the background weight of 0.25, while sample–attribute pairs without labels are excluded from the loss. There is no extra class-balance term on metallic pixels.

A.3 Lightweight cross-attribute communication

This subsection provides the full design of the refinement module introduced in Sec. 3.4. Fig. 6 shows the full communication block on the left and a directed cross-attribute attention example on the right.

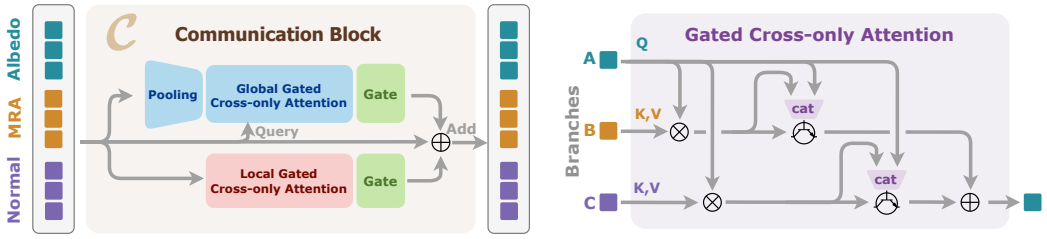

Figure 6: Cross-attribute communication. Left: the albedo, MRA, and normal feature streams enter the communication block. For each receiving branch, a global path pools source features before attention, while a local path reads nearby source tokens. Each path has its own gate, and the gated messages are added to the receiving features through a residual update. Right: the albedo branch A provides queries, while MRA (B) and normal (C) provide keys and values in the illustrated directed attention operation. The same operation is applied with each attribute as the receiving branch. MRA is the packed material branch P in the equations.

Target tokens and projections. Let $H_i \in \mathbb{R}^{L\times d}$ denote the target-image token features of branch $i \in \{A, P, N\}$ at one communication site, where L is the number of target tokens and d is the backbone feature width. The communication module reads and updates only these target tokens; it does not read or update the input-image condition tokens or the text tokens. The equations below describe valid source and receiving branches. Messages are masked out when either branch lacks a valid attribute label during training. As shown on the right of Fig. 6, the receiver forms queries from its own features, while each of the other two branches supplies keys and values. The figure uses albedo as the receiver; in the model, the roles rotate so that all three attributes receive messages. Each branch is normalized and projected down, $Z_i = \mathrm{LN}_i(H_i)W_i^{\downarrow}$, then forms queries, keys, and values:

$$Q_i = Z_i W_i^Q, \qquad K_j = Z_j W_j^K, \qquad V_j = Z_j W_j^V, \qquad j \neq i. \tag{3}$$

Layer normalization and projections are attribute-specific. The local and global paths share the same query, key, and value projections. For each receiving branch, W_i^O and $W_i^\uparrow$ are shared across source branches and across the two paths. Linear biases and multi-head layout are omitted from the equations.

Local and global messages. The left of Fig. 6 shows how the directed interactions are used in parallel local and global paths. The receiver's features bypass both paths and provide the residual to which the gated messages are added. At target location p, the local path reads a 3×3 neighborhood $\mathcal{N}(p)$ of the source. The global path reads $\bar{Z}_j$, the source features average-pooled onto an 8×8 grid:

$$\begin{aligned} C^{\mathrm{loc}}_{i\leftarrow j}(p) &= \mathrm{Attn}(Q_i(p),\, K_j[\mathcal{N}(p)],\, V_j[\mathcal{N}(p)])\,, \\ C^{\mathrm{glob}}_{i\leftarrow j}(p) &= \mathrm{Attn}\big(Q_i(p),\, \bar{Z}_j W_j^K,\, \bar{Z}_j W_j^V\big)\,. \end{aligned} \tag{4}$$

The local path adds a learned bias to the center attention logit, separately for each directed edge and attention head, and masks positions outside the token grid. The context is mapped back to the backbone width, $U^s_{i\leftarrow j} = C^s_{i\leftarrow j} W_i^O W_i^\uparrow$ for $s \in \{\mathrm{loc}, \mathrm{glob}\}$.

