

HMGS: Hybrid Model of Gaussian Splatting for Enhancing 3D Reconstruction with Reflections^{*}

Hengbin Zhang¹, Chengliang Wang^{1**}, Ji Liu¹, Tian Jiang², Yonggang Luo²,
and Lecheng Xie²

¹ College of Computer Science, Chongqing University, China
zhb@stu.cqu.edu.cn

{wangcl, liujiboy}@cqu.edu.cn

² AI Lab, Chongqing Changan Automobile Co., Ltd. Chongqing, 401120, China
{jiangtian, luoyg3, xielc}@changan.com.cn

Abstract. The advent of 3D Gaussian Splatting (3D-GS) marks a significant breakthrough in the field of 3D reconstruction, leveraging GPU rasterization technology to achieve real-time rendering with state-of-the-art quality. However, 3D-GS is limited by the capacity of low-order spherical harmonics to represent high-frequency reflective attributes, often resulting in the loss of critical information in scenes with highlights and reflections. To address this limitation, we propose HMGS, a hybrid model that enhances the original 3D-GS’s ability to capture reflective colors. Our approach employs a neural network to learn color components from both the camera viewing direction and the reflected light direction, which are then jointly trained with the original 3D-GS model. Furthermore, we introduce a smoothing loss for the viewing color component, effectively decoupling the two color components. Our method significantly improves the reconstruction performance of 3D-GS on datasets featuring metallic sheen, light reflections, and shadows, while also enhancing reconstruction quality on general datasets.

Keywords: 3D Gaussian Splatting · quality enhancement · lighting decomposition · reflection modeling

1 Introduction

High-quality 3D reconstruction from multiple views and support for real-time rendering are hot topics in current 3D computer vision. Neural Radiation Fields (NeRF) [22] combine novel view synthesis with neural networks to improve the quality of scene detail reconstruction with long training and inference time, making real-time rendering impractical. Despite numerous NeRF-based works aimed at reducing training costs [4, 9, 25] and improving reconstruction quality [1, 2], achieving both goals remains challenging.

^{*} This work was supported by Chongqing Natural Science Foundation Innovation and Development Joint Fund CSTB2023NSCQ-LZX0109

^{**} Corresponding author

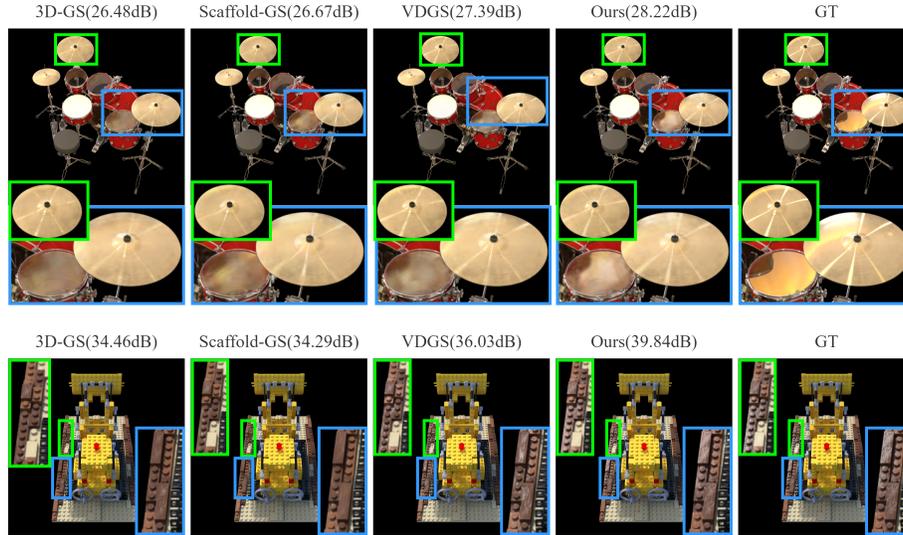


Fig. 1: Our method significantly enhances the modeling capability of 3D Gaussian Splatting (3D-GS) for objects with reflective properties. Specifically, the improvements are evident in the reconstruction quality of metallic sheen, reflected light, and shadows. The key to this enhancement lies in our hybrid model design, which captures more details through two distinct color fields and accurately models reflective components.

Recently, the newly proposed 3D Gaussian Splatting (3D-GS) [14] method has gained popularity as an alternative to NeRF, achieving both optimal training speed and rendering quality through explicit representation. Initially, it utilizes point cloud information obtained from Structure from Motion (SfM) to initialize Gaussian parameters and represent the scene with Gaussians. Retaining the concept of volume rendering, 3D-GS projects 3D Gaussians onto a 2D image plane and employs rasterization for rendering.

Nevertheless, 3D-GS struggles to model specular and reflective components in scenes, such as metallic sheen and light reflections (as shown in Fig. 1). VDGS [20] introduces a color density field based on viewing direction to capture the lost highlight information from 3D-GS, thereby slightly improving modeling performance. Scaffold-GS [18] adopts a neural anchor strategy, achieving a new state-of-the-art and increasing rendering speed.

These methods overlook the physical modeling of reflective views and fail to combine information from both viewing and reflection directions, leaving room for improvement in this task.

In this work, we first focus on improving the modeling performance of the reflection component. We decompose the scene into color components and construct a network to model the reflection component. This network supplements the color representation of reflective views in 3D-GS, achieving notable results. During the experiments, we observed that using only the reflection component

cannot represent diffuse and blurry reflections in the scene. Therefore, we add a network based on the camera view to enhance the correlation between Gaussian sphere features and observation directions, thereby improving the model’s ability to represent diffuse reflections and capture details. We combine the color components learned by these two networks using learnable weights and integrate them with the 3D-GS solution. This approach achieves high reconstruction quality in scenes with reflective properties. Our model also performs well when applied to general datasets and complex scenes, demonstrating its generalization capability.

