MaterialRefGS: Reflective Gaussian Splatting with Multi-view Consistent Material Inference

Wenyuan Zhang[1*] **Jimin Tang**[1*] **Weiqi Zhang**[1] **Yi Fang**[2]
Yu-Shen Liu[1†] **Zhizhong Han**[3]
School of Software, Tsinghua University, Beijing, China[1]
Center for AI and Robotics (CAIR), NYU Abu Dhabi, UAE[2]
Department of Computer Science, Wayne State University, Detroit, USA[3]
{zhangwen21,tangjm24,zwq23}@mails.tsinghua.edu.cn
yfang@nyu.edu liuyushen@tsinghua.edu.cn h312h@wayne.edu

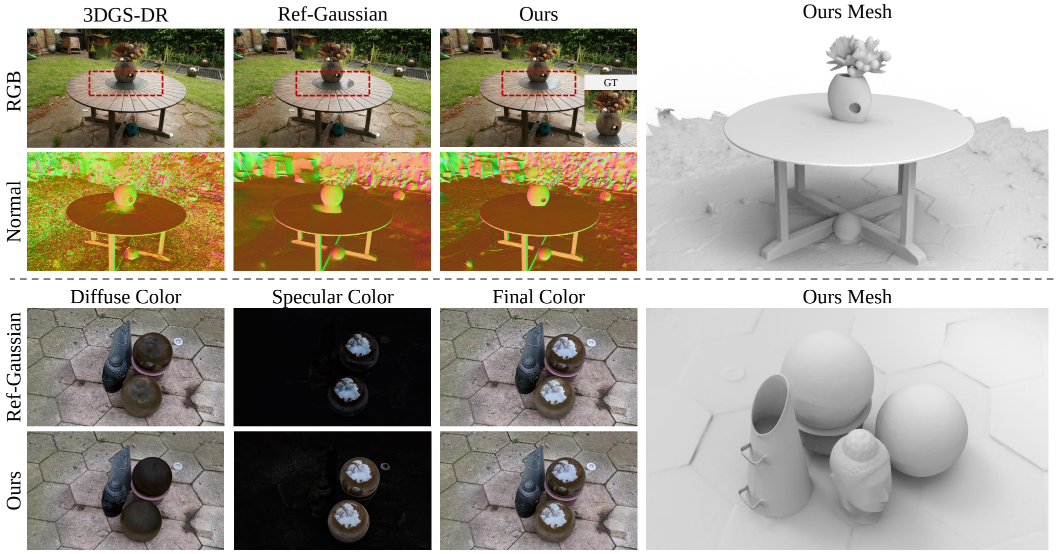

Figure 1: We highlight our novel view synthesis results on real-world scenes with complex reflection effects. Our **MaterialRefGS** outperforms the state-of-the-art methods in producing photorealistic renderings, disentangling physical materials, and recovering accurate scene geometry.

Abstract

Modeling reflections from 2D images is essential for photorealistic rendering and novel view synthesis. Recent approaches enhance Gaussian primitives with reflection-related material attributes to enable physically based rendering (PBR) with Gaussian Splatting. However, the material inference often lacks sufficient constraints, especially under limited environment modeling, resulting in illumination aliasing and reduced generalization. In this work, we revisit the problem from a multi-view perspective and show that multi-view consistent material inference with more physically-based environment modeling is key to learning accurate reflections with Gaussian Splatting. To this end, we enforce 2D Gaussians to produce multi-view consistent material maps during deferred shading. We also track photometric variations across views to identify highly reflective

[*] Equal contribution.

[†] The corresponding author is Yu-Shen Liu.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

regions, which serve as strong priors for reflection strength terms. To handle indirect illumination caused by inter-object occlusions, we further introduce an environment modeling strategy through ray tracing with 2DGS, enabling photorealistic rendering of indirect radiance. Experiments on widely used benchmarks show that our method faithfully recovers both illumination and geometry, achieving state-of-the-art rendering quality in novel views synthesis. Project Page: https://wen-yuan-zhang.github.io/MaterialRefGS.

1 Introduction

Learning scene appearance representations and recovering unseen views from multiple posed RGB images has been a long-standing task in computer vision and graphics [22, 36, 18, 64]. Recent advances in Neural Radiance Fields (NeRF) [36] leverage volume rendering to learn implicit scene representations for novel view synthesis. More recent efforts have been made by learning explicit 3D Gaussians (3DGS) [22] to achieve real-time rendering through a differentiable splatting procedure. Despite achieving photorealistic synthesis, 3DGS shows limited performance when confronted with complex reflective environments. This limitation arises from the contradiction between the simplistic geometric representations and the intricate shading mechanisms of real-world objects.

To tackle this problem, recent methods typically separate the rendering color into diffuse and specular components, and adopt inverse rendering frameworks [55, 11] to learn illumination decomposition through physically based rendering [40]. They endow each Gaussian primitive with learnable reflection-related properties, such as metallic and roughness [20, 24]. A splatting pass rasterizes these attributes into screen-space material maps [22, 18], followed by a lighting pass that evaluates a Bidirectional Reflectance Distribution Function (BRDF) [4] using material maps and environment lighting to synthesize the final image. This two-stage pipeline is known as deferred shading-based PBR [24, 56, 66]. It decomposes the view-dependent reflection effects into view-independent material properties by considering how the light interacts with the objects and environments, thereby improving the fidelity of novel views. However, the illumination decomposition poses several optimization challenges. First, inferring material properties from multi-view images is an ill-posed problem. All material parameters are optimized only through photometric loss after complex light transport, so multiple combinations of lighting and materials can explain the same pixels, which often leads to sub-optimal illumination decomposition [46, 58]. Second, the view-dependent behavior of 3D Gaussian representations conflicts with the goal of learning view-independent material properties. When the same physical attribute yields inconsistent appearances in different viewpoints, the BRDF struggles to infer accurate reflections from ambiguous observations, resulting in aliasing and degenerated illumination decomposition [10, 31].

To resolve these issues, we propose a novel approach that learns illumination decomposition for modeling reflections with 2DGS through multi-view consistent material inference. We first enforce Gaussians to produce multi-view consistent material buffers based on their physical attributes. This is achieved by aligning the projections of geometric surfaces on material maps from different views. We find that this constraint significantly improves illumination decomposition by limiting the view-specific overfitting. To better facilitate this process, we track photometric variations on object surfaces along the camera trajectory and quantify these variations as reflection scores. A spatial reflection fusion module is then applied to aggregate these per-view reflection scores into a multi-view consistent reflection strength prior. This prior is subsequently used as a supervision for the reflection strength attribute, i.e., metallic.