Bounded residual updates. Each message is normalized and scaled by the receiving token's root mean square (RMS):

$$\hat{m}^s_{i\leftarrow j}(p) = \mathrm{sg}\left(\frac{\|H_i(p)\|_2}{\sqrt{d}}\right) \frac{U^s_{i\leftarrow j}(p)}{\max\big(\|U^s_{i\leftarrow j}(p)\|_2, \varepsilon\big)}. \tag{5}$$

Here, sg denotes the stop-gradient operator and ε is a numerical stabilizer. A learnable scalar gate controls each message before summation:

$$H'_i(p) = H_i(p) + \sum_{j\neq i} \sum_{s\in\{\mathrm{loc},\mathrm{glob}\}} \alpha_{\max} \tanh\big(g^s_{i\leftarrow j}\big)\; \hat{m}^s_{i\leftarrow j}(p). \tag{6}$$

Each insertion site, directed edge, and local or global path has its own gate $g^s_{i\leftarrow j}$, shared across tokens and inputs. With $\alpha_{\max} = 0.05$, the ℓ_2 norm of each individual gated message is at most 5% of the receiving token RMS. This bound applies before messages are summed and does not bound the total update or final network output.

Initialization and placement. All gates start at zero, making the communication update exactly zero at initialization. $W^\uparrow$ is drawn from a small Gaussian with standard deviation 10^{-4} rather than initialized to zero, allowing the gate to receive a gradient at the first step. PBRSense inserts communication modules after blocks 29, 39, 49, and 59, using zero-based indices. All four modules use a 256-dimensional attention space and four attention heads. Parameters are not shared across sites. All communication parameters are trainable under the optimization settings in Sec. 4.1.

B Baseline Settings and Evaluation Protocol

B.1 Baseline inference settings

We use the released baseline weights with the inference settings in Table 4. Each method receives one input image independently. The listed resolution is the internal processing size, which can differ from the saved output size and the evaluation grid. The evaluation wrappers use random seed 42.

Checkpoints and input handling. Intrinsic Image Diffusion uses the released `iid_e250.pth` checkpoint. IntrinsicAnything uses its released albedo checkpoint, `last.ckpt`; its inference pipeline crops and pads the foreground before processing and restores the prediction to the input frame. Marigold uses `marigold-iid-appearance` and `marigold-normal`. Its separate normal model is used only for the reported normal comparison and is marked with *. RGB↔X uses the public replacement checkpoint. Its wrapper limits the longest input side to 1000 pixels and rounds dimensions down to multiples of eight, giving 1000^2 processing for the square benchmark inputs. It uses deterministic DDIM sampling with $\eta = 0$. IDArb uses its released intrinsic-decomposition checkpoint in single-view mode,

Table 4: Baseline inference settings used for the reported comparisons. Resolution refers to internal processing. Sampling counts refer to each input image; a multi-step sampler still produces one output unless averaging is specified.

Method	Processing size	Sampling settings
Intrinsic Image Diffusion	640×480	100 DDIM steps; average 10 samples
RGB↔X	1000^2	50 steps; one sample; text and image guidance scales 0
IntrinsicAnything	256^2	100 DDIM steps; one sample; guidance scale 1
Marigold	768^2	4 steps; ensemble size 1
IDArb	512^2	50 steps; one view and one sample; guidance scale 1
SuperMat	512^2	1 step; one sample; empty prompt
Teamwork	1024^2	50 steps; one sample; guidance scale 1
DiffusionRenderer	512^2	20 steps; one frame per output attribute; guidance scale 1
Raw FireRed-1.1	1024^2	1 step; guidance scale 1

with $\eta = 1$. SuperMat uses the released `supermat.pth` single-step checkpoint. Teamwork uses `decomposition_heterogeneous_sd3.safetensors` with the Stable Diffusion 3 Medium base. DiffusionRenderer uses `diffusion_renderer-inverse-svd` with a single conditioning frame and zero noise augmentation. Raw FireRed-1.1 uses the FireRed-Image-Edit-1.1 release without task-specific fine-tuning.

Output conversion. The wrappers export albedo, material, and available normal predictions to the common image convention before scoring. The Intrinsic Image Diffusion and RGB↔X wrappers raise input RGB values to the power 2.2 before inference; this method-specific preprocessing is separate from the scoring code. Metallic and roughness are placed in the red and green channels of the material PNG. SuperMat's ORM output supplies metallic from its blue channel and roughness from its green channel. For the separate scalar maps from RGB↔X, the first channel is used; DiffusionRenderer scalar maps are formed by averaging their RGB channels. IDArb's normal adapter flips the predicted x component before exporting to the camera-normal convention used by the evaluator. The common evaluator does not fit a rotation to normal predictions. Sampling and averaging settings are method-specific; in particular, the ten-sample averaging used for Intrinsic Image Diffusion is not applied to the other baselines.

