

Spec-Gaussian: Anisotropic View-Dependent Appearance for 3D Gaussian Splatting

Ziyi Yang^{1,3} Xinyu Gao¹ Yangtian Sun² Yihua Huang² Xiaoyang Lyu²
Wen Zhou³ Shaohui Jiao³ Xiaojuan Qi² Xiaogang Jin¹

¹Zhejiang University ²The University of Hong Kong ³ByteDance Inc.

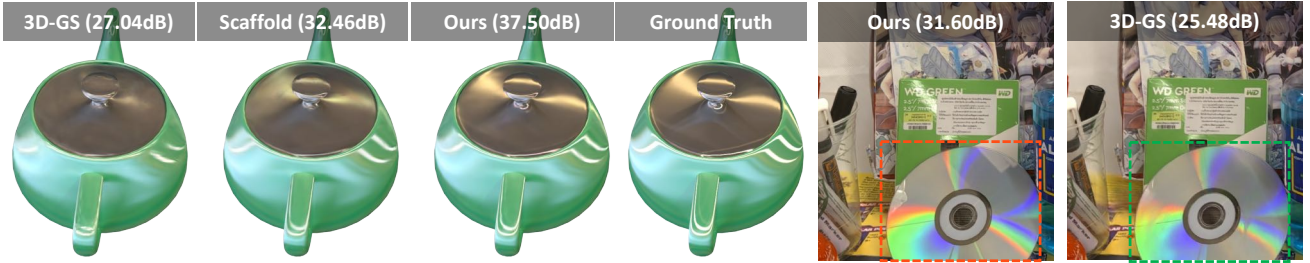


Figure 1. Our method not only achieves real-time rendering but also significantly enhances the capability of 3D-GS to model scenes with specular and anisotropic components. Key to this enhanced performance is our use of ASG appearance field to model the appearance of each 3D Gaussian, which results in substantial improvements in rendering quality for both complex and general scenes. Moreover, we employ anchor Gaussians to constrain the geometry of point-based representations, thereby improving the ability of 3D-GS to accurately model reflective parts and accelerating both training and rendering processes.

Abstract

The recent advancements in 3D Gaussian splatting (3D-GS) have not only facilitated real-time rendering through modern GPU rasterization pipelines but have also attained state-of-the-art rendering quality. Nevertheless, despite its exceptional rendering quality and performance on standard datasets, 3D-GS frequently encounters difficulties in accurately modeling specular and anisotropic components. This issue stems from the limited ability of spherical harmonics (SH) to represent high-frequency information. To overcome this challenge, we introduce Spec-Gaussian, an approach that utilizes an anisotropic spherical Gaussian (ASG) appearance field instead of SH for modeling the view-dependent appearance of each 3D Gaussian. Additionally, we have developed a coarse-to-fine training strategy to improve learning efficiency and eliminate floaters caused by overfitting in real-world scenes. Our experimental results demonstrate that our method surpasses existing approaches in terms of rendering quality. Thanks to ASG, we have significantly improved the ability of 3D-GS to model scenes with specular and anisotropic components without increasing the number of 3D Gaussians. This improvement extends the applicability of 3D GS to handle

intricate scenarios with specular and anisotropic surfaces. Our codes and datasets will be released.

1. Introduction

从一组图像中进行高质量重建和逼真渲染对于多种应用至关重要，例如增强现实/虚拟现实（AR/VR）、3D内容制作和艺术创作。经典方法采用原始表示形式，如网格[34]和点[4, 60]，并利用为当代GPU优化的光栅化管线实现实时渲染。

相比之下，神经辐射场（NeRF）[6, 32, 33]利用神经隐式表示提供连续的场景表示，并采用体渲染来生成渲染结果。这种方法可以更好地保留场景细节，并更有效地重建场景几何。

近年来，3D高斯散射（3D-GS）[21]作为一种领先技术出现，提供了最先进的质量和实时速度。该方法优化了一组3D高斯函数，能够同时捕捉3D场景的外观和几何，提供连续表示，保留细节并生成高质量的结果。此外，针对3D高斯函数的CUDA定制可微光栅化管线使得即使在高分辨率下也能实现实时渲染。

尽管表现出色，3D-GS在建模场景中的镜面成分方面仍存在困难（见图1）。这个问题主要源于低阶球谐函数（SH）捕捉这些场景所需的高频信息的能力

有限。因此，这对3D-GS在建模镜面成分时构成了挑战。1 scenes with reflections and specular components, as illustrated in Fig. 1 and Fig. 7.

