

---

# Supplementary Materials for “MaterialRefGS: Reflective Gaussian Splatting with Multi-view Consistent Material Inference”

---

Wenyuan Zhang<sup>1\*</sup> Jimin Tang<sup>1\*</sup> Weiqi Zhang<sup>1</sup> Yi Fang<sup>2</sup>

Yu-Shen Liu<sup>1†</sup> Zhizhong Han<sup>3</sup>

School of Software, Tsinghua University, Beijing, China<sup>1</sup>

Center for AI and Robotics (CAIR), NYU Abu Dhabi, UAE<sup>2</sup>

Department of Computer Science, Wayne State University, Detroit, USA<sup>3</sup>

{zhangwen21, tangjm24, zwq23}@mails.tsinghua.edu.cn

yfang@nyu.edu liuyushen@tsinghua.edu.cn h312h@wayne.edu

## 1 Implementation Details

**Multi-view Material Consistency.** We impose multi-view material consistency by enforcing the rendered material maps from different views to perform consistent appearance. Specifically, we pre-compute the neighboring views for each camera by selecting those views in terms of the distance between camera centers and angles between camera orientations. During each training iteration, for a reference view, we sample a source view from its neighboring-view list, and render material maps for both reference and source views. For every patch centered at every pixel in the reference view, we warp it onto the source view. We then compute the mean squared error for each pair of the corresponding patches and average the results among all patches to obtain the multi-view consistency loss  $\mathcal{L}_{mv}$ . Several existing methods compute multi-view warping loss based on ground truth RGB images, which is minimized to optimize scene geometry, such as surface depth and normals [3, 1, 4]. However, these methods suffer from cross-view inconsistent appearances, which limits the effectiveness of multi-view clues. In contrast, we decouple the view-dependent appearances into view-independent material maps. This allows us to directly impose the multi-view consistency loss on the rendered material maps, providing strong geometry clues for multi-view consistent material inference.

**Reflection Strength Prior.** The view selection strategy used for computing per-view reflection strength score is different from that in the multi-view material consistency. Since our goal is to highlight photometric variations across different views, we can not select too close views that have too similar appearances. To this end, we shoot rays from each camera center toward the center image pixel, and compute both the angular and distance between different viewpoint rays. For a given reference view, we select source views whose viewing angles fall between  $10^\circ$  and  $40^\circ$ , and whose ray distances fall between 0.1 and 0.5. These selected source views are then used to compute the reflection strength score for the reference view, as defined in Eq. 6 in the main paper.

Recall the reflection prior supervision (Eq. 7 in the main paper):

$$\mathcal{L}_{ref}(u, v) = \begin{cases} w_{ref} \cdot \Gamma(u, v) \cdot |M_0(u, v) - \Psi^M(u, v)| & \text{if } \Psi^M(u, v) < M_0 \\ 0 & \text{if } \Psi^M(u, v) \geq M_0 \end{cases}, \quad (1)$$

where  $M_0(u, v)$  is a pre-computed target metallic value, and  $\Gamma(u, v)$  is a binary mask indicating whether the supervision is applied at pixel  $(u, v)$ . We first predict a foreground mask [6] and then apply SAM2 [15] to segment the foreground into multiple regions, each representing a semantically

---

\*Equal contribution.

†The corresponding author is Yu-Shen Liu.

Table 1: Numerical comparisons of the indirect illuminance accuracy on synthetic datasets.

Scenes	Toaster		Coffee		Teapot		Tbell		Angel	
Methods	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
GaussianShader [9]	18.24	0.821	32.70	0.951	12.67	0.723	22.02	0.951	17.83	0.876
3DGS-DR [11]	18.93	0.839	33.55	0.953	14.81	0.792	26.26	0.968	18.96	0.901
Ref-Gaussian [18]	20.60	0.852	34.72	0.955	17.51	0.835	28.46	0.970	21.71	0.921
EnvGS [17]	19.20	0.844	33.17	0.954	13.04	0.776	23.94	0.962	18.17	0.892
Ours	25.97	0.942	35.37	0.959	21.50	0.912	31.14	0.985	25.56	0.957

Table 2: Numerical comparisons on Synthetic4Relight dataset.