B.2 Prediction alignment and valid regions

The evaluation reads the RGB channels of the saved prediction PNGs and scales their values to $[0, 1]$ by division by 255. Prediction alpha channels do not define the scoring mask. A prediction must match its declared output size; outputs below the ground-truth resolution are resized once to 1024×1024 with bicubic interpolation. This resizing is applied to encoded RGB values, including normal maps, before normal vectors are decoded.

For DTC, the albedo mask is the ground-truth albedo alpha channel greater than 128 in 8-bit values. The metallic and roughness mask additionally requires the material PNG alpha channel to exceed 128 and its material-alpha channel to be at least 128. The normal mask intersects the albedo foreground with the normal-valid mask, then applies two iterations of four-neighbor erosion. Each expected view must contain at least 500 valid pixels for the evaluated attribute.

For Polyhaven-315, all masks also intersect the source-image foreground and the rendered object mask. Albedo uses the albedo alpha channel; metallic and roughness use the material coverage and material-alpha masks. Normals use the normal-valid mask with the same erosion rule. Nonempty attribute masks contribute to the reported evaluation subset even when they contain fewer than 500 pixels. Empty masks contribute no pixels to that attribute's score. These masks depend on the reference data, not on the method being evaluated.

B.3 METRICS AND AGGREGATION

For each object, we pool valid-pixel statistics across its evaluated views before computing a score. Albedo PSNR uses all three color channels and a peak value of one. Metallic and roughness RMSE are computed separately on the red and green channels of the material output. Thus, the reported RMSE is an average of object-level RMSE values, rather than the square root of a dataset-wide mean squared error.

For scale-invariant PSNR, one scalar is fitted jointly over all valid RGB values and views of each object:

$$c^* = \frac{\sum_k \widehat{A}_k A_k}{\sum_k \widehat{A}_k^2}, \qquad e_{\mathrm{SI}} = \frac{1}{n} \sum_k (c^* \widehat{A}_k - A_k)^2. \tag{7}$$

Here, k indexes the n valid scalar color values. If the denominator is zero, the scale is set to one. The scaled prediction is not clipped again before scoring. Both PSNR variants are capped at 120 dB when their mean squared error is at most 10^{-12}.

Normals are represented in camera coordinates, with x pointing right, y up, and z toward the camera. Predicted and reference RGB values are mapped to vectors by $\mathbf{n} = 2\mathbf{c} - 1$ and normalized. Angular errors are computed in degrees from the clipped dot product of the unit vectors. A predicted vector with norm below 10^{-6} receives an error of 180° instead of being excluded.

Object scores are averaged with equal weight within each dataset. Combined results weight each dataset mean by its number of scoreable objects for the corresponding metric. The material-alpha output is not evaluated as a separate attribute. A dash in a comparison table denotes an unreported attribute; it is not treated as a zero error.

C ADDITIONAL COMMUNICATION ABLATIONS

The backbone adaptation study is reported in Sec. 4.3 and Table 2. The main comparison in Table 1 includes PBRSense with and without communication at 20,000 training steps. Here, we provide additional comparisons of the communication structure at a fixed budget of 4,000 steps.

Controlled settings. All variants start from FireRed-Image-Edit-1.1, use the same training data and native 1024^2 resolution, and update backbone blocks 29–49 (zero-based, inclusive). All results use the fixed 4,000-step checkpoint and the same single-step inference protocol. The variant labeled "w/o communication" disables both paths throughout training and inference. GCO denotes gated cross-only communication. GCO-Local and GCO-Global retain only the local and global path, respectively, from the start of training. The simple mixer applies joint attention along the attribute axis and serves as the symmetric mixing comparison.

Aggregation. Views are pooled within each object before computing its metric. We combine DTC and Polyhaven-315 by weighting each dataset mean by its number of valid objects for that metric. For albedo and material metrics, the weights are 2,000 and 315; for normal angle, they are 2,000 and 315. This gives an equal-weight mean over all valid objects across the two datasets.

Results. Table 5 reports the structure comparisons. The full communication module gives the best combined albedo PSNR, metallic RMSE, and roughness RMSE among the listed structures. At 4,000 steps, the full module reduces metallic RMSE from 0.1222 to 0.1171 and roughness RMSE from 0.1605 to 0.1580 relative to the no-communication variant, while increasing albedo PSNR from 20.208 to 20.318 dB. GCO-Local is slightly better on si-PSNR (24.054 vs. 24.051 dB) and normal angle (9.331° vs. 9.365°). These differences are small and do not establish the ranking under longer training.