In summary, our main contributions are as follows:

- We propose a hybrid model that decomposes the scene’s color representation. The model’s reflective component uses a simple physical model to represent specular colors, while the camera-direction component represents diffuse colors, both integrated with the original 3D-GS.
- We employ a smoothing loss function to control the viewing component’s color, enhancing the interpretability of the scene’s color decomposition by visualizing the rendering effects of the two components.
- Our model can better represents 3D objects with metallic sheen, light reflections, and shadows.

2 Related Work

2.1 Implicit Neural Radiance Fields

NeRF [22] has made remarkable breakthroughs in the field of 3D reconstruction. It combines multiple-view images and camera parameters as input to establish a multi-layer perceptron (MLP) model. This model represents the color and density of each point in the scene and generates realistic images through volume rendering, achieving astonishing results. Subsequent research has adapted NeRF to different scenarios from various perspectives, including unconstrained images [21], deformable scenes [37], generalization [47], and pose estimation [5]. These works have greatly expanded the applicability of NeRF and demonstrated its powerful modeling capabilities.

However, NeRF requires a large amount of per-pixel computation, significantly slowing down the training and rendering speed, which limits its application in large-scale scene reconstruction and real-time rendering. Many researchers have devoted their efforts to enhancing the efficiency of NeRF. These efforts include using spatial decomposition to employ smaller MLPs for each scene [28, 29], modifying the network structure [31], utilizing spatial voxels and octrees [17, 41], leveraging LiDAR priors [3], and designing specialized hardware architectures [23]. Although these methods have greatly improved the training or rendering speed while maintaining image quality, the essential need for querying ray casting has not changed, and there is still a gap in achieving real-time rendering.

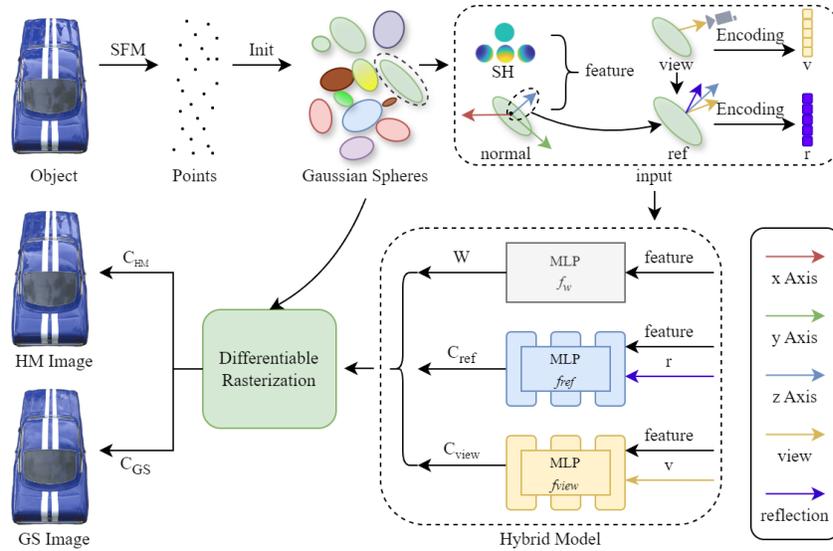


Fig. 2: Overview of the pipeline for HMGS. The object is converted into points using Structure from Motion (SfM) and initialized as Gaussian spheres. Each Gaussian sphere’s spherical harmonics features, normals (estimated using the shortest axis), encoded view and reflection directions are input into the hybrid model. The hybrid model uses Multi-Layer Perceptrons to learn view-dependent, reflection-dependent color fields and computes a learnable reflection weight W to synthesize the final color. Differentiable rasterization is used to render the image, with the output supervised by the original 3D-GS rendered image.

2.2 3D Gaussian Splatting

Unlike NeRF, which relies on volumetric ray marching for multi-layer perceptron optimization, 3D-GS [14] utilizes explicit spherical harmonic functions and rasterization techniques for rapid, high-quality, real-time rendering. It initiates Gaussian ellipsoids from sparse point clouds obtained through Structure from Motion (SfM) [32] and optimizes these ellipsoids to model scenes, with effective rendering achieved through ellipsoid rasterization into images. The point-based approach of 3DGS is well-suited for fast GPU/CUDA-based rasterization, which has garnered attention for its applications in autonomous driving [38,46], human reconstruction [16], and editing [30]. However, 3D-GS incurs significant storage overhead, especially when dealing with unbounded large-scale scenes where the SfM points can increase to millions.

Some methods [8,26] have proposed compressed 3D Gaussian representations to reduce redundancy and accelerate rendering. Additionally, 3D-GS has encountered issues such as over-densification, over-fitting with limited images, and reliance on accurate SfM initialization for reconstruction quality. Solutions

include introducing regularization techniques like depth [6] and progressive frequency regularization [43] to enhance reconstruction and rendering. Furthermore, optimizations to the 3D-GS method [13] have successfully relaxed the requirement for precise SfM-generated initial point clouds, enabling training from sub-optimal point clouds while achieving comparable or superior results.