	Novel View Synthesis			Albedo Accuracy			Roughness
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	MSE↓
InvRender [20]	30.74	0.953	0.086	28.28	0.935	0.072	0.008
GS-IR [13]	33.95	0.965	0.057	19.48	0.896	0.117	0.011
IRGS [7]	34.51	0.970	0.051	33.08	0.952	0.060	0.008
Ours	<b>34.74</b>	<b>0.971</b>	<b>0.049</b>	<b>33.57</b>	<b>0.954</b>	<b>0.058</b>	<b>0.007</b>

coherent part of the scene. Note that we do not constrain SAM2 to be object aware. Instead, we allow it to adaptively partition the image into regions based on semantic cues. Therefore, we are convinced that pixels within the same semantic region tend to share similar material properties. Based on the observation that highly reflective surfaces are often reliably identified early during optimization, we aim to propagate high metallic values to neighboring pixels. To this end, we use a binary indicator  $\Gamma(u, v)$  to exclude unreliable supervision from low-metallic regions. Specifically, if the average metallic value of all pixels in a region is smaller than 0.5, we set  $\Gamma(u, v) = 0$  for pixels in that region, otherwise we set 1. Additionally, we define  $M_0(u, v)$  as the average of the top 30% metallic values among the pixels within each region. The pixel with a metallic that is smaller than  $M_0(u, v)$  is encouraged to move towards  $M_0(u, v)$  with a weight  $w_{ref}$ . The weight  $w_{ref}$  here represents our confidence that a given pixel has a high metallic value. We compute the reflection strength prior offline every 10k iterations, which introduces no significant increase of the computational overhead.

**Hyperparameters.** In our loss function,

$$\mathcal{L} = \mathcal{L}_c + \lambda_{n-d}\mathcal{L}_{n-d} + \lambda_n\mathcal{L}_n + \lambda_{mv}\mathcal{L}_{mv} + \lambda_{ref}\mathcal{L}_{ref}, \quad (2)$$

we set  $\lambda_{n-d} = 0.05$ ,  $\lambda_n = 0.01$ , following previous methods [8, 2, 19]. And we set  $\lambda_{mv} = 0.015$ ,  $\lambda_{ref} = 0.01$ . Additionally, our code is build upon Ref-Gaussian [18], and all of our experiments are conducted on a single NVIDIA RTX 3090 GPU.

**Mesh Extraction.** We render depth maps for training views and then reconstruct the mesh using TSDF fusion [21, 8]. The voxel size is set to 0.004, and the SDF truncation threshold is set to 0.02. For unbounded scenes, we set a maximum depth range of 7.0 to include background regions.

## 2 More Evaluations

### 2.1 Comparisons on Indirect Illumination

We evaluate the rendering performance in indirect illumination regions labeled by ground truth masks calculated on the ground truth meshes from each view. The experiments are conducted on ShinyBlender [16] and GlossySynthetic [14] datasets, where there are ground truth meshes. The results are reported in the Tab. 1. We do not include LPIPS metric here, as it is designed for full image evaluation. Our method shows significant improvements over the baselines in indirect illumination regions, demonstrating the effectiveness of our environment modeling module.

### 2.2 Comparisons with Inverse Rendering Methods

It is not fair to directly compare our method with inverse rendering methods [5, 13, 10], as the illumination modeling approaches are much different. Specifically, our approach focuses on modeling reflection effects and therefore applies BRDF only to synthesize the specular components. In contrast,