D ADDITIONAL QUALITATIVE RESULTS

Fig. 7 and Fig. 8 show additional PBRSense results on the test datasets and in-the-wild photographs, respectively. These examples cover different object types and illustrate the recovery of fine details, including text.

Table 5: Communication structure at 4,000 steps with backbone blocks 29–49 updated. Local-only and global-only variants are trained with the other path disabled. Red and orange backgrounds indicate the best and second-best values in each column. PSNR is in dB and normal angle is in degrees.

Method	A-PSNR ↑	A-si-PSNR ↑	M-RMSE↓	R-RMSE↓	N-angle (°)↓
PBRSense	20.318	24.051	0.1171	0.1580	9.365
w/o communication	20.208	23.960	0.1222	0.1605	9.384
GCO-Local	20.279	24.054	0.1211	0.1588	9.331
GCO-Global	20.238	23.993	0.1222	0.1596	9.346
simple mixer	20.207	23.953	0.1215	0.1601	9.363

E Additional 3D Material Results

E.1 Material-prior injection

Visible surface observations. We keep the input mesh and camera fixed and project occupied surface-voxel centers from the 1024^3 grid into the 1024×1024 input view. The trusted image region is obtained by thresholding the observation alpha at $254/255$, resizing its mask with nearest-neighbor interpolation when needed, and eroding it with a square radius of two pixels. We construct a depth buffer from the nearest projected occupied-voxel center at each trusted pixel. A voxel with projected depth d_v is observed if it lies in front of the camera, projects inside the trusted image region, and satisfies

$$d_v \le d_{\min}(p(v)) + 1.5\delta, \tag{8}$$

where $p(v)$ is its projected pixel and $\delta = \sqrt{3}s_{\text{mesh}}/1024$ is the voxel diagonal after applying the mesh scale s_{mesh} used by the camera transformation. This depth shell admits nearby surface voxels while excluding deeper occluded regions. We bilinearly sample the predicted albedo and material maps at the observed voxel projections.

Latent material observations. The frozen texture encoder used by Pixal3D maps the partial sparse attributes to 32-channel texture latents on a 64^3 grid. We apply the texture model's latent normalization to obtain the observed target y_j at latent position j. Its observation weight $w_j \in [0, 1]$ is the fraction of occupied surface voxels in the corresponding 16^3 block that passed the visibility test. Unobserved positions have zero weight, and we define $y_j = 0$ there for notation. The full latent coordinate set is retained, so the generator predicts material in unobserved regions as well. The injected observations contain albedo, metallic, and roughness channel; predicted normal maps are not injected.

Correction during sampling. Let x_t be the current normalized texture latent and $\widehat{x}_0$ the clean-latent estimate from the frozen flow model. We correct this estimate using the observed target:

$$\overline{x}_{0,j} = \widehat{x}_{0,j} + \lambda w_j (y_j - \widehat{x}_{0,j}), \qquad \lambda = 0.75. \tag{9}$$

The correction is therefore weighted by surface coverage rather than applied uniformly to all visible latent positions. With the sampler's noise floor $\sigma_{\min} = 10^{-5}$, the corrected velocity and Euler update are

$$\begin{aligned} \overline{v}_t &= \frac{(1-\sigma_{\min})x_t - \overline{x}_0}{\sigma_{\min} + (1-\sigma_{\min})t}, \\ x_{t'} &= x_t - (t - t')\overline{v}_t. \end{aligned} \tag{10}$$

Correction is enabled for $t > 0.2$. We use 12 sampling steps and time rescaling with factor 3; under this schedule, all 12 model evaluations occur above the threshold. Both the raw and guided runs retain RGB conditioning and use guidance scale 1 and random seed 42. The geometry, camera, and sampling schedule are shared; the guided run adds the latent correction without training any model parameters.

E.2 Additional qualitative results

Fig. 9 presents qualitative comparisons of 3D material generation on several rendered objects. Fig. 10 shows comparisons of PBR attributes. In the displayed examples, using PBRSense predictions as a material prior makes the generated materials more consistent with the input appearance. The examples show improved metallic estimates and corrections to some albedo regions that Pixal3D estimates incorrectly.