2.3 Reflective Object Rendering

Rendering and reconstructing reflective objects presents a formidable challenge due to their complex interaction with light, which is intricately linked to both material properties and surface geometry. Recent progress, building upon pioneering techniques such as NeRF and 3DGS, has significantly advanced the field of reflective object rendering. Specifically, Ref-NeRF [34] introduces a novel approach by emphasizing the role of the incident light direction over the traditional observation direction, thereby enhancing the accuracy of specular reflection predictions. UniSDF [35] further innovates by integrating distinct networks for observation and incident light directions, offering a more nuanced model of reflective phenomena. GaussianShader [12], in a collaborative training regimen, refines the prediction of normal vectors, aligning them with the shortest principal axis direction and depth gradients to ensure the geometric fidelity of Gaussian spheres. Moreover, inverse rendering techniques [7, 33, 45] have emerged, capable of deciphering the underlying surface, material properties, and lighting conditions, thereby facilitating the high-fidelity reconstruction of reflective surfaces. Nonetheless, these sophisticated methods are not without trade-offs, often demanding more computational resources and storage capacity.

3 Method

3.1 Overview

Our method is an enhancement built upon the vanilla 3D Gaussian Splatting (3D-GS) algorithm. The overview of our proposed pipeline is illustrated in Fig.2. We separate the color into diffuse and specular components, using neural networks to learn these two components independently. A learnable weight coefficient and smoothing loss are employed to decouple the colors effectively. Section 3.2 will introduce the fundamentals of 3D-GS, followed by a detailed explanation of our model design in Section 3.3. The design of the loss functions will be discussed in Section 3.4.

3.2 Preliminaries

Compared to NeRF, 3D-GS has significantly impacted 3D reconstruction due to its real-time rendering capabilities and superior reconstruction quality. 3D-GS is a point-based method that uses third-order spherical harmonics to describe the anisotropic properties of Gaussians for scene representation. Each Gaussian is

characterized by several attributes: a central point position (mean), a covariance matrix Σ , scale, opacity, and spherical harmonics features for color representation.

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (1)$$

Where x represents any position in the scene and Σ is the covariance matrix. A 3D Gaussian can be projected onto a 2D plane using a 2D covariance matrix. This covariance matrix can be further decomposed into a rotation matrix R and a scaling matrix S .

$$\Sigma = R S S^T R^T \quad (2)$$

The projected 2D image is then divided into 16×16 regular blocks. For each 2D Gaussian within these blocks, the Gaussians are sorted by depth. The color of each pixel on the image plane is rendered using point-based volume rendering techniques, specifically alpha blending.

$$C(x') = \sum_{i \in N} c_i \sigma_i \prod_{j=1}^{i-1} (1 - \sigma_j), \quad \sigma_i = \alpha_i G'_i(x'), \quad (3) \quad (3)$$

Where x' represents the pixel coordinates, and N is the number of sorted 2D projected Gaussians. Utilizing differentiable rasterization, the attributes of the Gaussians can be directly optimized.

3.3 Hybrid Model

We propose a hybrid model that seamlessly integrates view-dependent and reflection-dependent color fields with the original 3D-GS pipeline to reconstruct scenes with potential reflective components accurately. The view-dependent color field captures complex information about weakly and moderately specular colors, while the reflection-dependent color field models strongly specular and mirror-like components. As shown in the Fig. 2, the final rendered result is computed by combining the colors from both components using a learnable reflection weight. During training, the original 3D-GS supervises the results, initializes parameters, and controls geometric information.

$$\mathbf{C}_{HM} = \mathbf{W} \cdot \mathbf{C}_{diffuse} + (1 - \mathbf{W}) \cdot \mathbf{C}_{specular} \quad (4)$$

View-Dependent Color Field. By introducing the view direction, VDGS [20] learns a density or color increment to supplement the reconstruction details of the original 3D-GS. Similarly, we introduce the view direction field to learn the diffuse colors in the scene. Unlike VDGS, we prune the scale and other geometric parameters of the Gaussians, using spherical harmonics features to encode color and incorporating the normal vectors of each Gaussian. Specifically:

$$\mathbf{C}_{diffuse} = f_{view}(\mathbf{sh}, \mathbf{n}, \omega_o) \quad (5)$$



Fig. 3: Visualization of color decomposition by our method on "garden spheres", "toy-car", "sedan" scenes in Ref Real dataset.

where \mathbf{sh} are the 48-dimensional spherical harmonics color features, \mathbf{n} is the surface normal of each Gaussian, and ω_o is the camera viewing direction.

Normal Estimation. Directly estimating Gaussian normals is challenging since 3D-GS is a point-based method, and each Gaussian represents a small, discrete part of local space, making it difficult to form a continuous surface. Existing methods, such as SuGaR [10], 2DGS [11], and GauStudio [40], regularize surface distributions, while others, like 3DGSR [19] and GSDF [42], combine with signed distance fields (SDF). However, experiments show that predicting precise normals introduces extra computational cost with limited improvement in color reconstruction performance.

We directly use the shortest axis of the Gaussian as the normal direction. This approach leverages the observation that Gaussians tend to flatten during optimization, providing a reasonable approximation of the normal without additional parameters or costs.