In addition, we observe that secondary reflection effects caused by inter-object occlusions often lead to degraded novel view synthesis. To address this, we propose an improved environment modeling strategy via differentiable ray tracing, which combines splatted indirect radiance and queried direct radiance with an on-the-fly estimation of occlusion probability. This approach effectively provides physically grounded signals in occluded regions, enabling more realistic indirect illumination. Our numerical and visual evaluations on widely used benchmarks demonstrate our superiority over the latest methods in terms of material inference and novel view synthesis. Our contributions are summarized as follows:

- We propose a novel approach to modeling reflections through Gaussian Splatting with multi-view consistent material inference, including multi-view material consistency constraint and reflection strength prior supervision. Our approach provides a new perspective for modeling reflections through physically grounded illumination decomposition.

- We introduce a differentiable environment modeling strategy through 2DGS based ray tracing, which enhances photorealistic rendering of indirectly illuminated regions caused by inter-object occlusions.
- We achieve state-of-the-art performance of novel view synthesis both in numerical results and visual comparisons on widely used benchmarks.

2 Related Work

2.1 Novel View Synthesis

The task of novel view synthesis aims to predict unseen views of a scene from a set of posed RGB images. Traditional methods typically rely on image interpolation [45] or inpainting [2] to generate novel views. With the rapid development of deep learning [70, 69, 67, 68, 5, 39, 28, 52, 32], novel view synthesis has gradually shifted toward learning-based approaches. Neural Radiance Fields (NeRF) [36, 62, 63, 65, 16, 51] pioneers this task by learning a mapping from 5D coordinates to volume densities as the scene representations. More recently, 3D Gaussian Splatting (3DGS) [22] has emerged as a new paradigm for real-time rendering by rasterizing Gaussian ellipsoids into images in a splatting manner. Various extensions support diverse scales and scenes through novel data structures such as hierarchies [23, 34] and octrees [44]. Others address sparse-view challenges by incorporating geometric priors [7, 19, 27, 17]. Beyond static scenes, some works also explore 3DGS for dynamic scenes [50], semantic-aware manipulation [42], and content generation [71, 60, 9]. Recent efforts aim to extract high-quality surfaces from 3DGS by flattening 3D Gaussians into 2D disks [18] and leveraging differentiable kernels to rasterize them into images. To better align Gaussians with object surfaces, regularization strategies such as depth-normal consistency [18, 15] and neural gradient supervision [61, 29, 35] are applied. In our work, we adopt 2D Gaussians as the foundational representation due to their effectiveness in modeling surface geometry and normals.

2.2 Modeling Reflections in NeRF and 3DGS

The view-dependent color representations used in original NeRF [36] and 3DGS [22], such as neural networks or spherical harmonics, struggle to capture high-frequency specular reflections that are commonly observed in real-world scenes. Existing solutions typically decompose the outgoing radiance into diffuse and specular components and blend them using learnable weights. To better model the specular reflections, some methods introduce directional encodings like Integrated Directional Encoding [47] and Gaussian Directional Encoding [26]. Other approaches extract accurate meshes to provide reliable normal for reflection modeling [33, 48, 13, 25]. Recent advances in 3DGS [22] offer new perspectives for addressing this challenge. Inspired by inverse rendering, Relight3DGS [11] assigns each Gaussian with physical properties such as metallic and roughness, and performs PBR on the Gaussians to synthesize the final image. Recent studies have proven that rendering per-Gaussian illumination attributes into material maps followed by deferred shading-based PBR [24, 56, 66, 20] yields better performance than per-Gaussian shading [11]. To improve environment interactions, some methods develop ray tracing techniques for Gaussians [37, 14, 53]. However, these methods primarily focus on per-view light-material interactions and neglect the globally consistent geometric information inherent in the multi-view settings. To fill in this gap, we propose leveraging multi-view cues to facilitate the disentanglement of material properties, enabling more accurate and realistic modeling of reflections in 3D Gaussians. Notably, our method is not equivalent to inverse rendering. Our goal is to model specular color through illumination decomposition, while relying on the Gaussians to render the diffuse color. In contrast, inverse rendering methods evaluate all lighting effects through BRDF, making them more suitable for quantitatively evaluating material decomposition and for relighting tasks. However, due to the complexity of learning diffuse component, these methods are limited to simple object-centric scenes.

3 Method

Given a set of posed RGB images $\{I_j\}_{j=1}^{N}$ that represents a scene with high reflections, we aim to synthesize a novel image from an unseen viewpoint. We learn a set of 2D Gaussians as the scene representations. We begin by introducing the preliminaries (Sec. 3.1), and then describe our multi-view material inference strategy (Sec. 3.2) and environment modeling strategy (Sec. 3.3). Finally, we detail the optimization procedure (Sec. 3.4). An overview of our method is provided in Fig. 2.

3.1 Preliminary

3D Gaussian Splatting (3DGS) [22] has become paradigms for learning 3D representations from multi-view images. A scene is represented by Gaussian functions $\{G_i\}_{i=1}^{K}$ with attributes like mean x_i, opacity o_i and scaling s_i. We also attach several reflection-related material attributes to Gaussians, including diffuse color $c_d \in \mathbb{R}^3$, albedo $a \in \mathbb{R}$, metallic $m \in \mathbb{R}$ and roughness $r \in \mathbb{R}$. We can then rasterize these Gaussians into images using

$$\hat{\Psi} = \sum_{i=1}^{N} \psi_i * o_i * p_i * \prod_{j=1}^{i-1}(1 - o_j), \tag{1}$$

where o_i, p_i are the opacity and screen-space probability [72] of the i-th Gaussian, respectively, and ψ_i denotes a selected attribute of G_i. By choosing different attributes such as c_{di}, a_i, m_i or r_i as ψ_i, we can render the corresponding material maps Ψ^{C_d}, Ψ^A, Ψ^M, Ψ^R, respectively. To facilitate surface reconstruction, 2DGS [18] flattens each 3D Gaussian into a 2D disk by setting one scaling dimension to zero. We adopt 2DGS as our base representation for better surface and normal alignment.

Our deferred shading-based PBR adopts a simplified version of the Disney BRDF model [4]. Given a viewing direction ω_o, the rendered color on the ray-surface intersection can be computed by

$$\begin{gathered} c(\omega_o) = (1 - m)c_d + L_s(\omega_o, a, m, r, n), \\ L_s(\omega_o, a, m, r, n) = \int_{\Omega} L_i(\omega_i) f_s(\omega_i, \omega_o)(\omega_i \cdot n) d\omega_i, \\ f_s(\omega_i, \omega_o) = \frac{DFG}{4(\omega_i \cdot n)(\omega_o \cdot n)} \end{gathered} \tag{2}$$

where $\omega_i, n, L_s, f_s, D, F, G$ denote the incident direction, normal, outgoing specular radiance, BRDF term, normal distribution function, Fresnel term and shadowing-masking term, respectively. Since computing the integral of L_s over the upper hemisphere Ω is computationally expensive, we adopt the split-sum approximation [38, 56, 31], which separates the integral into two components,

$$L_s(\omega_o, a, m, r, n) \approx \int_{\Omega} f_s(\omega_i, \omega_o)(\omega_i \cdot n) d\omega_i \cdot \int_{\Omega} L_i(\omega_i) D(\omega_i, \omega_o)(\omega_i \cdot n) d\omega_i, \tag{3}$$

where the first term can be precomputed using a, m, r and stored in a look-up table. The second term can be queried from a set of learnable environment cubemaps using reflected direction and r.

