MVInverse: Feed-forward Multiview Inverse Rendering in Seconds

Supplementary Material

1. Architecture Details

We provide additional details of our network architecture. Our DINO encoder and the alternating-attention transformer are both initialized from the pretrained weights released by Pi3 [14], ensuring stable convergence and strong geometric priors. For the ResNeXt image encoder, we adopt the publicly available weights provided by [10], which offer robust, multi-level representations for detail preservation. Following Pi3 [14], our alternating-attention module contains 18 alternating attention blocks to effectively aggregate multi-view information.

We describe the feature-fusion strategy used to integrate outputs from the encoder stack. As shown in Figure 1, multi-scale features from the DINO encoder and the ResNeXt backbone are projected to a shared embedding space and fused through a lightweight convolutional refinement block. This design preserves view-dependent details while maintaining geometric consistency across viewpoints.

2. Training Details

In the pretraining stage, we supervise albedo using a scale-invariant reconstruction loss. To account for the scale ambiguity in albedo prediction, we first estimate a per-channel scale factor that aligns the prediction to the pseudo-ground-truth using least-squares error minimization. Specifically, for a predicted albedo map $A \in \mathbb{R}^{H\times W\times 3}$ and ground-truth A^*, we compute the 3-channel scale factor as

$$s^* = \arg\min_{s\in\mathbb{R}^3} \|A \odot s - A^*\|_2^2, \tag{1}$$

where s is a 3D scale vector and $\odot$ denotes channel-wise multiplication. The scale-invariant albedo loss is then defined as

$$\mathcal{L}_{\text{mse}} = \frac{1}{N}\|A \odot s^* - A\|_2^2. \tag{2}$$

Since this loss is unstable at early iterations, we first warm up the network using a vanilla MSE loss for several epochs before switching to the scale-invariant formulation, following [2, 6].

Throughout the pretraining stage, both the DINO encoder and ResNeXt encoder remain frozen to preserve their pretrained representations. To accommodate varying input frame counts, we adopt a dynamic batch sizing strategy similar to VGGT. Each GPU processes up to 12 images, and each batch is formed by randomly sampling 2 to 12 images of the same scene. For datasets that provide only single-view images (e.g., PRID [13]), we replicate each monocular image along the batch dimension to match the required number of input frames, effectively treating them as multiple captures from a static camera. We train for 80 epochs on two A100 GPUs at a fixed long-side resolution of 518 (with the short side randomly sampled), with each epoch consisting of 1000 iterations.After this initial training, we freeze all attention blocks and continue training only the prediction heads at a long-side resolution of 770 for an additional 50 epochs. This fine-tuning stage focuses on improving high-frequency details while keeping the reasoning in backbones fixed. The full training pipeline requires approximately 3–4 days. For all stages, we use the Adam optimizer with an initial learning rate of 5×10^{-5}.

Table 1 summarizes all datasets and their corresponding statistics. While most of our benchmarks consist of indoor environments, we additionally include two outdoor datasets—MatrixCity [5] and Sekai (for which we use pseudo-labels generated by DiffusionRenderer)—as well as the object-centric ABO dataset [3]. Following the masking strategy used in [2], we apply masks to exclude unreliable regions during supervision. Specifically, if a dataset provides masks for specular or mirror surfaces, we use these masks and compute losses only within the masked regions. In the absence of such annotations, we instead mask out pixels whose albedo values fall below 0.01 or above 0.99 to prevent these non-informative values from affecting optimization.

3. More Quantitative Results

Coordinate Space for Normals. While material properties such as albedo, metallicity, and roughness are intrinsically view-independent, surface normals can be defined in either world-space or camera-space coordinate systems. We conduct a simple ablation study to evaluate the impact of these two representations, with results summarized in Table 2. Our findings indicate that predicting normals in camera-space yields better performance compared to world-space. Specifically, the world-space coordinate system is defined relative to the first camera view.

4. More Qualitative Results

In this section, we provide additional qualitative results to further demonstrate the effectiveness, robustness, and generalization capability of our feed-forward inverse rendering model.