Reflection-Dependent Color Field. The original 3D-GS ignores the modeling of the physical properties of object reflections in the scene. We propose a reflection direction color field to capture this information more accurately. Based

on the simple physical illumination model, we can calculate the reflection direction ω_r from the camera viewing direction and normals:

$$\omega_r = 2(\omega_o \cdot \mathbf{n})\mathbf{n} - \omega_o \quad (6)$$

where ω_o is obtained by subtracting the camera position from the Gaussian center. The reflection direction is then input into the reflection view color field:

$$\mathbf{C}_{specular} = f_{ref}(\mathbf{sh}, \mathbf{n}, \omega_r) \quad (7)$$

The main difference between the two color fields is the input view direction. As shown in Eq. 4, we also use a weight network to combine the two color components into the final color output:

$$\mathbf{W} = \text{sigmoid}(f_w(\mathbf{sh}, \mathbf{n})) \quad (8)$$

Fig. 3 shows the hybrid model’s final output colors along with the individual color components, demonstrating effective color decomposition and accurate reconstruction of scenes with reflective properties.

3.4 Optimization

Building on the original \mathcal{L}_1 and \mathcal{L}_{D-SSIM} joint loss \mathcal{L}_{rgb} used for optimizing Gaussian attributes, we incorporate the photometric loss for the hybrid model following the same loss function structure.

$$\mathcal{L}_{HMrgb} = \lambda \mathcal{L}_1(C, \hat{C}) + (1 - \lambda) \mathcal{L}_{D-SSIM}(C, \hat{C}) \quad (9)$$

where λ is a balancing coefficient, C is the ground truth color, and \hat{C} is the predicted rendered color by the hybrid model. The \mathcal{L}_1 term calculates the absolute error between the ground truth and prediction, while \mathcal{L}_{D-SSIM} refers to the differentiable structural similarity index measure.

To improve the interpretability of color decomposition, we observed that photometric loss alone yielded poor results. Therefore, we introduce a smoothing loss for the viewing-dependent color to penalize abrupt changes between neighboring pixels, encouraging the generated image to be smoother and more natural, effectively decoupling the color components:

$$\mathcal{L}_{smooth} = \sum_{i,j} (|\mathbf{I}_{i,j} - \mathbf{I}_{i+1,j}| + |\mathbf{I}_{i,j} - \mathbf{I}_{i,j+1}|) \quad (10)$$

where $\mathbf{I}_{i,j}$ represents the pixel value at position (i, j) in the image \mathbf{I} . Overall, the final loss function is:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{rgb} + \lambda_2 \mathcal{L}_{HMrgb} + \lambda_{smooth} \mathcal{L}_{smooth} \quad (11)$$

The value of λ_{smooth} directly impacts the model’s decomposition performance. Fig. 4 shows the effect of different λ_{smooth} values on the rendering of reflection-dependent colors.

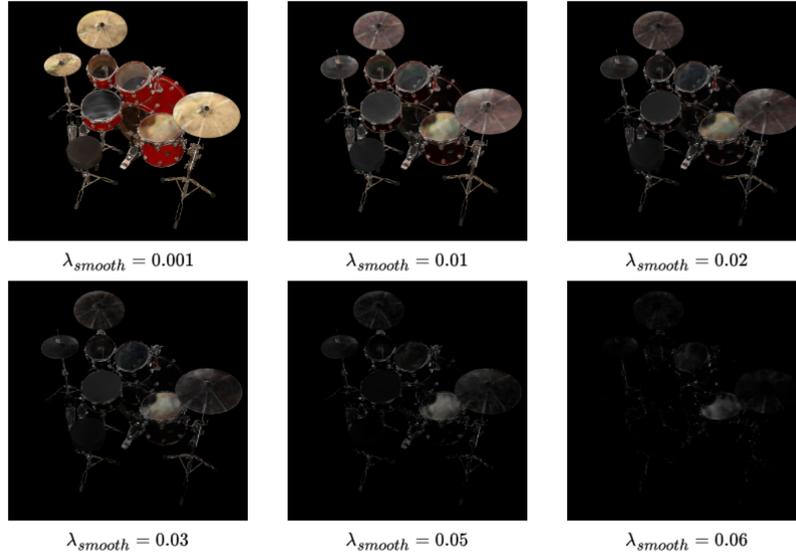


Fig. 4: Visualization of the impact of different λ_{smooth} values on color decomposition. The figure shows $C_{specular}$ and the performance is best when $\lambda_{smooth} = 0.02$.

4 Experiments

4.1 Experimental Settings

Datasets and Evaluation. We comprehensively evaluate our method across three publicly available datasets, encompassing a total of 17 scenes. The NeRF Synthetic dataset [22] includes objects with complex geometries and realistic non-Lambertian materials, demonstrating our method’s ability to reconstruct objects accurately under weak reflection conditions. The Shiny Blender dataset [34] comprises six different objects rich in reflective projection information, showcasing our method’s superior reconstruction capabilities for reflective objects. To further validate the effectiveness of our approach in modeling reflections in real-world scenarios, we utilize the Ref Real dataset [34]. Additionally, to evaluate our method’s performance in general scenes, we selected the garden scene from the MipNeRF360 dataset [2] and the truck scene from the Tanks and Temples dataset [15]. In addition to qualitative visualizations, we report standard metrics such as PSNR, SSIM [36], and LPIPS [44] for quantitative analysis. Our method is compared against the primary baseline 3D-GS and other leading methods. This comprehensive evaluation demonstrates the robustness and effectiveness of our hybrid model in various reconstruction scenarios.