Similar to reflective Gaussian methods [20, 56], we decouple the diffuse component c_d and the specular component L_s, assigning the prediction of c_d to Gaussian rasterization. Unlike classical graphics pipelines that jointly infer diffuse and specular terms from albedo and roughness [31, 14], we find that such a design significantly increases optimization difficulty, especially in complex real-world scenes. Moreover, since the diffuse component is relatively insensitive to viewing direction, delegating its prediction to the Gaussians allows us to better disentangle reflection effects from illumination.

3.2 Multi-view Consistent Material Inference

Current methods model scene reflections by learning a set of Gaussians associated with material properties through PBR. The underlying assumption is that view-dependent specular variations can be disentangled into view-independent material attributes, while evaluating the final view-dependent appearance can be deputed to BRDF. However, this assumption often breaks down in practice, as shown in Fig. 8 (a), where the learned material maps exhibit significant discontinuities and inconsistency across different views. Since Gaussians cover different pixels and contribute varying weights across viewpoints during alpha blending, material parameters exhibit significant inconsistency from different perspectives. This inconsistency hampers accurate illumination decomposition, as the BRDF struggles to infer a global physical reflectance effect from such inconsistent material

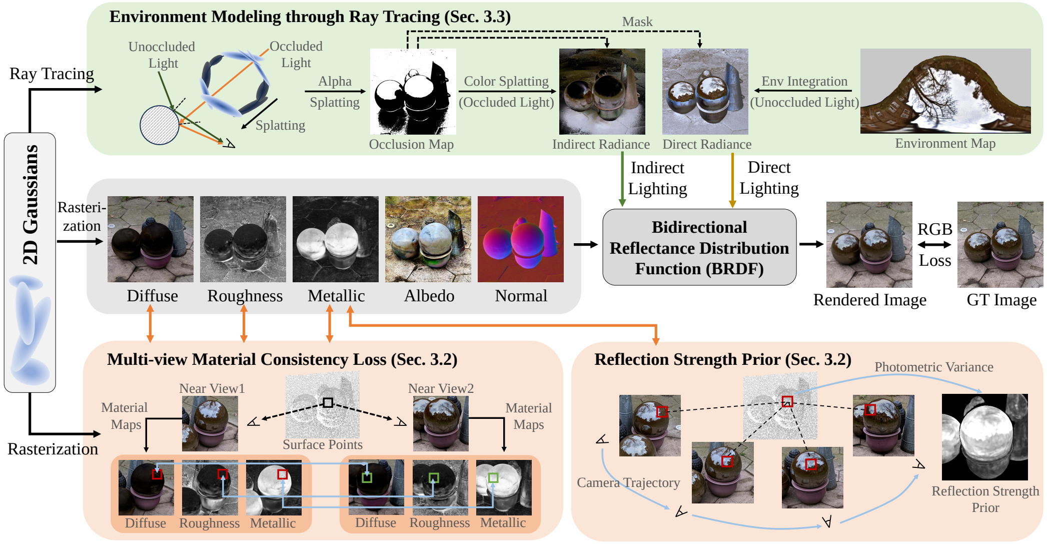

Figure 2: Overview of our method. We learn illumination decomposition by imposing multi-view material consistency constraint and reflection strength prior supervision on the rasterized material maps (Sec. 3.2). To facilitate this process, we introduce an environment modeling strategy through ray tracing with 2DGS, which effectively captures photorealistic incident lighting effects (Sec. 3.3).

observations. To overcome this obstacle, we propose to learn illumination decomposition by exploring multi-view consistent geometric clues as material inference constraints and guidance.

Multi-view Material Consistency. Based on the above analysis, we constrain the 2D Gaussians to produce multi-view consistent material maps, which is essential for accurate light-object-environment interactions inference. Specifically, for a surface point p visible from both viewpoint v_i and v_j, we want the projection $\pi(p)$ of p on the two material maps $\Psi_i(\pi_i(p)), \Psi_j(\pi_j(p))$ to be the same. Drawing inspiration from multi-view stereo methods [8, 6], we impose constraints on plane patches between adjacent views. We first sample a 7×7 pixel patch $P(\pi_i(p))$ around $\pi_i(p)$, back-project it into 3D space along v_i using the rendered depth d_i at $\pi_i(p)$, rotate it with the rendered normal at $\pi_i(p)$, and reproject it into Ψ_j using the rendered depth at $\pi_j(p)$, to form a warped patch $P'(\pi_j(p))$,

$$P'(\pi_j(p)) = H_{ij}P(\pi_i(p)), H_{ij} = K_j(R_{ij} - \frac{T_{ij}n_i^T}{d_i})K_i^{-1}, \tag{4}$$

where K_i, K_j are the intrinsic parameters of camera v_i and v_j, R_{ij}, T_{ij}, H_{ij} denote the relative rotation, relative translation and homography matrix from v_i to v_j, respectively. We then enforce consistency between the two patches on the material maps using an MSE loss,

$$\mathcal{L}_{mv} = \|\Psi_i[P(\pi_i(p))] - \Psi_j[P'(\pi_j(p))]\|_2. \tag{5}$$

In practice, we once select one reference view along with multiple source views, warp the patch from the reference view to each source view, and compute a loss for every reference–source patch pair. The multi-view material consistency constraint is imposed to diffuse, roughness and metallic maps Ψ^{C_d}, Ψ^R, Ψ^M. We do not use this constraint on the albedo map Ψ^A, since albedo contributes little to most non-reflective surfaces, and enforcing consistency is ineffective and potentially harmful.