F Failure Cases and Discussion

Below, we present several failure cases in which PBRSense struggles with locally dense details. In the two examples below, our method fails to faithfully recover the fine details of the mechanical structures in the input images and instead produces noise-like patterns. This issue may be related to the patch size and the limited number of denoising steps.

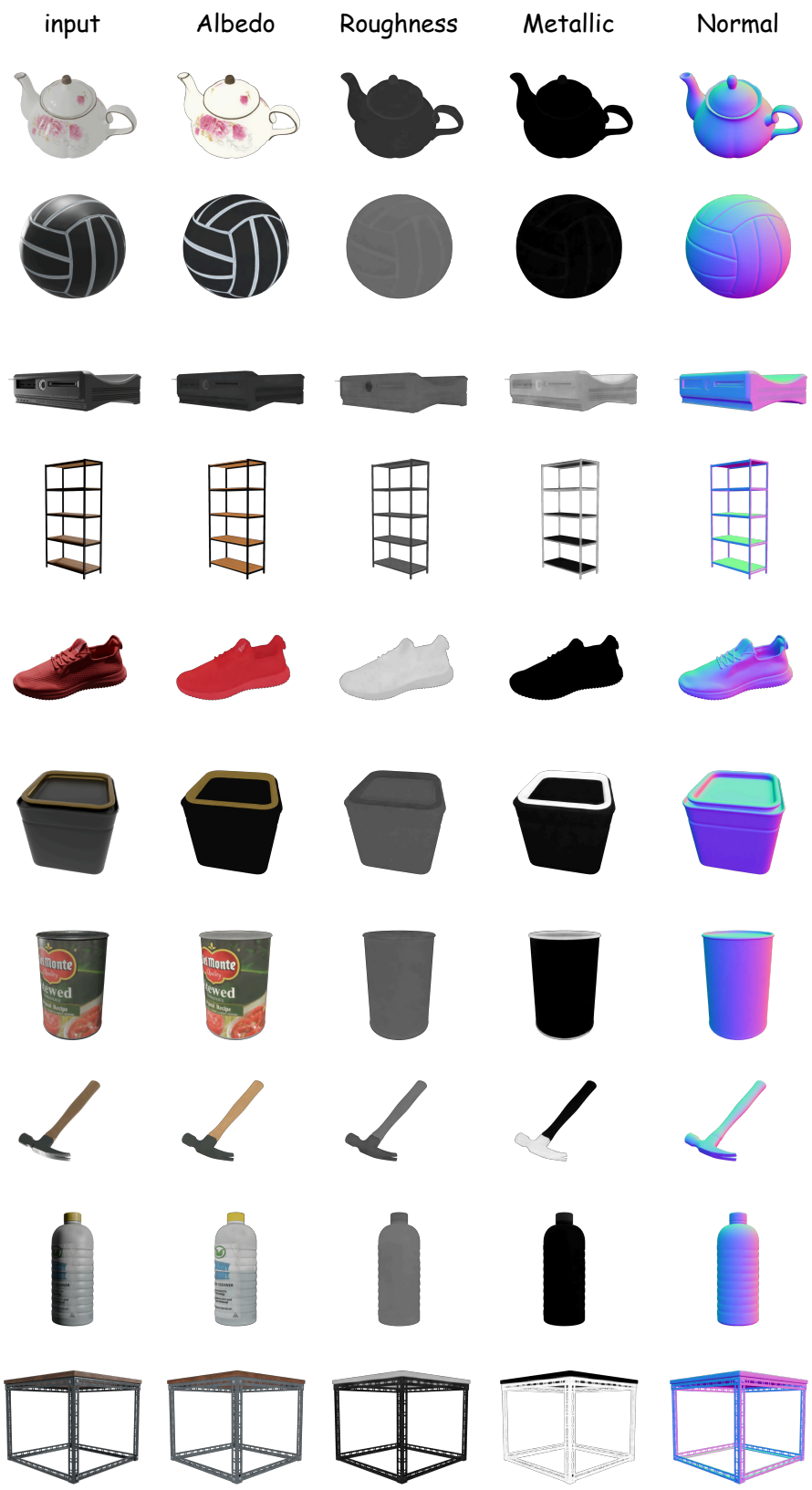

Figure 7: Additional results on the test dataset.