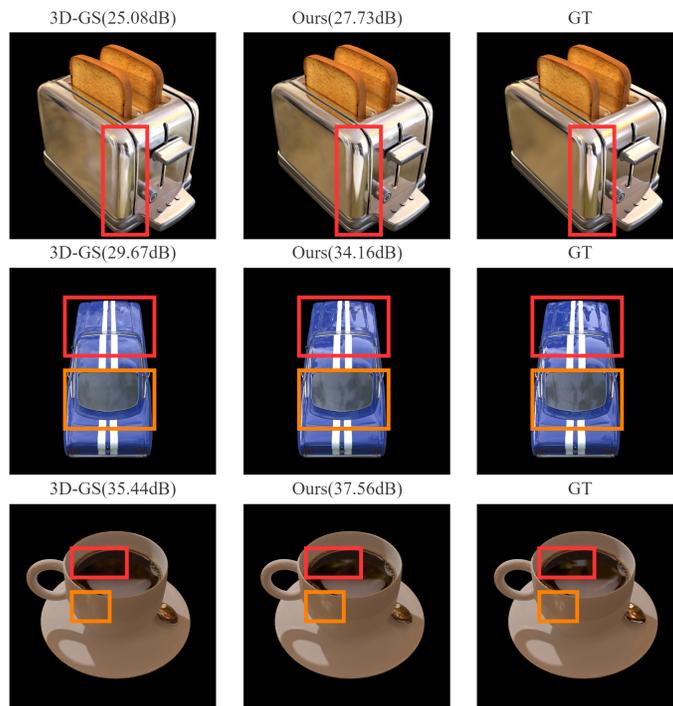


Fig. 5: Comparative visualization of reconstruction results for various scenes from the Shiny Blender dataset using our method.

Implementation Details. We implement our proposed method using PyTorch [27], with all training and evaluation conducted on an Nvidia RTX 4090. The training consists of 30,000 epochs. During the initial 3,000 epochs, when the spherical harmonics orders are still varying, we use only the original 3D-GS loss \mathcal{L}_{rgb} for initialization and set λ_1 to 1. After 3,000 epochs, we enable all losses and set λ_{smooth} to 0.02. For synthetic datasets, we set $\lambda_1 = \lambda_2 = 1$, and for real-world datasets, we set $\lambda_1 = \lambda_2 = 0.5$, effectively controlling the number of generated Gaussians. To reduce training costs and ensure fairness, the training on real scene datasets uses the provided images at the lowest resolution. Our method utilizes two three-layer MLPs for learning color and a single-layer MLP for the weight. The input viewing and reflection directions are encoded using spherical harmonics provided by tinycudann [24]. The training and testing dataset split follows the same strategy as 3D-GS. We select every eighth image for the test set if a dataset does not provide a predefined split.

4.2 Results Analysis

Quantitatively, our method achieves excellent overall performance among the compared models. Tab. 1 shows that our method gets the highest scores in terms

Table 1: View Synthesis Comparison Results on NeRF Synthetic Dataset.

| Method Scenes | PSNR \uparrow | | | | | | | | |
|--------------------|--------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | chair | drums | figus | hotdog | lego | materials | mic | ship | avg. |
| NeRF [22] | 33.00 | 25.01 | 30.13 | 36.18 | 32.54 | 29.62 | 32.91 | 28.65 | 31.00 |
| Mip-NeRF [1] | 35.12 | 25.36 | 33.19 | 37.34 | 35.92 | 30.64 | 36.76 | 30.52 | 33.11 |
| Ref-NeRF [34] | 33.98 | 25.43 | 28.74 | 37.04 | 35.10 | 27.10 | 33.65 | 29.24 | 31.29 |
| 3D-GS [14] | 35.57 | 26.28 | 35.50 | 38.07 | 36.00 | 30.48 | 36.79 | 31.68 | 33.80 |
| VDGS [20] | 35.52 | 26.54 | 35.60 | 38.03 | 36.20 | 30.56 | 36.88 | 31.47 | 33.85 |
| Scaffold-GS [18] | 34.82 | 26.13 | 35.10 | 37.77 | 35.46 | 30.55 | 36.65 | 31.08 | 33.45 |
| Spec-Gaussian [39] | 35.48 | 26.82 | 35.91 | 38.08 | 36.14 | 30.91 | 36.96 | 31.81 | 34.01 |
| GS-Shader [12] | 35.83 | 26.36 | 34.97 | 37.85 | 35.87 | 30.07 | 35.23 | 30.82 | 33.38 |
| Ours | 35.82 | 26.85 | 36.20 | 38.38 | 36.82 | 30.40 | 37.13 | 31.98 | 34.20 |
| Method Scenes | SSIM \uparrow | | | | | | | | |
| | chair | drums | figus | hotdog | lego | materials | mic | ship | avg. |
| NeRF [22] | 0.967 | 0.925 | 0.964 | 0.974 | 0.961 | 0.949 | 0.980 | 0.856 | 0.947 |
| Mip-NeRF [1] | 0.981 | 0.933 | 0.980 | 0.982 | 0.980 | 0.959 | 0.992 | 0.885 | 0.962 |
| Ref-NeRF [34] | 0.974 | 0.929 | 0.954 | 0.979 | 0.975 | 0.921 | 0.983 | 0.864 | 0.947 |
| 3D-GS [14] | 0.988 | 0.955 | 0.987 | 0.985 | 0.983 | 0.960 | 0.993 | 0.906 | 0.970 |
| VDGS [20] | 0.987 | 0.954 | 0.988 | 0.985 | 0.983 | 0.961 | 0.993 | 0.907 | 0.970 |
| Scaffold-GS [18] | 0.985 | 0.947 | 0.985 | 0.984 | 0.980 | 0.959 | 0.992 | 0.900 | 0.967 |
| Spec-Gaussian [39] | 0.987 | 0.955 | 0.988 | 0.985 | 0.982 | 0.963 | 0.993 | 0.905 | 0.970 |
| GS-Shader [12] | 0.987 | 0.949 | 0.985 | 0.985 | 0.983 | 0.960 | 0.991 | 0.905 | 0.968 |
| Ours | 0.988 | 0.956 | 0.988 | 0.987 | 0.985 | 0.960 | 0.993 | 0.909 | 0.971 |
| Method Scenes | LPIPS \downarrow | | | | | | | | |
| | chair | drums | figus | hotdog | lego | materials | mic | ship | avg. |
| NeRF [22] | 0.046 | 0.091 | 0.044 | 0.121 | 0.050 | 0.063 | 0.028 | 0.206 | 0.081 |
| Mip-NeRF [1] | 0.020 | 0.064 | 0.021 | 0.026 | 0.018 | 0.040 | 0.008 | 0.135 | 0.042 |
| Ref-NeRF [34] | 0.029 | 0.073 | 0.056 | 0.028 | 0.025 | 0.078 | 0.018 | 0.158 | 0.058 |
| 3D-GS [14] | 0.010 | 0.037 | 0.012 | 0.020 | 0.016 | 0.037 | 0.006 | 0.106 | 0.031 |
| VDGS [20] | 0.011 | 0.037 | 0.011 | 0.020 | 0.015 | 0.037 | 0.006 | 0.104 | 0.030 |
| Scaffold-GS [18] | 0.014 | 0.049 | 0.014 | 0.023 | 0.019 | 0.042 | 0.008 | 0.110 | 0.035 |
| Spec-Gaussian [39] | 0.011 | 0.034 | 0.011 | 0.018 | 0.016 | 0.033 | 0.006 | 0.098 | 0.028 |
| GS-Shader [12] | 0.012 | 0.040 | 0.013 | 0.019 | 0.014 | 0.033 | 0.006 | 0.098 | 0.029 |
| Ours | 0.010 | 0.036 | 0.011 | 0.016 | 0.013 | 0.037 | 0.006 | 0.097 | 0.028 |