Multi-view Consistent Reflection Strength Prior. Multi-view consistency on material maps alone is insufficient to provide clear guidance for illumination decomposition. Based on the observation that highly reflective surfaces exhibit significantly different appearances across different viewpoints [3, 13], we explore multi-view photometric variations as explicit supervision for reflection strength. We first apply luminance normalization [54] on the ground truth RGB images to eliminate brightness inconsistencies caused by shadows and textures. As illustrated in Fig. 3, given a reference view v_r, we select M nearby views $\{v_{ni}\}_{i=i}^M$ along the camera trajectory. For each pixel (u, v) in v_r, we sample a 3×3 patch $P_r(u, v)$ and warp it into the near views as $\{P'_{ni}(u, v)\}_{i=1}^M$ using Eq. 4. We then compute the averaged per-pixel variance among these patches as a reflection score for v_r, using standard deviation,

$$refscore = \frac{1}{9} \sum_{(x,y) \in P_r(u,v)} \text{std}(\Psi_r[P_r(u,v)], \Psi_{n1}[P'_{n1}(u,v)], ..., \Psi_{nM}[P'_{nM}(u,v)]), \tag{6}$$

where Ψ denotes the normalized RGB image and $\mathrm{std}(\cdot)$ is a per-pixel standard deviation operator. Since the reflected environments on the images may appear similar from certain viewing angles, the obtained per-view reflection scores are often inconsistent, as shown in Fig. 3 (a). To address this, we further introduce a spatial reflection fusion module to aggregate multi-view reflection scores. We back-project the per-view reflection scores into 3D space using depth maps to form a reflection score point cloud. For each query pixel, we perform a ball query [41] around its back-projected 3D location within the point cloud and compute the averaged top-K scores, thus yielding the final reflection strength prior w_{ref}, as illustrated in Fig. 3 (b). The prior indicates how likely the surface has a high reflection strength, therefore can serve as a weight of the constraint on material maps Ψ^M,

$$\mathcal{L}_{ref}(u,v) = \begin{cases} w_{ref} \cdot \Gamma(u,v) \cdot |M_0(u,v) - \Psi^M(u,v)| & \text{if } \Psi^M(u,v) < M_0 \\ 0 & \text{if } \Psi^M(u,v) \geq M_0 \end{cases}, \tag{7}$$

where $M_0(u,v)$ is a pre-computed target value, and $\Gamma(u,v)$ is a binary mask indicating whether the supervision is applied at pixel (u,v). To adapt our constraint to centric scene structures, we extract foreground masks using [12] and segment them into semantically meaningful regions using SAM2 [43]. For each region, $\Gamma(u,v)$ is set to 1 if the averaged reflection prior w_{ref} of all pixels in the region exceeds a threshold, and 0 otherwise. The target metallic value $M_0(u,v)$ is determined according to the averaged metallic value within each region. In a word, if a region more likely corresponds to a highly reflective surface, we apply a constraint that encourages higher metallic values for these pixels, as illustrated in Fig 3 (c). The intensity of the constraint is controlled by w_{ref}, allowing the supervision to be adaptively modulated based on our confidence of the surface reflectivity. Similar observations were discussed in [24, 56, 53], where highly reflective surfaces are often reliably identified early during the optimization.

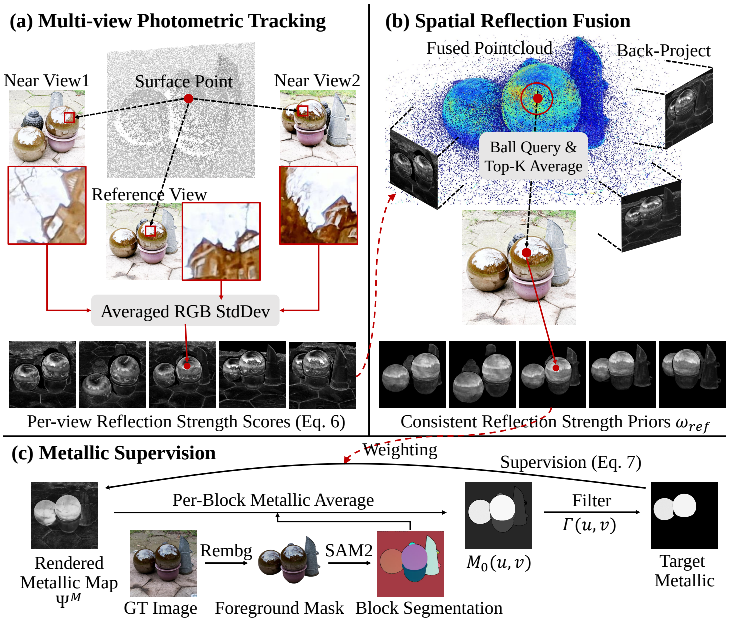

Figure 3: Illustration of computing reflection strength priors.

Normal Prior. We also incorporate monocular normal priors obtained by a pre-trained network [57] to supervise the normals rendered by 2DGS. We find it to be an effective cue for scene geometry inference during the early training stage.

3.3 Environment Modeling through Ray Tracing

When using split-sum approximation to model the specular illumination in Eq. 3, an incident light ω_i fails to retrieve a plausible radiance from the environment map if it is occluded by other objects in the scene. To address this issue, we decompose the incident radiance into direct and indirect components, and introduce an occlusion probability $O(\omega_i) \in [0,1]$ indicating how likely ω_i is occluded,

$$L_i(\omega_i) = L_{indirect}(\omega_i) + (1 - O(\omega_i)) \cdot L_{direct}(\omega_i). \tag{8}$$

The direct lighting $L_{direct}(\omega_i)$ can be obtained by querying a learnable environment map using the reflected direction and roughness. The environment map is a mip-mapped cubemap constructed with multiple roughness levels. To estimate the color of the indirect light which is occluded by other objects, we perform a Gaussian ray tracing [53]. Starting from a surface point, we trace a ray along the reflected direction and identify all intersected Gaussians, where each Gaussian has been transformed into a Bounding Volume Hierarchy (BVH) with two triangles. The intersected Gaussians

Table 1: Numrical evaluations on all four datasets. Best results are highlighted as 1st , 2nd , 3rd .

Datasets	ShinyBlender [47]				GlossySynthetic [33]			Ref-Real [47]			Mip-NeRF 360 [1]		
Methods	PSNR↑	SSIM↑	LPIPS↓	MAE° ↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RefNeRF [47]	33.13	0.961	0.080	18.38	25.65	0.905	0.112	23.62	0.646	0.239	-	-	-
ENVIDR [30]	32.88	0.969	0.072	2.74	29.06	0.947	0.060	23.00	0.606	0.332	-	-	-
3DGS [22]	30.37	0.947	0.083	-	26.01	0.886	0.089	23.85	0.660	0.230	27.21	0.815	0.214
GaussianShader [20]	31.94	0.957	0.068	7.00	27.11	0.922	0.082	23.46	0.521	0.257	-	-	-
2DGS [18]	29.58	0.946	0.084	-	26.07	0.918	0.088	24.15	0.661	0.292	27.03	0.805	0.223
3DGS-DR [24]	33.94	0.971	0.059	2.62	29.49	0.952	0.054	23.99	0.664	0.229	26.44	0.796	0.249
Ref-Gaussian [56]	35.04	0.973	0.056	4.59	30.08	0.957	0.050	24.61	0.685	0.252	26.62	0.781	0.272
EnvGS [53]	33.83	0.969	0.066	6.36	28.17	0.938	0.067	24.85	0.688	0.215	27.36	0.799	0.222
Ours	35.57	0.976	0.049	2.04	30.83	0.962	0.046	25.04	0.703	0.185	27.06	0.809	0.181