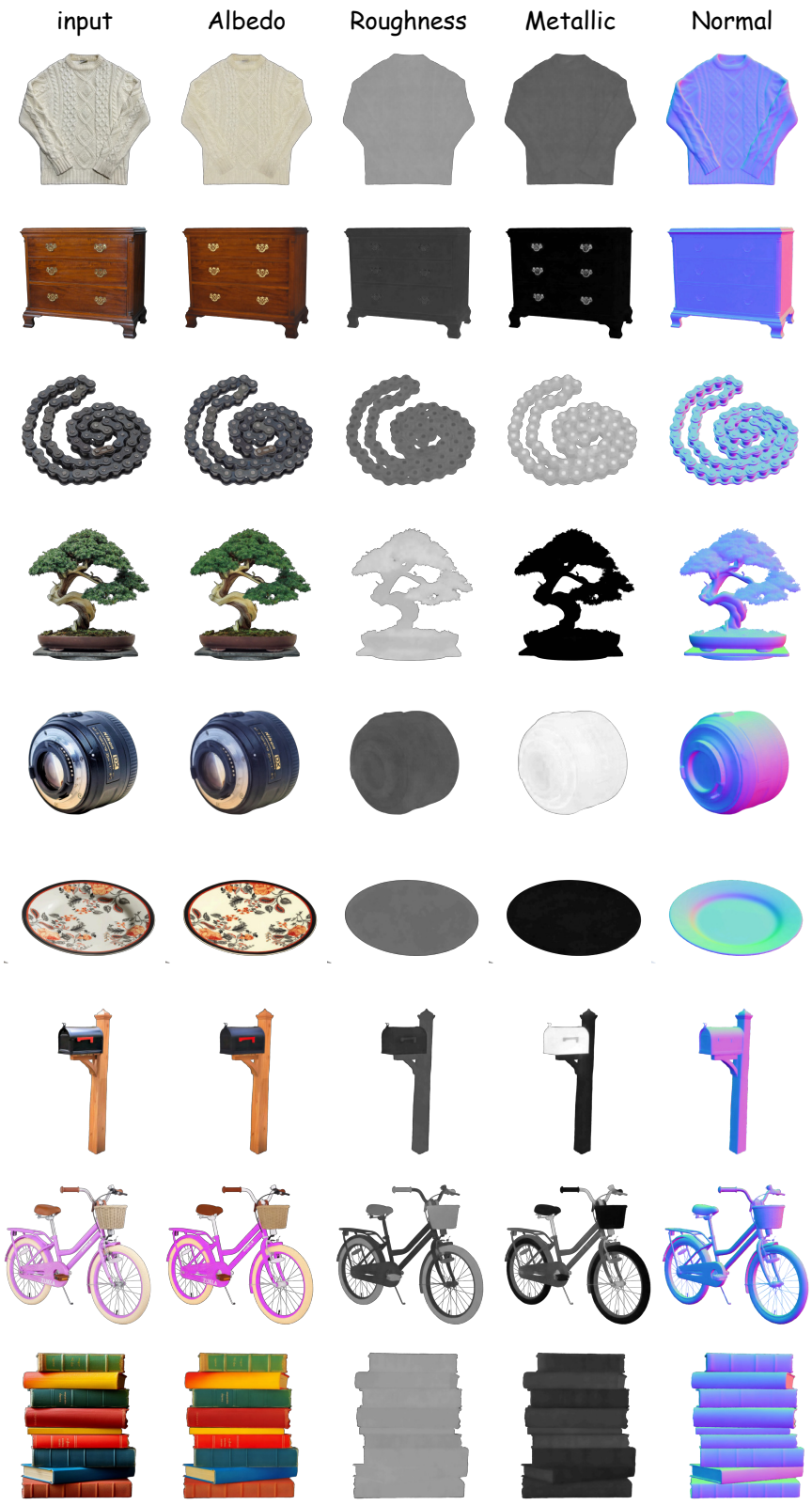

Figure 8: Additional results on in-the-wild photographs.



Figure 9: Qualitative rendered results for 3D material generation.