Single-view Prediction. Figure 3, 4 presents supplementary single-view predictions of metallic and roughness maps on the InteriorVerse [17] dataset. Since these material

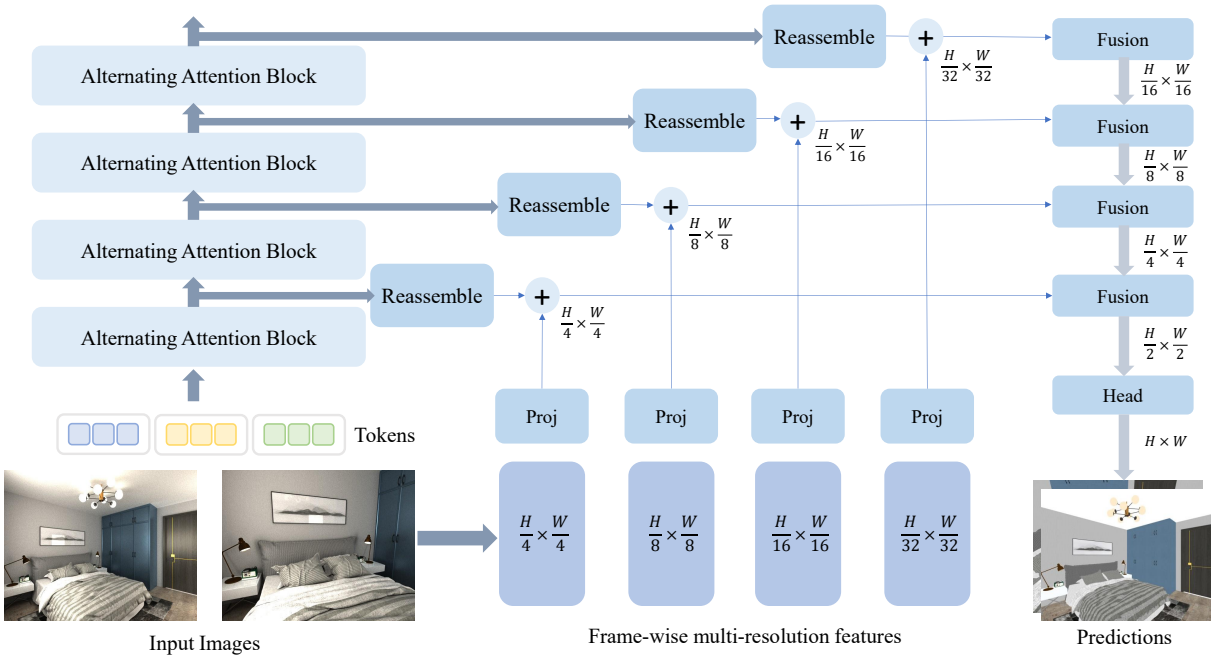

Figure 1. **Feature Fusion Detail** We illustrate the feature-fusion strategy used to integrate outputs from the encoder stack. Multi-scale features extracted from the DINO encoder (left in the figure) and the ResNeXt backbone (bottom) are projected into a shared embedding space and subsequently fused through a lightweight convolutional refinement block. The *Reassemble* and *Fusion* blocks follow the same design as in [11].

Table 1. **Summary of the dataset.** Sekai-Drone [7] uses pseudo-labels generated by DiffusionRenderer [8]. For the ABO [3], we use a randomly selected subset consisting of 100,000 images.

Dataset	# of Images	View	Scene Type	Intrinsic Types
Hypersim [12]	70,838	Multi-view	Indoor	Albedo, Normal, Diffuse Shading
Structured3D [16]	78,463	Multi-view	Indoor	Albedo, Normal
CGIntrinsic [6]	20,160	Multi-view	Indoor	Albedo
PRID [13]	21,478	Single-view	Indoor	Albedo
InteriorVerse [17]	52,769	Multi-view	Indoor	Albedo, Metallic, Roughness, Normal
MatrixCity [5]	44,804	Multi-view	Outdoor	Albedo, Metallic, Roughness, Normal
Sekai-Drone* [7]	127,246	Multi-view	Outdoor	Albedo
ABO* [3]	100,000	Multi-view	Object	Albedo, Metallic, Roughness, Normal

Table 2. Normal Comparison between camera-space and world-space normals. Both trained on Hypersim for 20k iterations with batch size 12. For the world-space model, we choose the first view as reference frame.

	NYUv2			ScanNet			iBims-1		
	mean↓	11.25° ↑	30° ↑	mean↓	11.25° ↑	30° ↑	mean↓	11.25° ↑	30° ↑
Cam-space	**15.926**	**59.622**	**84.925**	**14.357**	**64.479**	**87.517**	**16.738**	**65.988**	**83.551**
World-space	16.824	54.818	84.464	16.728	48.445	86.457	17.819	61.520	82.820

channels are not qualitatively visualized in the main paper, we include extended comparisons here against three recent state-of-the-art baselines—RGB↔X [15], IntrinsicImageDiffusion [4], and DiffusionRenderer [8]. As shown in the figures, our predictions are closer to ground truth compared to other methods.