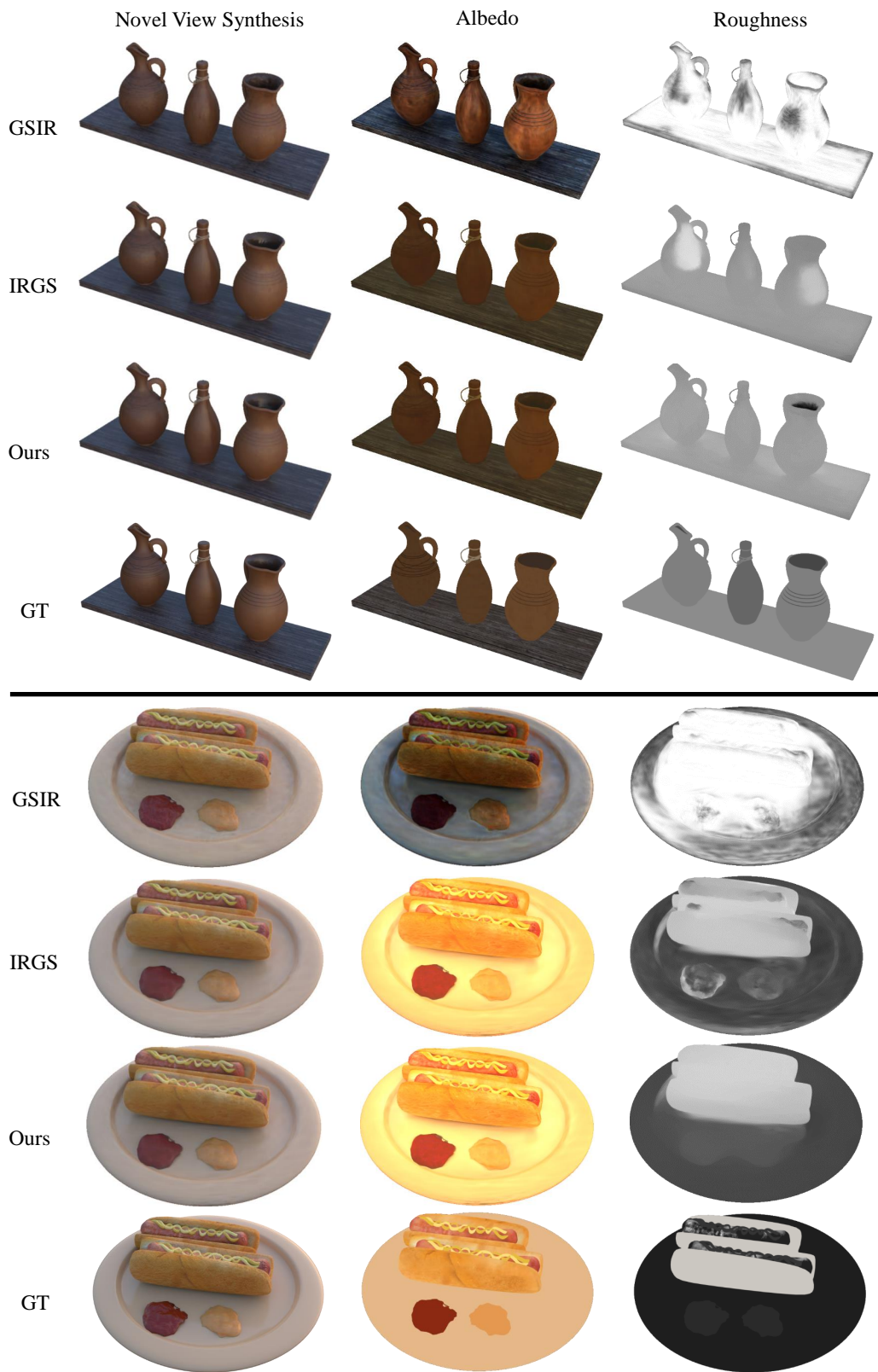


Figure 1: Visual comparisons of inverse rendering on Synthetic4Relight dataset.

Table 3: Training and inference resource consumption.

Methods	Memory (GB)	Training Time (h)	Inference FPS
GaussianShader [9]	13.20	0.48	28
Ref-Gaussian [18]	4.15	0.58	122
EnvGS [17]	5.19	2.6	54
Ours	5.36	0.84	104



Figure 2: Visual comparisons on synthetic datasets ShinyBlender [16] and GlossySynthetic [14].

inverse rendering methods incorporate diffuse color into the full BRDF evaluation. Moreover, they often omit the metallic term for simplicity. To numerically evaluate the effectiveness of our method in material decomposition, we integrate our multi-view material consistency constraint into the inverse rendering method IRGS [7] and impose it on the rendered albedo and roughness maps, while keeping all other experimental settings the same as the original IRGS. We report numerical comparisons on Synthetic4Relight dataset [20] in Tab. 2, where we achieve the state-of-the-art results on all metrics. Our visual comparisons in Fig. 1 show that we obtain significantly more uniform and smooth material decomposition results than baseline methods, such as the bottles and the hotdog plate.