为了解决这个问题，我们提出了一种新方法，称为*Spec-Gaussian*，该方法结合了各向异性球面高斯 (ASG) [54]用于建模各向异性和镜面反射成分、基于锚点的几何感知3D高斯以加速和减少存储需求，以及一种有效的训练机制以消除浮动物体并提高学习效率。具体来说，该方法包括三个关键设计：1) 一种新的3D高斯表示，利用ASG外观场而不是SH来建模每个3D高斯的外观。低阶SH无法有效建模高频信息，而少数阶的ASG则可以。这一新设计使得3D-GS能够更有效地在静态场景中建模各向异性和镜面反射成分。2) 一种混合方法，采用稀疏锚点来控制其子高斯的位置和表示。这种策略使我们能够只存储锚点高斯，显著减少需要存储的3D高斯总数，并增强几何结构。3) 一种专门为3D-GS设计的粗到细训练方案，旨在消除浮动物体并提高学习效率。这一策略通过在初始阶段优化低分辨率渲染，有效缩短了学习时间，避免了需要增加3D高斯数量，并通过规范学习过程来防止生成导致浮动物体的不必要几何结构。

通过结合这些进展，我们的方法可以渲染出高质量的镜面高光和各向异性效果，如图4所示，同时保持高斯的效率。此外，综合实验表明，我们的方法不仅赋予3D-GS建模镜面高光的能力，而且在通用基准测试中也达到了最先进的结果。

总而言之，我们工作的主要贡献如下：

- 提出了一种新颖的ASG外观场，用于建模每个3D高斯的视角依赖外观，使得3D-GS能够在不牺牲渲染速度的情况下有效表示具有镜面反射和各向异性成分的场景。
- 提出了一种基于锚点的混合模型，以减少学习ASG外观场带来的计算和存储开销。
- 提出了一种粗到细的训练方案，有效规范训练过程以消除浮动物体并提高3D-GS在现实场景中的学习效率。
- 创建了一个各向异性数据集，用于评估我们模型在表示各向异性方面的能力。大量实验表明，我们的方法在建模具有镜面高光和各向异性场景方面的有效性。

2. Related Work

2.1. Implicit Neural Radiance Fields

Neural rendering has attracted significant interest in the academic community for its unparalleled ability to generate photorealistic images. Methods like NeRF [32] utilize Multi-Layer Perceptrons (MLPs) to model the geometry and radiance fields of a scene. Leveraging the volumetric rendering equation and the inherent continuity and smoothness of MLPs, NeRF achieves high-quality scene reconstruction from a set of posed images, establishing itself as the state-of-the-art (SOTA) method for novel view

synthesis. Subsequent research has extended the utility of NeRF to various applications, including mesh reconstruction [25, 46, 52], inverse rendering [29, 42, 56, 63], optimization of camera parameters [27, 36, 47, 48], few-shot learning [12, 51, 55], and anti-aliasing [1–3].

However, this stream of methods relies on ray casting rather than rasterization to determine the color of each pixel. Consequently, every sampling point along the ray necessitates querying the MLPs, leading to significantly slow rendering speed and prolonged training convergence. This limitation substantially impedes their application in large-scene modeling and real-time rendering.

To reduce the training time of MLP-based NeRF methods and improve rendering speed, subsequent work has enhanced NeRF’s efficiency in various ways. Structure-based techniques [8, 13, 16, 38, 61] have sought to improve inference or training efficiency by caching or distilling the implicit neural representation into more efficient data structures. Hybrid methods [28, 43] increase efficiency by incorporating explicit voxel-based data structures. Factorization methods [5, 9, 15, 17] apply a low-rank tensor assumption to decompose the scene into low-dimensional planes or vectors, achieving better geometric consistency. Compared to continuous implicit representations, the convergence of individual voxels in the grid is independent, significantly reducing training time. Additionally, Instant-NGP [33] utilizes a hash grid with a corresponding CUDA implementation for faster feature querying, enabling rapid training and interactive rendering of neural radiance fields.

Despite achieving higher quality and faster rendering, these methods have not fundamentally overcome the substantial query overhead associated with ray casting. As a result, a notable gap remains before achieving real-time rendering. In this work, we build upon the recent 3D-GS [21], a point-based rendering method that leverages rasterization. Compared to ray casting-based methods, it significantly enhances both training and rendering speed.

2.2. Point-based Neural Radiance Fields

Point-based representations, similar to triangle mesh-based methods, can exploit the highly efficient rasterization pipeline of modern GPUs to achieve real-time rendering. Although these methods offer breakneck rendering speeds and are well-suited for editing tasks, they often suffer from holes and outliers, leading to artifacts in the rendered images. This issue arises from the discrete nature of point clouds, which can create gaps in the primitives and, consequently, in the rendered image.