We additionally provide more single-view prediction results on the test set of Hypersim [12] and Structured3D [16] in Figure 5, 6. We also provide in the wild results in IIW [1] dataset in Figure 7. These examples highlight our model's ability to generalize to diverse synthetic or real-world scenes.

Multi-view Prediction. Figure 8 show multi-view prediction results on the DL3DV [9] dataset, which serves as an out-of-distribution (OOD) benchmark due to its diverse large-scale scenes and complex geometry. For each multi-view input sequence, we visualize the predicted albedo, metallic, roughness, camera-space normals, and diffuse shading. Across all examples, our method demonstrates

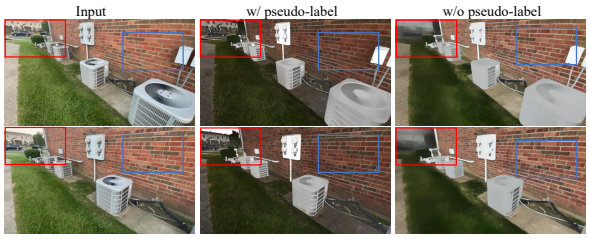

Figure 2. **Impact of pseudo-labels on albedo prediction. Red**: Without pseudo-labeled data generated by DiffusionRenderer [8], the model produces blurry patches near sky regions due to the lack of outdoor training samples. Incorporating pseudo-labels significantly improves prediction clarity in these areas. **Blue**: A side effect of introducing pseudo-labels is a tendency toward darker albedo estimates (e.g., on walls and floors).

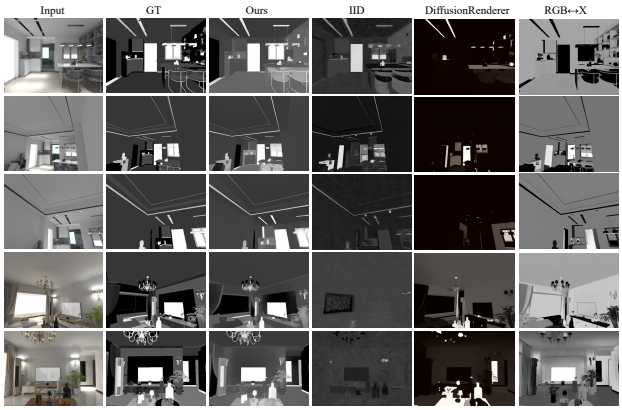

Figure 3. **Qualitative comparison for metallic prediction.**

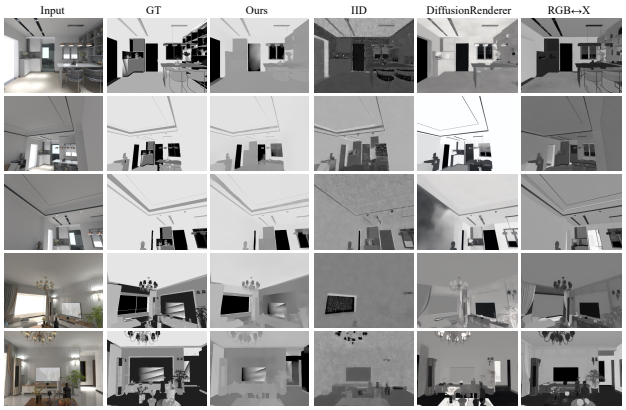

Figure 4. **Qualitative comparison for roughness prediction.**

strong cross-view consistency and stable material predictions.