of PSNR, SSIM, and LPIPS on the NeRF Synthetic dataset. Tab. 2 demonstrates that our model still performs commendably on the Shiny Blender dataset, particularly in terms of structural and perceptual metrics.

Fig. 1 and Fig. 5 illustrate the visualization of rendering results from our method across five different scenes in the synthetic dataset. The improvements in reconstruction quality are particularly noticeable in the metallic highlights, as seen on the cymbals in the drums scene and the surface of the toaster. Reflective

Table 2: View Synthesis Comparison Results on Shiny Blender Dataset.

| Method Scenes | PSNR \uparrow | | | | | | |
|---------------------------|--------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | ball | car | coffee | helmet | teapot | toaster | avg. |
| Mip-NeRF [1] | 25.94 | 26.50 | 30.36 | 27.39 | 46.00 | 22.37 | 29.76 |
| Ref-NeRF [34] | 29.14 | 30.41 | 33.99 | 29.92 | 45.19 | 25.29 | 32.32 |
| 3D-GS [14] | 27.68 | 27.32 | 32.70 | 27.68 | 45.52 | 21.14 | 30.34 |
| VDGS [20] | 27.25 | 27.50 | 32.74 | 28.20 | 45.37 | 22.03 | 30.52 |
| Scaffold-GS [18] | 28.36 | 27.48 | 32.91 | 28.47 | 44.97 | 22.29 | 30.74 |
| Spec-Gaussian [39] | 28.60 | 27.46 | 32.90 | 28.40 | 45.77 | 22.65 | 30.96 |
| GS-Shader [12] | 30.98 | 27.90 | 32.39 | 28.32 | 45.86 | 26.21 | 31.94 |
| Ours | 27.83 | 27.62 | 32.94 | 28.01 | 45.99 | 22.52 | 30.82 |
| Method Scenes | SSIM \uparrow | | | | | | |
| | ball | car | coffee | helmet | teapot | toaster | avg. |
| Mip-NeRF [1] | 0.935 | 0.922 | 0.966 | 0.939 | 0.997 | 0.891 | 0.942 |
| Ref-NeRF [34] | 0.956 | 0.949 | 0.972 | 0.955 | 0.995 | 0.910 | 0.956 |
| 3D-GS [14] | 0.937 | 0.929 | 0.971 | 0.950 | 0.997 | 0.896 | 0.947 |
| VDGS [20] | 0.936 | 0.929 | 0.971 | 0.951 | 0.997 | 0.901 | 0.948 |
| Scaffold-GS [18] | 0.944 | 0.927 | 0.971 | 0.951 | 0.997 | 0.899 | 0.948 |
| Spec-Gaussian [39] | 0.943 | 0.928 | 0.970 | 0.951 | 0.997 | 0.908 | 0.950 |
| GS-Shader [12] | 0.965 | 0.931 | 0.971 | 0.950 | 0.996 | 0.929 | 0.957 |
| Ours | 0.936 | 0.929 | 0.971 | 0.951 | 0.997 | 0.903 | 0.948 |
| Method Scenes | LPIPS \downarrow | | | | | | |
| | ball | car | coffee | helmet | teapot | toaster | avg. |
| Mip-NeRF [1] | 0.168 | 0.059 | 0.086 | 0.108 | 0.008 | 0.123 | 0.092 |
| Ref-NeRF [34] | 0.307 | 0.051 | 0.082 | 0.087 | 0.013 | 0.118 | 0.109 |
| 3D-GS [14] | 0.162 | 0.048 | 0.080 | 0.083 | 0.007 | 0.125 | 0.084 |
| VDGS [20] | 0.164 | 0.047 | 0.079 | 0.076 | 0.008 | 0.115 | 0.082 |
| Scaffold-GS [18] | 0.146 | 0.051 | 0.081 | 0.073 | 0.007 | 0.107 | 0.078 |
| Spec-Gaussian [39] | 0.144 | 0.047 | 0.080 | 0.073 | 0.007 | 0.099 | 0.075 |
| GS-Shader [12] | 0.121 | 0.045 | 0.078 | 0.076 | 0.007 | 0.079 | 0.068 |
| Ours | 0.166 | 0.047 | 0.078 | 0.080 | 0.007 | 0.119 | 0.083 |