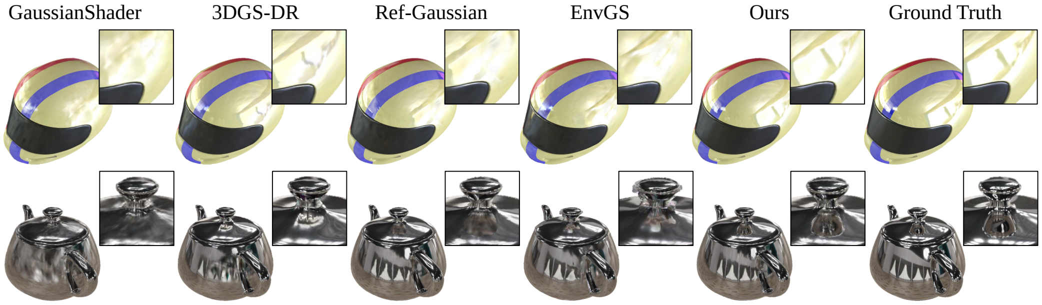

Figure 4: Visual comparisons on Synthetic Datasets. Our method successfully recovers fine-grained reflections on the helmet, as well as inter-reflection effects on the teapot.

are depth-sorted, and a splatting is performed to compute both the accumulated transmittance, denoted as $O(\omega_i)$, and the resulting lighting of the indirect illumination, denoted as $L_{indirect}(\omega_i)$,

$$L_{indirect}(\omega_i) = \sum_{i=1}^{N} c_d * o_i * p_i * \prod_{j=1}^{i-1}(1 - o_j) + c_r, O(\omega_i) = \sum_{i=1}^{N} o_i * p_i * \prod_{j=1}^{i-1}(1 - o_j), \tag{9}$$

where N is the number of the intersected Gaussians during ray tracing. We also incorporate a residual term c_r in the indirect radiance to account for noise and higher-order lighting effects [58, 56]. The ray tracing procedure naturally handles both occluded and unoccluded cases, where unoccluded rays yield $L_{indirect}(\omega_i) = O(\omega_i) = 0$. Therefore, we only need one ray tracing pass and one environment query to obtain the full incident radiance. Compared to Ref-Gaussian [56] which relies on an offline binary visibility indicator to separate direct and indirect terms and estimates indirect light solely through a residual color, our method evaluates occlusion in a fully differentiable manner. Moreover, this design allows Gaussians to participate in environment illumination modeling and be jointly optimized, leading to more physically grounded modeling and improved generalization.

3.4 Optimization

We train our method for a total of 30k iterations. We begin by training a 2DGS [18] with normal priors during the first 3k iterations to ensure geometric stability. After that, we incorporate PBR and our environment illumination modeling into the training process. At 10k iteration, we remove the normal prior to avoid potential bias from inaccurate predictions, and introduce our multi-view regularization terms. We also adopt normal propagation [24, 56] to propagate reliable normals to neighboring Gaussians for consistency and stability. The loss function can be written as

$$\mathcal{L} = \mathcal{L}_c + \lambda_{n-d}\mathcal{L}_{n-d} + \lambda_n\mathcal{L}_n + \lambda_{mv}\mathcal{L}_{mv} + \lambda_{ref}\mathcal{L}_{ref}, \tag{10}$$

where $\mathcal{L}_c = 0.8 * \mathcal{L}_{rgb} + 0.2 * \mathcal{L}_{D-SSIM}$ is the photometric loss commonly used in Gaussian-based methods [22, 56], $\mathcal{L}_{n-d}$ denotes the depth-normal consistency loss used in 2DGS [18], and $\mathcal{L}_n = |1 - N^T\hat{N}|$ is the normal prior loss. $\mathcal{L}_{mv}, \mathcal{L}_{ref}$ correspond to our multi-view consistency loss (Eq. 5) and reflection strength loss (Eq. 7), respectively.

4 Experiments

4.1 Experiment Settings

Datasets & Metrics. We evaluate the performance of our method on widely used benchmarks, including two synthetic datasets, ShinyBlender [47] and GlossySynthetic [33], as well as two real-

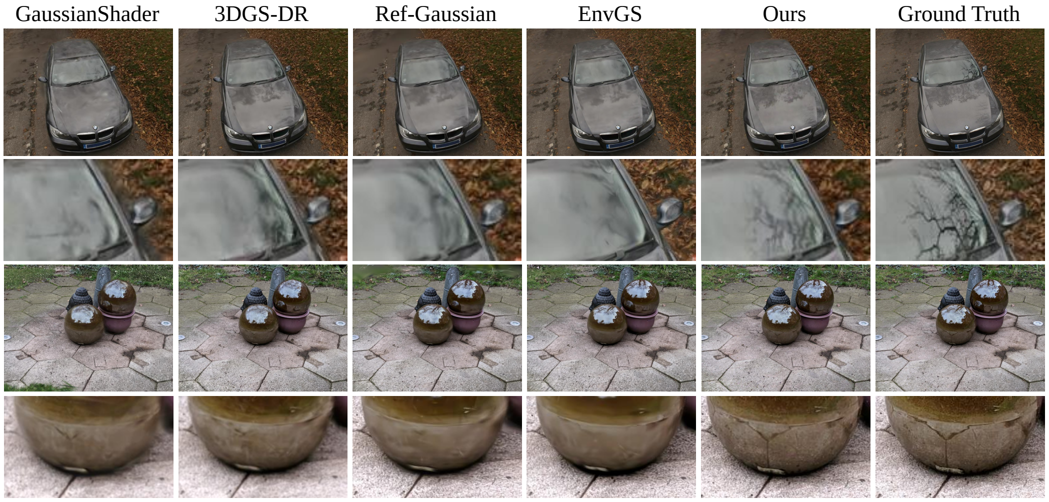

Figure 5: Visual comparisons in real-world Ref-Real [47] dataset. Our method accurately reconstructs reflection textures from surrounding environments on highly reflective surfaces.