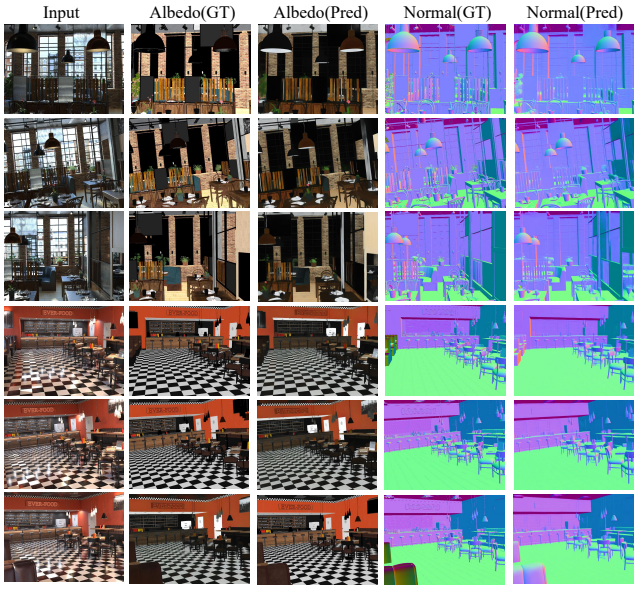

Figure 5. **Qualitative comparison on in-distribution Hypersim [12] dataset.**

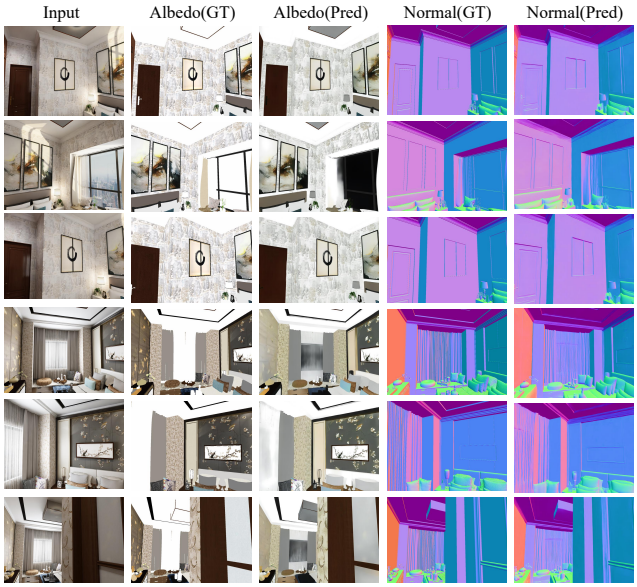

Figure 6. **Qualitative comparison on in-distribution Structured3D [16] dataset.**

5. Discussions and Limitations

Impact of pseudo-labeled data. We leveraged a subset of real-world pseudo-labels generated by DiffusionRenderer to train our albedo model, aiming to improve generalization in real-world outdoor scenes. Here we discuss the impact of pseudo-labels. As illustrated in Figure 2, incorporating pseudo-labels significantly enhances prediction quality in challenging regions, such as near-sky areas, which are oth-