projections are evident on the car hood, car windows, and the surface of the liquid in the coffee cup. Additionally, the detailed diffuse lighting is well captured, as demonstrated in the Lego scene.

When reconstructing real-world datasets, we show in Tab. 3 and Fig. 3 that our method outperforms the primary baseline, original 3D-GS, successfully modeling and decomposing reflective highlights in real-world scenes.

Table 3: View Synthesis Comparison Results on Real Scene Dataset.

| Method Scenes | PSNR \uparrow | | | | | |
|-------------------|-----------------|--------------|---------------|--------------|--------------|--------------|
| | toycar | sedan | gardenspheres | truck | garden | avg. |
| 3D-GS [14] | 26.10 | 26.00 | 25.22 | 25.33 | 29.02 | 26.33 |
| Ours | 26.39 | 26.15 | 25.32 | 25.41 | 29.31 | 26.51 |

| Method Scenes | SSIM \uparrow | | | | | |
|-------------------|-----------------|--------------|---------------|--------------|--------------|--------------|
| | toycar | sedan | gardenspheres | truck | garden | avg. |
| 3D-GS [14] | 0.767 | 0.768 | 0.757 | 0.878 | 0.914 | 0.817 |
| Ours | 0.772 | 0.771 | 0.765 | 0.884 | 0.915 | 0.821 |

| Method Scenes | LPIPS \downarrow | | | | | |
|-------------------|--------------------|--------------|---------------|--------------|--------------|--------------|
| | toycar | sedan | gardenspheres | truck | garden | avg. |
| 3D-GS [14] | 0.133 | 0.206 | 0.128 | 0.148 | 0.061 | 0.135 |
| Ours | 0.140 | 0.206 | 0.130 | 0.132 | 0.060 | 0.134 |

4.3 Ablation Study

All ablation experiments are conducted on the NeRF Synthetic and Shiny Blender datasets. To evaluate the effectiveness of our training strategy, we perform ablation studies in Tab. 4 by individually removing the reflection-dependent color field module, the view-dependent color field module, and the smoothing loss function. Additionally, we compare the visual impact of different smoothing loss function coefficients in Fig. 4. Removing modules or loss functions from our network generally results in declining reconstruction quality.

Table 4: Ablation Studies on Model Design.

| Model | | | | NeRF Synthetic | | | Shiny Blender | | |
|-------------|--------------|--------------|--------------|-----------------|-----------------|--------------------|-----------------|-----------------|--------------------|
| | ColorV | ColorR | SmoothLoss | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow |
| A | | | | 33.80 | 0.970 | 0.031 | 30.34 | 0.947 | 0.084 |
| B | \checkmark | | | 33.85 | 0.970 | 0.030 | 30.52 | 0.947 | 0.083 |
| C | | \checkmark | | 33.91 | 0.970 | 0.030 | 30.59 | 0.947 | 0.084 |
| D | \checkmark | \checkmark | | 34.00 | 0.970 | 0.031 | 30.68 | 0.947 | 0.084 |
| Ours | \checkmark | \checkmark | \checkmark | 34.20 | 0.971 | 0.028 | 30.82 | 0.948 | 0.083 |

4.4 Limitation

Despite the significant advantages of HMGS over traditional methods in object reconstruction, it has certain limitations. Its performance on complex, large-scale, real-world datasets still has room for improvement. Future research may

focus on enhancing model performance in real-world, large-scale scenes and precisely predicting normals to achieve more accurate reflection directions without substantially increasing training overhead.

5 Conclusion

In this work, we propose a novel method to enhance the modeling quality of 3D Gaussian Splatting (3D-GS) in scenes with reflective properties while also achieving excellent reconstruction results on general datasets. By introducing the learning of color components from both the viewing and reflection directions, our method effectively decomposes and captures reflection information within scenes. It significantly enhances the reconstruction performance for reflective light and shadows. We conduct extensive experiments on publicly available synthetic and real-world datasets, demonstrating the effectiveness and superiority of our method’s design.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)
3. Cao, J., Li, Z., Wang, N., Ma, C.: Lightning nerf: Efficient hybrid scene representation for autonomous driving. arXiv preprint arXiv:2403.05907 (2024)
4. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision. pp. 333–350. Springer (2022)
5. Chen, S., Bhargat, Y., Li, X., Bian, J.W., Li, K., Wang, Z., Prisacariu, V.A.: Neural refinement for absolute pose regression with feature synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20987–20996 (2024)
6. Chung, J., Oh, J., Lee, K.M.: Depth-regularized optimization for 3d gaussian splatting in few-shot images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 811–820 (2024)
7. Fan, Y., Skorokhodov, I., Voynov, O., Ignatyev, S., Burnaev, E., Wonka, P., Wang, Y.: Factored-neus: Reconstructing surfaces, illumination, and materials of possibly glossy objects. ArXiv [abs/2305.17929](https://arxiv.org/abs/2305.17929) (2023), <https://api.semanticscholar.org/CorpusID:258960329>
8. Fan, Z., Wang, K., Wen, K., Zhu, Z., Xu, D., Wang, Z.: Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. arXiv preprint arXiv:2311.17245 (2023)
9. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5501–5510 (2022)