### 2.3 Training Resources

We report our training time, inference efficiency and memory consumption in Tab. 3. Since EnvGS [17] optimizes an additional set of Gaussians to model the environment, it significantly increases the training and inference time. Our method achieves a balance between rendering quality, memory consumption, training speed and inference efficiency.

### 2.4 Additional Visual Comparisons

**Comparisons in Novel View Synthesis.** We provide more visual comparisons of novel view synthesis in Fig. 2, where our method accurately recovers reflection effects caused by environment lighting or indirect illumination without artifacts, such as the shadow of the coffee cup rim projected onto



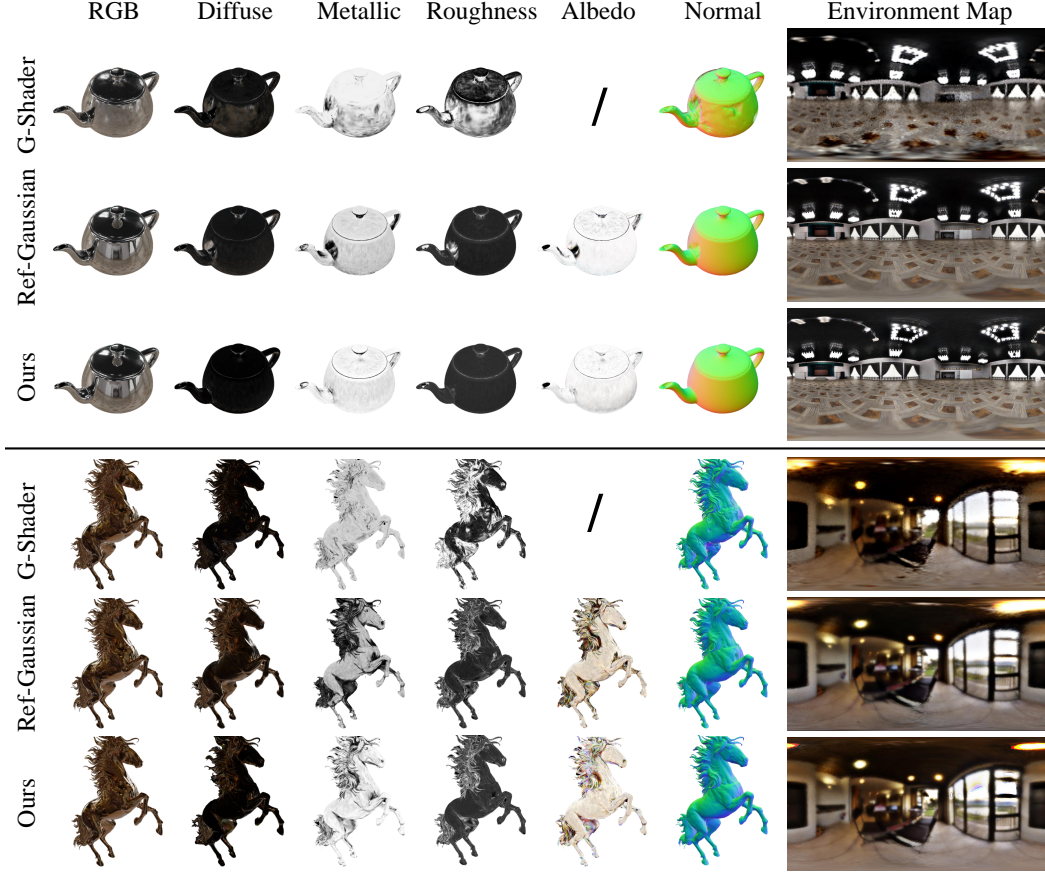


Figure 3: Visual comparisons of material decomposition results and the learned environment maps on synthetic datasets.

the liquid and the delicate textures on the metallic horse saddle. The superior results come from our multi-view consistent material inference with the improved environment modeling strategy.

**Comparisons in Material Decomposition.** We further provide additional visual comparisons of material decomposition and the learned environment maps on the synthetic datasets, as shown in Fig. 3. We do not include the albedo map of GaussianShader because it does not explicitly model the albedo term. The visualization results show that our method produces significantly more uniform and consistent material maps compared to the baseline methods. In addition, our approach identifies accurate material properties in regions with complex geometry, such as the mane of the horse, where Ref-Gaussian and GaussianShader produce incorrect metallic and roughness estimations. This improvement comes from our reflection strength prior, which is evaluated independently from the learning process and provides physically grounded guidance for material inference.