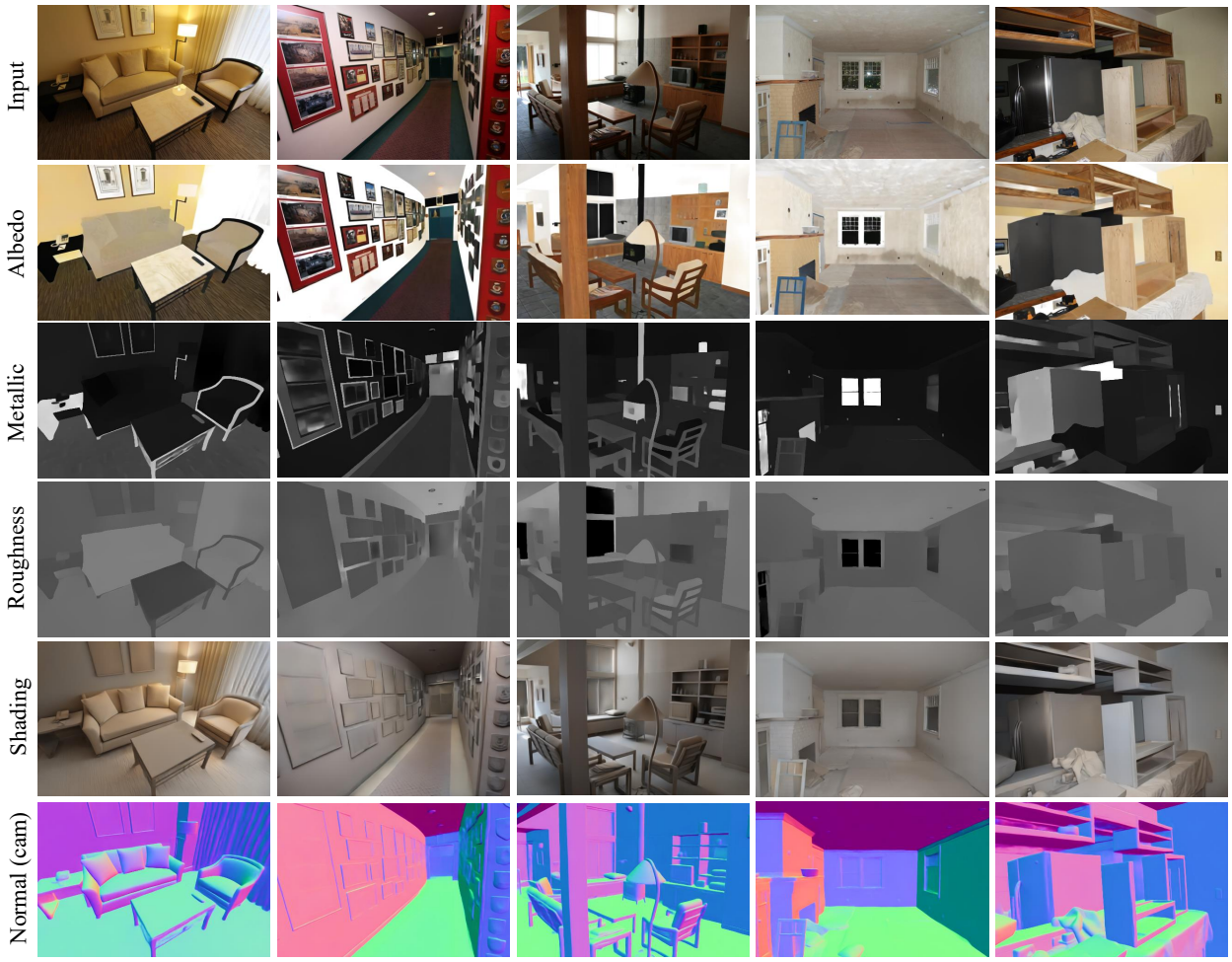

Figure 7. **More single-view results on IIW** We show additional albedo predictions on the IIW dataset to highlight the ability of our model to predict accurate and visually pleasing intrinsic images, including albedo, metallic, roughness, diffuse shading and camera-space normals.

erwise underrepresented in our indoor-dominated training data. Specifically, without pseudo-labels, the model tends to produce blurry or inconsistent albedo estimates near sky regions due to the lack of outdoor supervision. The introduction of pseudo-labels mitigates this issue, resulting in sharper and more physically plausible predictions. However, this approach also has some drawbacks: the model exhibits a tendency toward darker albedo estimates in certain regions, such as walls and floors, which results from the sub-optimal quality of the pseudo-labels. While the use of pseudo-labels is a pragmatic solution given the limited availability of annotated outdoor data, it remains a compromise. To further improve generalization to outdoor environments, a more comprehensive dataset covering diverse outdoor distributions would be required.

Performance Limitations. As our method is fundamentally data-driven, its performance can be constrained by the distribution of the training data when applied to out-of-distribution (OOD) inputs. This occasionally leads to predictions with inaccurate chromaticity. For example, in the top-most scene in Figure 8, the albedo of the wooden table is predicted darker than expected, while in the middle scene, the predicted albedo of the wooden floor exhibits noticeable tonal variations. However, it is important to note that such behavior is a common limitation of data-driven approaches; for instance, Figure.5 in the main paper demonstrates a similar drawback of DiffusionRenderer (the picture with a white cat on the floor). Beyond expanding the diversity of training data, this issue might also be mitigated by explicitly incorporating chromaticity information into the albedo prediction process, following the approach of [2].

Moreover, labeled data for metallic and roughness predictions are much sparser compared to albedo, with annota-

tions available only in a few datasets such as MatrixCity, InteriorVerse, and ABO. This scarcity significantly limits the model's generalization capability, leading to inconsistent or unreliable predictions for these material properties. For example, the model rarely encounters trees with ground-truth metallic or roughness labels, resulting in blurry and questionable predictions in these areas, as illustrated in Figure 9.

Furthermore, existing datasets often employ differing modeling conventions for complex materials like specular and mirror surfaces.For instance, Hypersim [12] typically assigns near-zero albedo values to these regions, whereas Structured3D [16] tends to represent them with high reflectance values. This divergence in annotation strategies introduces a degree of ambiguity during training, which can manifest as grayish or less defined outputs in the model's predictions (see the window area in Figure 6). Since our model is discriminative, these conflicting supervisory signals drive the optimization toward a median value, leading to blurred results in affected regions. While we mitigate extreme cases by masking values below 0.01 or above 0.99, the inherent variance in how specular surfaces are handled remains a challenge for achieving perfectly sharp predictions.