To address these discontinuity issues, differentiable point-based rendering [14, 22, 23, 60] has been extensively explored for fitting complex geometric shapes. Notably, Zhang et al. [62] employ differentiable surface splatting and utilize a radial basis function (RBF) kernel to compute

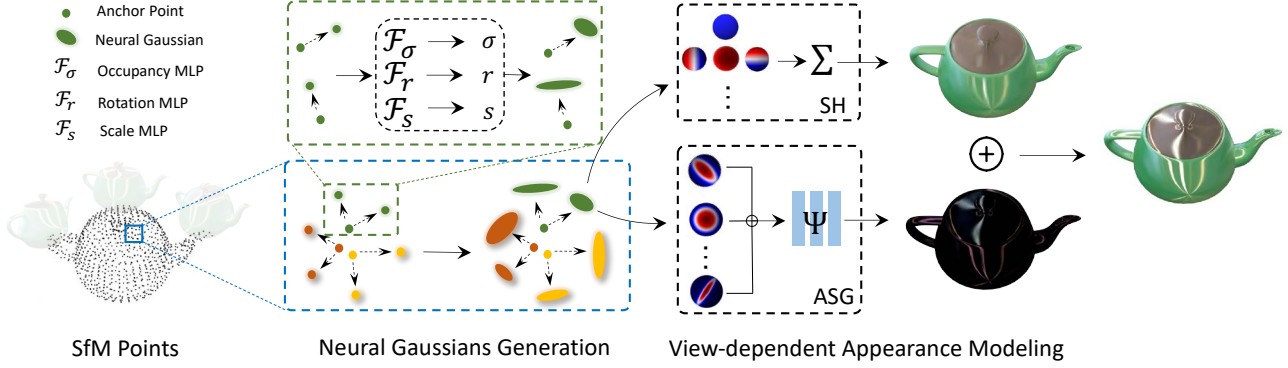


Figure 2. **Pipeline of our proposed Spec-Gaussian.** 优化过程从COLMAP生成的SfM点或随机生成的点开始，作为锚点高斯的初始状态。在视图体内，每个可见锚点高斯会生成 k 个神经高斯，使用相应的偏移量。它们的属性，如不透明度 σ 、旋转 r 和缩放 s ，通过相应的小型MLP解码。为了克服低阶SH和纯MLP在建模高频信息方面的局限性，我们另外使用ASG结合特征解耦MLP来建模每个神经高斯的视角依赖外观。随后，不透明度 $\sigma > 0$ 的神经高斯通过可微高斯光栅化管道渲染，有效捕捉场景中的镜面高光和各向异性。

the contribution of each point to each pixel.

Recently, 3D-GS [21] has employed anisotropic 3D Gaussians, initialized from Structure from Motion (SfM), to represent 3D scenes. The innovative densification mechanism and CUDA-customized differentiable Gaussian rasterization pipeline of 3D-GS have not only achieved state-of-the-art (SOTA) rendering quality but also significantly surpassed the threshold of real-time rendering. Many concurrent works have rapidly extended 3D-GS to a variety of downstream applications, including dynamic scenes [18, 24, 31, 50, 57, 58], text-to-3D generation [10, 11, 26, 44, 59], avatars [19, 35, 39, 64, 65], and scene editing [7, 53].

Despite achieving SOTA results on commonly used benchmark datasets, 3D-GS still struggles to model scenes with specular and reflective components, which limits its practical application in real-time rendering at the photorealistic level. In this work, by replacing spherical harmonics (SH) with an anisotropic spherical Gaussian (ASG) appearance field, we have enabled 3D-GS to model complex specular scenes more effectively. Furthermore, this improvement enhances rendering quality in general scenes without significantly impacting rendering speed.

3. Method

我们方法的概述如图2所示。模型的输入是一组静态场景的已定位图像，以及通过SfM[40]获得的稀疏点云。我们方法的核心是使用ASG外观场来替代SH，以建模3D高斯的外观（详见第3.2节）。为了减少ASG带来的存储开销和渲染速度压力，我们设计了一种混合高斯模型，利用稀疏锚点高斯来促进神经高斯的生成，以建模3D场景（详见第3.3节）。最后，我们引入了一种简单而有效的粗到细训练策略，以减少现实场景中的浮动物体（详见第3.4节）。