We also provide decomposition results on real-world Ref-Real dataset [16], as shown in Fig. 4. The baseline methods tend to explain high-frequency textures and inter-reflections between objects as diffuse terms and rely on view-inconsistent splatting as a compensation, such as the inter-reflections between two metallic spheres and the environment reflections on car windows. In contrast, our method thoroughly decouples diffuse and specular components through multi-view consistent material inference, resulting in physically grounded material properties and more realistic novel view synthesis. Furthermore, our learned environment maps show significant sharper and more detailed structures compared to baseline methods. This is because our approach relies entirely on querying the environment map to model direct illumination, which enables more accurate and grounded environment modeling. Note that the noisy colors in the environment maps correspond to positions where incident radiance is entirely determined by indirect lighting, and these querying directions in the cubemaps are never optimized.

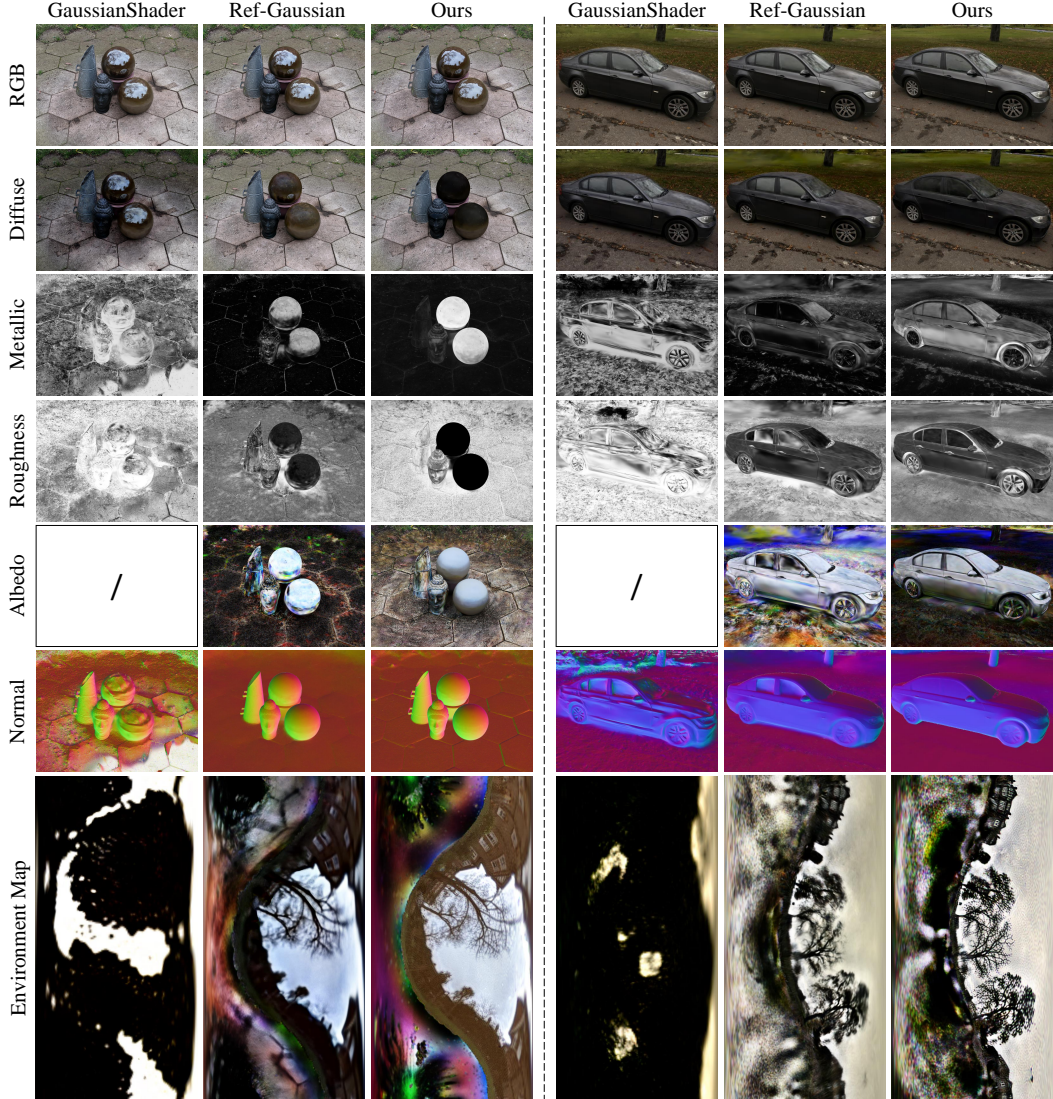


Figure 4: Visual comparisons in material decomposition.