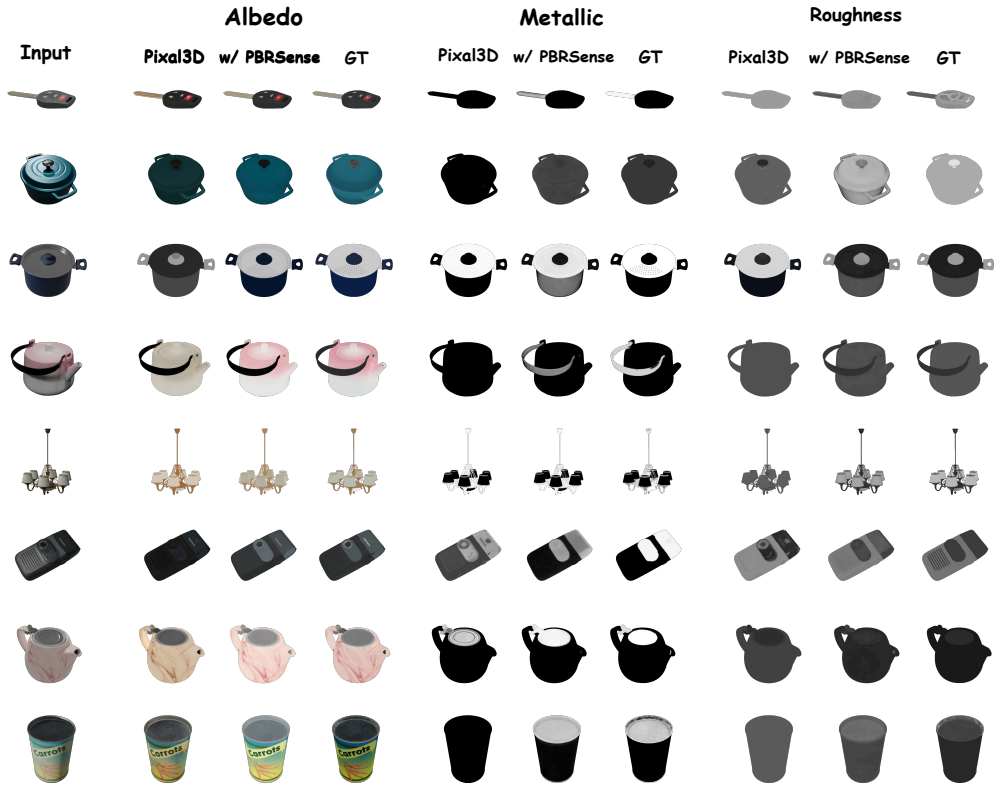

Figure 10: Qualitative 3D material attribute results.

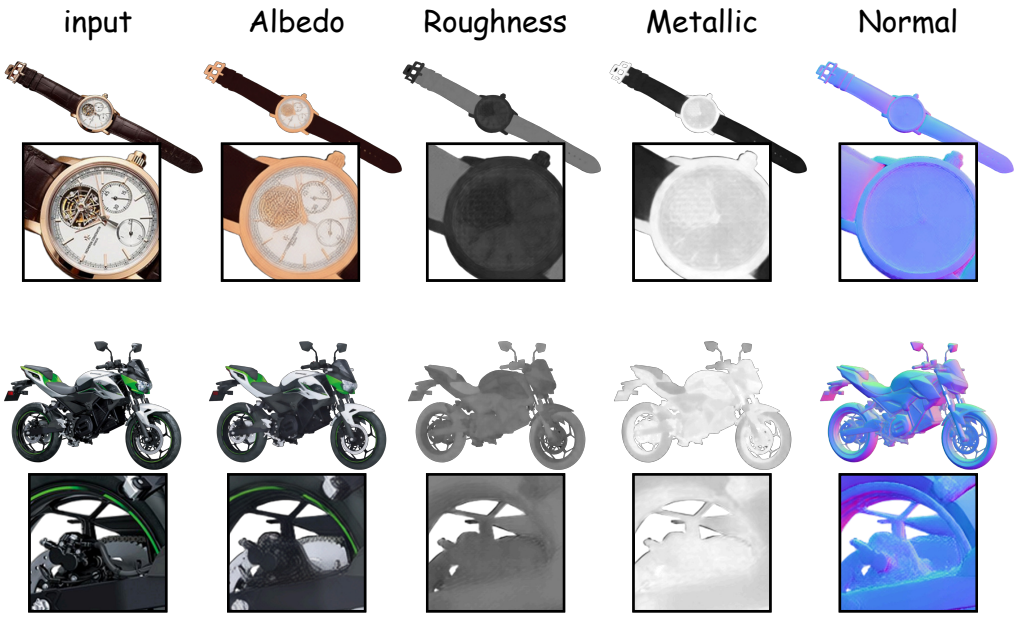

Figure 11: Failure cases