References

[1] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33(4): 1–12, 2014. 2

[2] Chris Careaga and Yağız Aksoy. Colorful diffuse intrinsic image decomposition in the wild. *ACM Trans. Graph.*, 43 (6), 2024. 1, 4

[3] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21126–21136, 2022. 1, 2

[4] 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*, pages 5198–5208, 2024. 2

[5] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3205–3215, 2023. 1, 2

[6] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *Proceedings of the European conference on computer vision (ECCV)*, pages 371–387, 2018. 1, 2

[7] Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, et al. Sekai: A video dataset towards world exploration. *arXiv preprint arXiv:2506.15675*, 2025. 2

[8] Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Chih-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, et al. Diffusion renderer: Neural inverse and forward rendering with video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26069–26080, 2025. 2, 3

[9] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 2, 6, 7

[10] Dhruv Kumar Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018. 1

[11] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 2

[12] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021. 2, 3, 5

[13] Yujie Wang, Qingnan Fan, Kun Li, Dongdong Chen, Jingyu Yang, Jianzhi Lu, Dani Lischinski, and Baoquan Chen. High quality rendered dataset and non-local graph convolutional network for intrinsic image decomposition. *Journal of Image and Graphics*, 2022. 1, 2

[14] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. 1

[15] Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. Rgb $\leftrightarrow$ x: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 2

[16] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *European Conference on Computer Vision*, pages 519–535. Springer, 2020. 2, 3, 5

[17] Jingsen Zhu, Fujun Luan, Yuchi Huo, Zihao Lin, Zhihua Zhong, Dianbing Xi, Rui Wang, Hujun Bao, Jiaxiang Zheng, and Rui Tang. Learning-based inverse rendering of complex indoor scenes with differentiable monte carlo raytracing. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 1, 2