**Visualization of Reflection Strength.** We provide more visualization cases in terms of reflection strength prior and our learned reflection strength term, i.e., metallic, as shown in Fig. 5. Our reflection strength prior accurately identifies highly reflective surfaces and provides explicit guidance for the learning of metallic component, which significantly improves material decomposition and novel view rendering quality.

## 2.5 Video Demonstration

We make a video to comprehensively demonstrate our results. In the first part of the video, we present qualitative results of surrounded novel view rendering on one representative case from each of Shiny-Blender [16], GlossySynthetic [14] and Ref-Real [16] datasets. Our method show clear advantages in modeling both realistic environment illumination and physically grounded inter-reflections. In the second part, we show the results of material decomposition on both synthetic and real-world datasets. Compared to Ref-Gaussian and GaussianShader, our approach achieves a more thorough decomposition, enabling accurate interactions between objects and environments for novel view synthesis. In the third part, we demonstrate our effectiveness on downstream applications including material editing and environment editing. By modifying the diffuse color, roughness attribute and

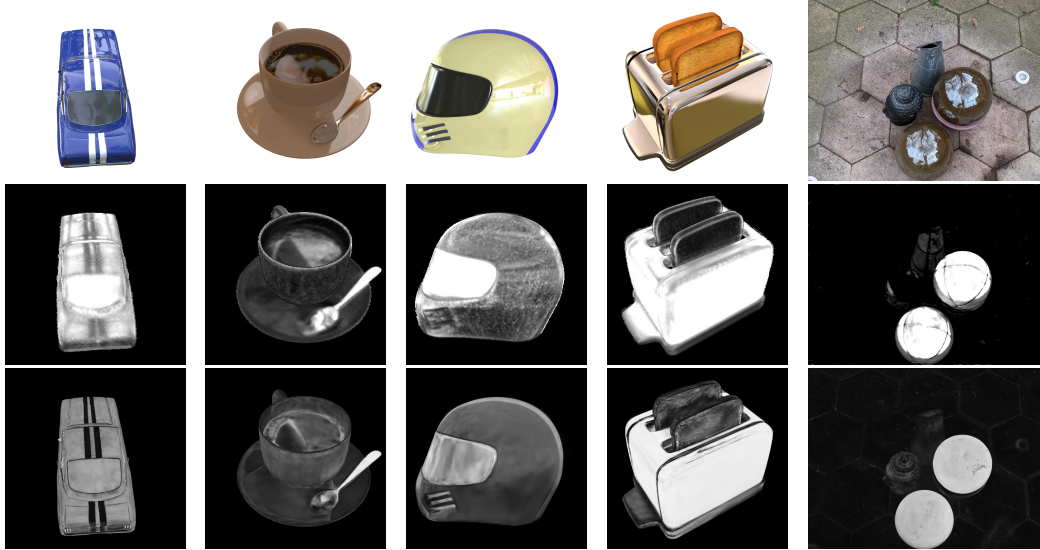


Figure 5: Visualization of reflection strength. From top to bottom: reference image, our predicted reflection strength prior, and our learned reflection strength term, i.e., metallic.