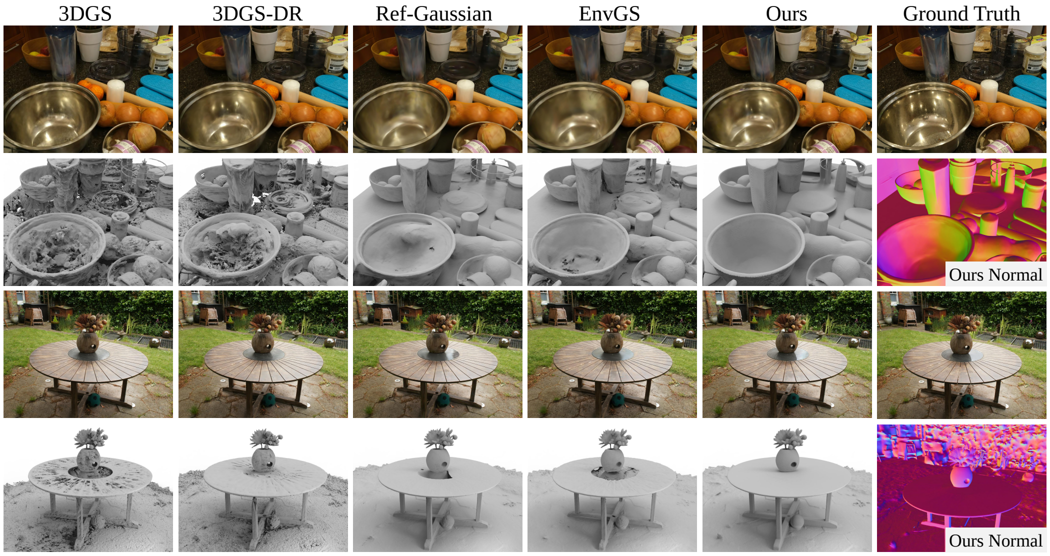

Figure 6: Visual comparisons in real-world Mip-NeRF 360 [1] dataset. Our method faithfully reconstructs highly reflective surfaces, such as the aluminum bowl and the metal plate.

world datasets, Ref-Real [47] and Mip-NeRF 360 [1]. All of these datasets contain challenging scenes with prominent reflective surfaces. To evaluate the quality of novel view synthesis, we report PSNR, SSIM [49] and LPIPS [59]. We also evaluate the accuracy of the predicted normals using Mean Angular Error (MAE).

Baselines. We compare our method with the state-of-the-art reflection modeling methods, including NeRF-based methods: Ref-NeRF [47], ENVIDR [30], as well as GS-based methods: 3DGS [22], GaussianShader [20], 2DGS [18], 3DGS-DR [24], Ref-Gaussian [56] and EnvGS [53].

4.2 Comparisons

Comparisons on Synthetic Dataset. We first evaluate our method on two synthetic datasets, ShinyBlender [47] and GlossySynthetic [33], and report the numerical results in Tab. 1, where we achieve the best performance across all metrics on both datasets. We further provide visual comparisons in Fig. 4, where our method accurately captures the environment reflections on the helmet. In addition, our approach effectively models secondary light reflections, such as the self-reflection of the teapot lid knob on the metallic lid, which benefits from our environment modeling.

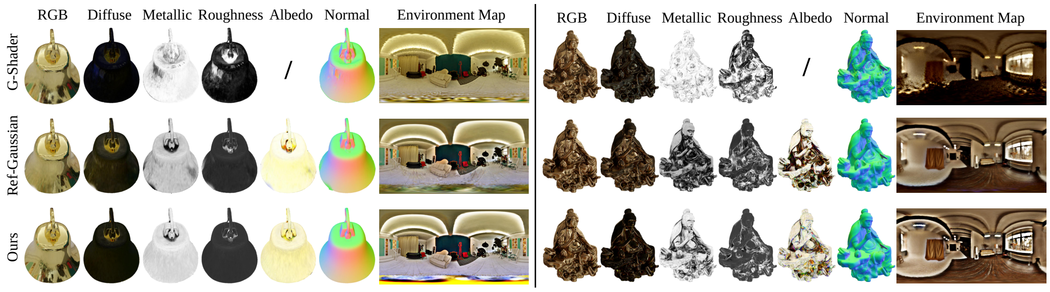

Figure 7: Visualization of illumination decomposition results and the learned environment maps. Our method produces more uniform and physically plausible material maps, along with sharper and more detailed environment maps. Best viewed with zoom in.

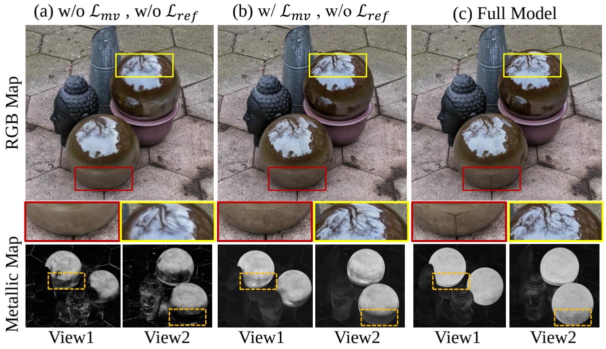
(a) w/o $\mathcal{L}_{mv}$, w/o $\mathcal{L}_{ref}$
(b) w/ $\mathcal{L}_{mv}$, w/o $\mathcal{L}_{ref}$
(c) Full Model
RGB Map
Metallic Map
View1
View2
View1
View2
View1
View2

Figure 8: Ablation study on multi-view consistent material inference strategies.

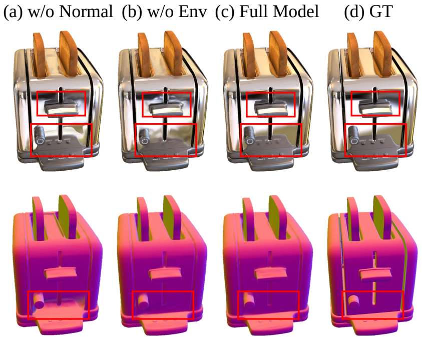

Figure 9: Ablation study on normal prior and environment modeling.

Comparisons on Ref-Real Dataset. We also report quantitative comparisons on the real-world dataset Ref-Real [47] in Tab. 1, where our method consistently achieves state-of-the-art performance across all metrics. Visual comparisons in Fig. 5 show that our method accurately reconstructs the reflection textures from the surrounding environments on highly reflective surfaces, such as tree branches reflected on car windows and ground seams reflected on the metallic sphere.

Comparisons on Mip-NeRF 360 Dataset. We further evaluate our method on the more challenging real-world dataset Mip-NeRF 360 [1], as reported in Tab. 1. Existing reflective 3DGS methods often show degenerated performance on such complex environments with few reflective surfaces. In contrast, our method achieves competitive results and outperforms all baselines in terms of LPIPS, highlighting our strong generalization ability. Visual comparisons are provided in Fig. 6, where our method faithfully recovers highly reflective surfaces such as the aluminum bowl and the metal plate, which even remain difficult for existing GS-based reconstruction methods.

Comparisons of Illumination Decomposition. To validate the effectiveness of our illumination decomposition, we visualize the decomposed material components and the learned environment maps on GlossySynthetic dataset [33], as shown in Fig. 7. Note that our method cannot be directly compared with inverse rendering methods [11, 31, 21], as the illumination modeling approaches are much different.