10. Guédon, A., Lepetit, V.: 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. pp. 5354–5363 (2024)
11. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. arXiv preprint arXiv:2403.17888 (2024)
12. Jiang, Y., Tu, J., Liu, Y., Gao, X., Long, X., Wang, W., Ma, Y.: Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5322–5332 (2024)
13. Jung, J., Han, J., An, H., Kang, J., Park, S., Kim, S.: Relaxing accurate initialization constraint for 3d gaussian splatting. arXiv preprint arXiv:2403.09413 (2024)
14. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* **42**(4), 1–14 (2023)
15. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)* **36**(4), 1–13 (2017)
16. Li, M., Yao, S., Xie, Z., Chen, K., Jiang, Y.G.: Gaussianbody: Clothed human reconstruction via 3d gaussian splatting. arXiv preprint arXiv:2401.09720 (2024)
17. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. *Advances in Neural Information Processing Systems* **33**, 15651–15663 (2020)
18. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20654–20664 (2024)
19. Lyu, X., Sun, Y.T., Huang, Y.H., Wu, X., Yang, Z., Chen, Y., Pang, J., Qi, X.: 3dgsr: Implicit surface reconstruction with 3d gaussian splatting. arXiv preprint arXiv:2404.00409 (2024)
20. Malarz, D., Smolak, W., Tabor, J., Tadeja, S., Spurek, P.: Gaussian splatting with nerf-based color and opacity
21. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7210–7219 (2021)
22. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021)
23. Mubarik, M.H., Kanungo, R., Zirr, T., Kumar, R.: Hardware acceleration of neural graphics. In: Proceedings of the 50th Annual International Symposium on Computer Architecture. pp. 1–12 (2023)
24. Müller, T.: Tiny cuda neural network framework. Retrieved May **18**, 2023 (2021)
25. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)* **41**(4), 1–15 (2022)
26. Niedermayr, S., Stumpfegger, J., Westermann, R.: Compressed 3d gaussian splatting for accelerated novel view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10349–10358 (2024)
27. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019)

28. Rebain, D., Jiang, W., Yazdani, S., Li, K., Yi, K.M., Tagliasacchi, A.: Derf: Decomposed radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14153–14161 (2021)
29. Reiser, C., Peng, S., Liao, Y., Geiger, A.: Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 14335–14345 (2021)
30. Silva, M.C., Dahaghin, M., Toso, M., Del Bue, A.: Contrastive gaussian clustering: Weakly supervised 3d scene segmentation. arXiv preprint arXiv:2404.12784 (2024)
31. Sitzmann, V., Rezkikov, S., Freeman, B., Tenenbaum, J., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems* **34**, 19313–19325 (2021)
32. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: ACM siggraph 2006 papers, pp. 835–846 (2006)
33. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7495–7504 (2021)
34. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5481–5490. IEEE (2022)
35. Wang, F., Rakotosaona, M.J., Niemeyer, M., Szeliski, R., Pollefeys, M., Tombari, F.: Unisdf: Unifying neural representations for high-fidelity 3d reconstruction of complex scenes with reflections. arXiv preprint arXiv:2312.13285 (2023)
36. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* **13**(4), 600–612 (2004)
37. Weng, C.Y., Curless, B., Srinivasan, P.P., Barron, J.T., Kemelmacher-Shlizerman, I.: Humannerf: Free-viewpoint rendering of moving people from monocular video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16210–16220 (2022)
38. Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street gaussians for modeling dynamic urban scenes. arXiv preprint arXiv:2401.01339 (2024)
39. Yang, Z., Gao, X., Sun, Y., Huang, Y., Lyu, X., Zhou, W., Jiao, S., Qi, X., Jin, X.: Spec-gaussian: Anisotropic view-dependent appearance for 3d gaussian splatting. arXiv preprint arXiv:2402.15870 (2024)
40. Ye, C., Nie, Y., Chang, J., Chen, Y., Zhi, Y., Han, X.: Gaustudio: A modular framework for 3d gaussian splatting and beyond. arXiv preprint arXiv:2403.19632 (2024)
41. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenotrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5752–5761 (2021)
42. Yu, M., Lu, T., Xu, L., Jiang, L., Xiangli, Y., Dai, B.: Gsdf: 3dgs meets sdf for improved rendering and reconstruction. arXiv preprint arXiv:2403.16964 (2024)
43. Zhang, J., Zhan, F., Xu, M., Lu, S., Xing, E.: Fregs: 3d gaussian splatting with progressive frequency regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21424–21433 (2024)
44. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)

45. Zhang, X., Srinivasan, P.P., Deng, B., Debevec, P., Freeman, W.T., Barron, J.T.: Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)* **40**(6), 1–18 (2021)
46. Zhou, X., Lin, Z., Shan, X., Wang, Y., Sun, D., Yang, M.H.: Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 21634–21643 (2024)
47. Zhu, H.: X-nerf: Explicit neural radiance field for multi-scene 360deg insufficient rgb-d views. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. pp. 5766–5775 (2023)