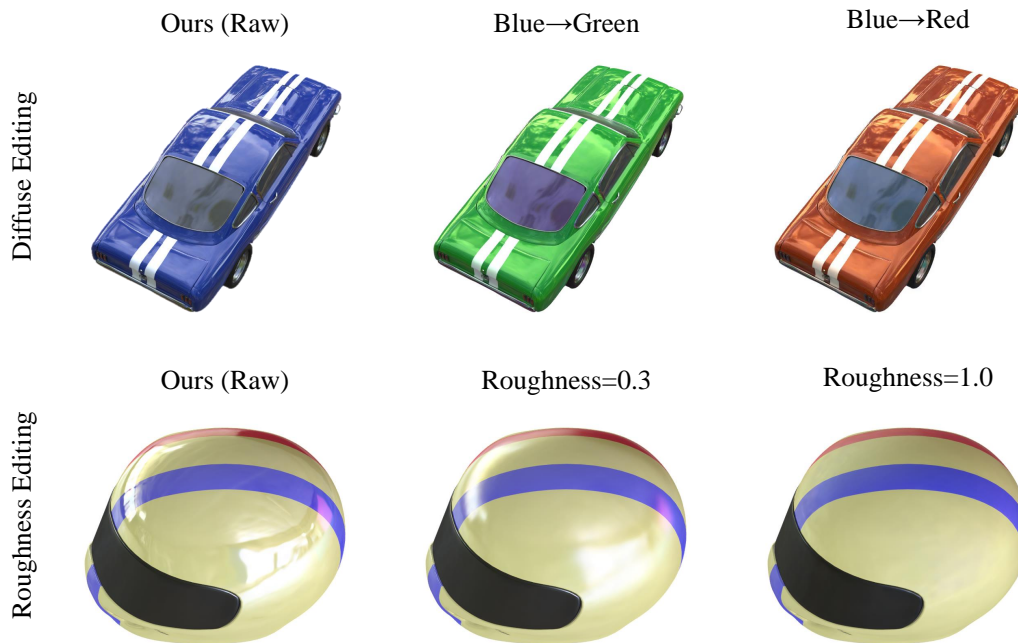


Figure 6: Visual comparisons in material editing.

environment maps, our method achieves visually realistic and coherent effects, showcasing our strong potential for diverse downstream tasks.

### 3 Applications

We further evaluate the effectiveness of our material decomposition in downstream applications. Similar to inverse rendering methods, our approach represents the scene using a set of material properties and an environment map. This representation allows us to manipulate these attributes to enable scene editing, including material editing and environment editing.



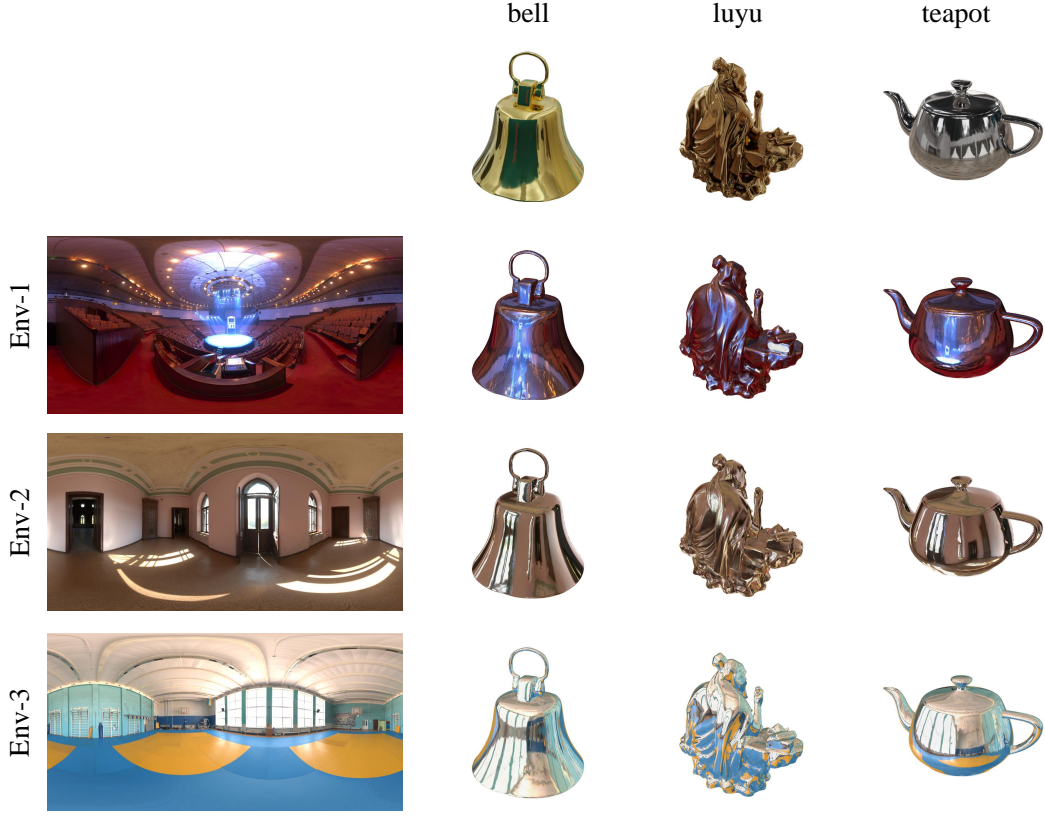


Figure 7: Visualization of the environment editing.