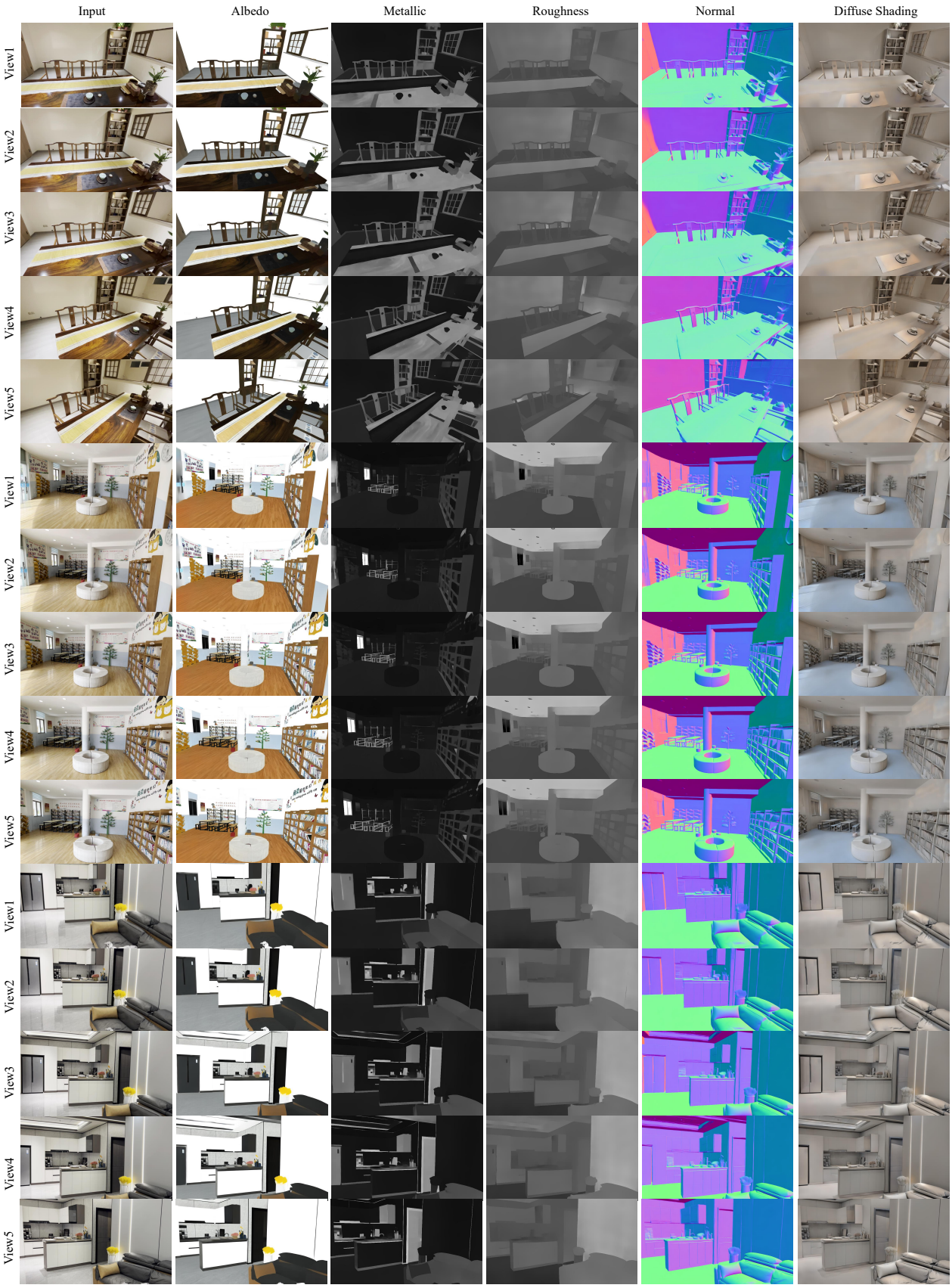

Figure 8. **Multi-view results on the DL3DV [9] dataset** demonstrating the cross-view consistency and generalization of our method.

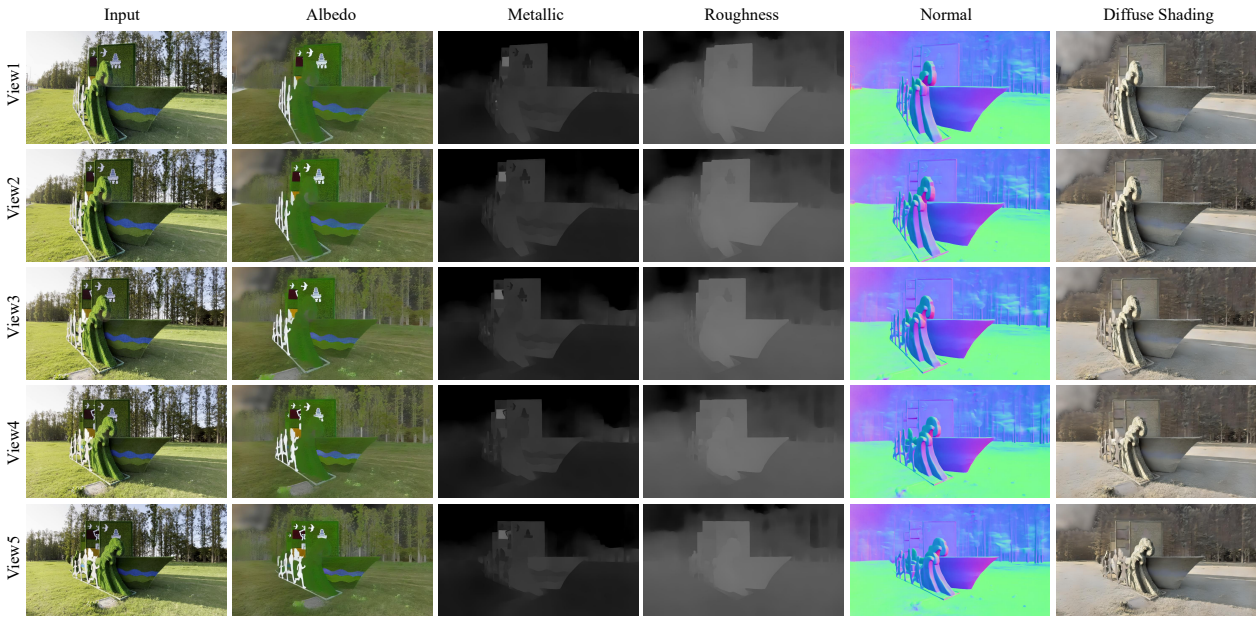

Figure 9. **Another example on the DL3DV [9] dataset** demonstrating its limitation when applied to outdoor scenes.