4.3 Ablation Study

Effectiveness of Each Module. We conduct ablation studies to evaluate the effectiveness of each module in our framework on both synthetic and real-world datasets. We start by analyzing the multi-view material inference strategies. Without any strategies, the result (Fig. 8 (a), "w/o $\mathcal{L}_{mv}$, w/o $\mathcal{L}_{ref}$" row in Tab. 2) show inconsistent material maps and weak reflections. Introducing $\mathcal{L}_{mv}$ makes the multi-view material maps uniform and consistent, leading to clearer reflections on the top of the sphere (Fig. 8 (b), "w/ $\mathcal{L}_{mv}$, w/o $\mathcal{L}_{ref}$" row in Tab. 2). Further adding $\mathcal{L}_{ref}$ improves the metallic in textureless reflective regions, making subtle reflections like ground seams more visible (Fig. 8 (c), "Full Model" row in Tab. 2). We also ablate the environment modeling strategy by removing ray tracing, relying on residual color and environment map to overfit incident illumination. This causes

Table 2: Ablation study on each one of our modules.

Datasets	ShinyBlender [47]			Ref-Real [47]		
Models	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
w/o $\mathcal{L}_{mv}$, w/o $\mathcal{L}_{ref}$	34.87	0.972	0.055	24.24	0.655	0.260
w/ $\mathcal{L}_{mv}$, w/o $\mathcal{L}_{ref}$	35.21	0.975	0.051	24.47	0.670	0.242
w/o $\mathcal{L}_n$	35.37	0.975	0.050	24.39	0.672	0.229
w/o Environment	34.69	0.976	0.049	24.76	0.681	0.199
Full Model	**35.57**	**0.976**	**0.049**	**25.04**	**0.703**	**0.185**

noticeably blurred reflections in inter-reflection regions (Fig. 9 (b), “w/o Environment” row in Tab. 2). Lastly, removing the normal prior from full model leas to degenerated geometry and color (Fig. 9 (a), “w/o $\mathcal{L}_n$” row in Tab. 2).

Table 3: Ablation study on normal prior.

Models	MAE↓	CD↓
w/o $\mathcal{L}_n$, w/o Reg, w/o Env	3.47	0.94
w/o $\mathcal{L}_n$	2.59	0.68
Full Model	2.04	0.60

Normal Prior. To evaluate the necessity of the normal prior, we conduct ablation studies on ShinyBlender dataset [47] under three experimental settings: Full Model, Full model without normal prior (w/o $\mathcal{L}_n$), Full model without normal prior, material regularization and environment modeling (w/o $\mathcal{L}_n$, w/o Reg, w/o Env). We report the normal accuracy in Tab. 3 using MAE, as well as the geometric reconstruction accuracy compared with ground truth meshes using Chamfer Distance (CD). The results indicate that, beyond the normal prior, our material regularization and environment modeling also contribute significantly to geometry learning. This is because both our material constraints and environment modeling are differentiable to the depth and normal, enabling end-to-end joint optimization of geometry, appearance, and material properties for improved overall performance.

5 Conclusion

We propose MaterialRefGS, a novel approach that learns multi-view illumination decomposition for reflective gaussian splatting through multi-view consistent material inference. To this end, we enforce the Gaussians to produce consistent material maps across different views, and explore reflection strength priors from photometric variants to provide explicit supervision for specular reflectance modeling. We also introduce a novel environment modeling strategy based on Gaussian ray tracing, which compensates for the indirect illumination caused by inter-object occlusion. Extensive ablation studies justify the effectiveness of our proposed modules, loss functions, and training strategies. Our evaluations show our superiority over the latest methods in rendering photorealistic novel views and recovering accurate geometry.

6 Acknowledgement

This work was supported by Deep Earth Probe and Mineral Resources Exploration – National Science and Technology Major Project (2024ZD1003405), and the National Natural Science Foundation of China (62272263), and in part by Kuaishou.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[2] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

[3] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Deep Reflectance Volumes: Relightable reconstructions from multi-view photometric images. In *European Conference on Computer Vision*, pages 294–311. Springer, 2020.

[4] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *ACM Siggraph*, volume 2012, pages 1–7. vol. 2012, 2012.

[5] Chao Chen, Yu-Shen Liu, and Zhizhong Han. NeuralTPS: Learning signed distance functions without priors from single sparse point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(1):565–582, 2025.

[6] 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.

[7] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2024.

[8] 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.

[9] Yikang Ding, Jiwen Liu, Wenyuan Zhang, Zekun Wang, Wentao Hu, Liyuan Cui, Mingming Lao, Yingchao Shao, Hui Liu, Xiaohan Li, et al. Kling-avatar: Grounding multimodal instructions for cascaded long-duration avatar animation synthesis. *arXiv preprint arXiv:2509.09595*, 2025.

[10] Kang Du, Zhihao Liang, and Zeyu Wang. GS-ID: Illumination Decomposition on Gaussian Splatting via Diffusion Prior and Parametric Light Source Optimization. *arXiv preprint arXiv:2408.08524*, 2024.

[11] 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.

[12] Daniel Gatis. Rembg: A tool to remove images background. https://github.com/danielgatis/rembg, 2023.

[13] Wenhang Ge, Tao Hu, Haoyu Zhao, Shu Liu, and Ying-Cong Chen. Ref-NeuS: Ambiguity-reduced neural implicit surface learning for multi-view reconstruction with reflection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4251–4260, 2023.

[14] 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.

[15] Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024.

[16] Liang Han, Xu Zhang, Haichuan Song, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. SparseRecon: Neural implicit surface reconstruction from sparse views with feature and depth consistencies. *Proceedings of International Conference on Computer Vision*, 2025.

[17] Liang Han, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Binocular-Guided 3D Gaussian Splatting with View Consistency for Sparse View Synthesis. In *Advances in Neural Information Processing Systems*, 2024.

[18] 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.

[19] Han Huang, Yulun Wu, Chao Deng, Ge Gao, Ming Gu, and Yu-Shen Liu. FatesGS: Fast and accurate sparse-view surface reconstruction using gaussian splatting with depth-feature consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[20] 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.

[21] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. TensoIR: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2023.

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[23] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024.

[24] Ye Keyang, Hou Qiming, and Zhou Kun. 3D Gaussian Splatting with Deferred Reflection. *ACM SIGGRAPH Conference Proceedings*, 2024.

[25] Jia Li, Lu Wang, Lei Zhang, and Beibei Wang. TensoSDF: Roughness-aware tensorial representation for robust geometry and material reconstruction. *ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024.

[26] Jiabao Li, Yuqi Li, Ciliang Sun, Chong Wang, and Jinhui Xiang. Spec-NeRF: Multi-spectral neural radiance fields. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2485–2489. IEEE, 2024.

[27] Qing Li, Huifang Feng, Xun Gong, and Yu-Shen Liu. VA-GS: Enhancing the geometric representation of gaussian splatting via view alignment. *Advances in Neural Information Processing Systems*, 2025.