Figure 8: Illustration of our limitation.

### 3.1 Material Editing

We first try editing the diffuse color. Specifically, we swap the blue channel of each Gaussian’s diffuse color attribute  $c_d \in \mathbb{R}^3$  with the green and red channels, respectively. As shown in Fig. 6, the results demonstrate that we successfully alter the overall rendered style of the car to green and red tones.

We then try editing the roughness attribute. Specifically, we set the roughness of each Gaussian to a fixed value of either 0.3 or 1.0. As shown in Fig. 6, larger roughness value results in blurrier environment reflections, while a value of 1.0 causes the helmet to lose reflective effects entirely.



### 3.2 Environment Editing

In the environment Editing experiment, we use our trained checkpoints and replace the learned environment maps with specific ones. The newly synthesized results are shown in Fig. 7. Due to our accurate illumination decomposition through multi-view consistent material inference, the rendered objects exhibit realistic lighting effects under different environment maps.

## 4 Discussion

### 4.1 Limitations

Although our method is capable of modeling reflection effects through material decomposition, it is not specifically designed for transparent materials. We can only reconstruct some of the transparent materials, such as the car windows shown in Fig. 8. This is because these regions exhibit high reflectance with reflected radiance dominating over transmitted radiance, and our method interprets them as solid surfaces with high reflectivity. However, in regions where reflection is significantly weaker than transmission, such as the glass cup in Fig. 8, our method fails to recover plausible appearance and geometry.

### 4.2 Future Works

One of the future works is to extend our MaterialRefGS to adapt to modeling transparent surfaces. Another future work is to incorporate material priors from pretrained networks [12] to provide robust supervision in underfitting regions. This may further improve the accuracy of material decomposition and enhance the quality of novel view synthesis.

## References

- [1] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. PGSR: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2024.
- [2] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.
- [3] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6260–6269, 2022.
- [4] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geo-NeuS: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 35:3403–3416, 2022.
- [5] Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing. In *European Conference on Computer Vision*, pages 73–89. Springer, 2024.
- [6] Daniel Gatis. Rembg: A tool to remove images background. <https://github.com/danielgatis/rembg>, 2023.
- [7] Chun Gu, Xiaofei Wei, Zixuan Zeng, Yuxuan Yao, and Li Zhang. IRGS: Inter-Reflective Gaussian Splatting with 2D Gaussian Ray Tracing. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.
- [8] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.
- [9] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. GaussianShader: 3D Gaussian Splatting with Shading Functions for Reflective Surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5322–5332, 2024.
- [10] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. TensorIR: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2023.
- [11] Ye Keyang, Hou Qiming, and Zhou Kun. 3D Gaussian Splatting with Deferred Reflection. *ACM SIGGRAPH Conference Proceedings*, 2024.
- [12] 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.
- [13] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. GS-IR: 3D Gaussian Splatting for Inverse Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21644–21653, 2024.
- [14] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. NeRO: Neural geometry and brdf reconstruction of reflective objects from multiview images. *ACM Transactions on Graphics (ToG)*, 42(4):1–22, 2023.
- [15] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

- [16] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.
- [17] Tao Xie, Xi Chen, Zhen Xu, Yiman Xie, Yudong Jin, Yujun Shen, Sida Peng, Hujun Bao, and Xiaowei Zhou. EnvGS: Modeling view-dependent appearance with environment gaussian. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.
- [18] Yuxuan Yao, Zixuan Zeng, Chun Gu, Xiatian Zhu, and Li Zhang. Reflective gaussian splatting. *International Conference on Learning Representations*, 2025.
- [19] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022.
- [20] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18643–18652, 2022.
- [21] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*, 2018.