[28] Shengtao Li, Yudong Liu, Ge Gao, Ming Gu, and Yu-Shen Liu. I-filtering: Implicit filtering for learning neural distance functions from 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[29] Shujuan Li, Yu-Shen Liu, and Zhizhong Han. GaussianUDF: Inferring Unsigned Distance Functions through 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[30] Ruofan Liang, Huiting Chen, Chunlin Li, Fan Chen, Selvakumar Panneer, and Nandita Vijaykumar. ENVIDR: Implicit differentiable renderer with neural environment lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 79–89, 2023.

[31] 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.

[32] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Point2Sequence: Learning the Shape Representation of 3D Point Clouds with an Attention-based Sequence to Sequence Network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8778–8785, 2019.

[33] 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.

[34] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024.

[35] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-Pull: Learning signed distance function from point clouds by learning to pull space onto surface. In *International Conference on Machine Learning*, pages 7246–7257. PMLR, 2021.

[36] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, pages 405–421. Springer, 2020.

[37] Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3D Gaussian Ray Tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics (TOG)*, 43(6):1–19, 2024.

[38] 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*, pages 8280–8290, 2022.

[39] Takeshi Noda, Chao Chen, Junsheng Zhou, Weiqi Zhang, Yu-Shen Liu, and Zhizhong Han. Learning bijective surface parameterization for inferring signed distance functions from sparse point clouds with grid deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[40] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. MIT Press, 2023.

[41] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[42] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[43] 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.

[44] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-GS: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024.

[45] Daniel R Steinwand, John A Hutchinson, and John P Snyder. Map projections for global and continental data sets and an analysis of pixel distortion caused by reprojection. *Photogrammetric Engineering & Remote Sensing*, 61(12):1487–1497, 1995.

[46] Zhe Jun Tang and Tat-Jen Cham. 3iGS: Factorised tensorial illumination for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 143–159. Springer, 2024.

[47] 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.

[48] Fangjinhua Wang, Marie-Julie Rakotosaona, Michael Niemeyer, Richard Szeliski, Marc Pollefeys, and Federico Tombari. UniSDF: Unifying neural representations for high-fidelity 3d reconstruction of complex scenes with reflections. *Advances in Neural Information Processing Systems*, 37:3157–3184, 2024.

[49] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[50] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024.

[51] Yulun Wu, Han Huang, Wenyuan Zhang, Chao Deng, Ge Gao, Ming Gu, and Yu-Shen Liu. Sparis: Neural implicit surface reconstruction of indoor scenes from sparse views. In *AAAI Conference on Artificial Intelligence*, 2025.

[52] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflake point deconvolution for point cloud completion and generation with skip-transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6320–6338, 2022.

[53] 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.

[54] Qingsen Yan, Yixu Feng, Cheng Zhang, Guansong Pang, Kangbiao Shi, Peng Wu, Wei Dong, Jinqiu Sun, and Yanning Zhang. HVI: A new color space for low-light image enhancement. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[55] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K Wong. PS-NeRF: Neural inverse rendering for multi-view photometric stereo. In *European Conference on Computer Vision*, pages 266–284. Springer, 2022.

[56] Yuxuan Yao, Zixuan Zeng, Chun Gu, Xiatian Zhu, and Li Zhang. Reflective gaussian splatting. *International Conference on Learning Representations*, 2025.

[57] Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. StableNormal: Reducing diffusion variance for stable and sharp normal. *ACM Transactions on Graphics (TOG)*, 43(6):1–18, 2024.

[58] Kai Ye, Chong Gao, Guanbin Li, Wenzheng Chen, and Baoquan Chen. GeoSplatting: Towards geometry guided gaussian splatting for physically-based inverse rendering. *arXiv preprint arXiv:2410.24204*, 2024.

[59] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[60] Weiqi Zhang, Junsheng Zhou, Haotian Geng, Wenyuan Zhang, and Yu-Shen Liu. GAP: Gaussianize any point clouds with text guidance. *Proceedings of International Conference on Computer Vision*, 2025.

[61] Wenyuan Zhang, Yu-Shen Liu, and Zhizhong Han. Neural Signed Distance Function Inference through Splatting 3D Gaussians Pulled on Zero-Level Set. In *Advances in Neural Information Processing Systems*, 2024.

[62] Wenyuan Zhang, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Learning unsigned distance functions from multi-view images with volume rendering priors. In *European Conference on Computer Vision*, pages 397–415. Springer, 2024.

[63] Wenyuan Zhang, Emily Yue ting Jia, Junsheng Zhou, Baorui Ma, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. NeRFPrior: Learning neural radiance field as a prior for indoor scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[64] Wenyuan Zhang, Ruofan Xing, Yunfan Zeng, Yu-Shen Liu, Kanle Shi, and Zhizhong Han. Fast Learning Radiance Fields by Shooting Much Fewer Rays. *IEEE Transactions on Image Processing*, 32:2703–2718, 2023.

[65] Wenyuan Zhang, Yixiao Yang, Han Huang, Liang Han, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. MonoInstance: Enhancing monocular priors via multi-view instance alignment for neural rendering and reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[66] Youjia Zhang, Anpei Chen, Yumin Wan, Zikai Song, Junqing Yu, Yawei Luo, and Wei Yang. Ref-GS: Directional factorization for 2d gaussian splatting. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[67] Junsheng Zhou, Baorui Ma, Shujuan Li, Yu-Shen Liu, Yi Fang, and Zhizhong Han. CAP-UDF: Learning unsigned distance functions progressively from raw point clouds with consistency-aware field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[68] Junsheng Zhou, Baorui Ma, Wenyuan Zhang, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Differentiable registration of images and lidar point clouds with voxelpoint-to-pixel matching. *Advances in Neural Information Processing Systems*, 36:51166–51177, 2023.

[69] Junsheng Zhou, Xingyu Shi, Haichuan Song, Yi Fang, Yu-Shen Liu, and Zhizhong Han. U-CAN: Unsupervised point cloud denoising with consistency-aware noise2noise matching. *Advances in Neural Information Processing Systems*, 2025.

[70] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3D: Exploring Unified 3D Representation at Scale. *International Conference on Learning Representations*, 2024.

[71] Junsheng Zhou, Weiqi Zhang, and Yu-Shen Liu. DiffGS: Functional gaussian splatting diffusion. In *Advances in Neural Information Processing Systems*, 2024.

[72] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378, 2001.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations and failure cases in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: We leverage the theory of physically based rendering that has been widely adopt by previous works. We do not make any assumptions in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the detailed information in reproducing our methods in Section 3 of the main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not include our code or data at this stage, but we will definitely release our code and data in the future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the training and testing details in the experiment section (Section 4).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report the average performance as the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report our training time with baseline methods in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the applications and potential impacts of our method in the introduction.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use the open-sourced datasets and codes under their licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Our method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.