3.1. Preliminaries

3.1.1 3D Gaussian Splatting

3D-GS [21] is a point-based method that employs anisotropic 3D Gaussians to represent scenes. Each 3D Gaussian is defined by a center position \mathbf{x} , opacity σ , and a 3D covariance matrix Σ , which is decomposed into a quaternion \mathbf{r} and scaling \mathbf{s} . The view-dependent appearance of each 3D Gaussian is represented using the first three orders of spherical harmonics (SH). This method not only retains the rendering details offered by volumetric rendering but also achieves real-time rendering through a CUDA-customized differentiable Gaussian rasterization process. Following [66], the 3D Gaussians can be projected to 2D using the 2D covariance matrix Σ' , defined as:

$$\Sigma' = JV\Sigma V^T J^T, \quad (1)$$

where J is the Jacobian of the affine approximation of the projective transformation, and V represents the view matrix, transitioning from world to camera coordinates. To facilitate learning, the 3D covariance matrix Σ is decomposed into two learnable components: the quaternion \mathbf{r} , representing rotation, and the 3D-vector \mathbf{s} , representing scaling. The resulting Σ is thus represented as the combination of a rotation matrix R and scaling matrix S as:

$$\Sigma = RSS^T R^T. \quad (2)$$

The color of each pixel on the image plane is then rendered through a point-based volumetric rendering (alpha blending) technique:

$$C(\mathbf{p}) = \sum_{i \in N} T_i \alpha_i c_i, \quad \alpha_i = \sigma_i e^{-\frac{1}{2}(\mathbf{p}-\mu_i)^T \Sigma'(\mathbf{p}-\mu_i)}, \quad (3)$$

where \mathbf{p} denotes the pixel coordinate, T_i is the transmittance defined by $\prod_{j=1}^{i-1} (1 - \alpha_j)$, c_i signifies the color of the sorted Gaussians associated with the queried pixel, and μ_i represents the coordinates of the 3D Gaussians when projected onto the 2D image plane.

3.1.2 Anisotropic Spherical Gaussian.

各向异性球面高斯 (ASG) [54] 在传统渲染管线中被设计用来高效地近似光照和阴影。不同于球面高斯 (SG), ASG 已被证明能够以相对较少的数量有效地表示各向异性场景。除了保留 SG 的基本属性外, ASG 还具有旋转不变性并且可以表示全频率信号。ASG 函数定义如下:

$$ASG(\nu | [\mathbf{x}, \mathbf{y}, \mathbf{z}], [\lambda, \mu], \xi) = \xi \cdot S(\nu; \mathbf{z}) \cdot e^{-\lambda(\nu \cdot \mathbf{x})^2 - \mu(\nu \cdot \mathbf{y})^2}, \quad (4)$$

其中, ν 是作为函数输入的单位方向; \mathbf{x} 、 \mathbf{y} 和 \mathbf{z} 分别对应切线、双切线和反射轴, 并且相互正交; $\lambda \in \mathbb{R}^1$ 和 $\mu \in \mathbb{R}^1$ 是 \mathbf{x} 轴和 \mathbf{y} 轴的锐度参数, 满足 $\lambda, \mu > 0$; $\xi \in \mathbb{R}^2$ 是反射幅度; S 是平滑项, 定义为 $S(\nu; \mathbf{z}) = \max(\nu \cdot \mathbf{z}, 0)$ 。

受到 ASG 在建模复杂各向异性场景中强大能力的启发, 我们提出将 ASG 集成到高斯溅射中, 以结合经典模型和新渲染管线的力量, 从而提高质量。对于 N 个 ASG, 我们预定义了正交基 \mathbf{x} 、 \mathbf{y} 和 \mathbf{z} , 并将它们初始化为均匀分布在半球上。在训练过程中, 我们允许剩余的 ASG 参数 λ 、 μ 和 ξ 是可学习的。我们使用反射方向 ω_r 作为输入来查询 ASG, 以建模视角依赖的镜面信息。注意, 我们为每个 3D 高斯使用 $N = 32$ 个 ASG。

3.2. Anisotropic View-Dependent Appearance

3.2.1 ASG Appearance Field for 3D Gaussians.

尽管 SH 已经使视角依赖的场景建模成为可能, 但低阶 SH 的低频特性使其难以建模具有复杂光学现象 (如镜面高光和各向异性效果) 的场景。因此, 我们提出使用基于方程 (4) 的 ASG 外观场来建模每个 3D 高斯的外观, 而不是使用 SH。然而, 引入 ASG 会增加每个 3D 高斯的特征维度, 从而提高模型的存储开销。为了解决这个问题, 我们采用一个紧凑的可学习 MLP Θ 来预测 N 个 ASG 的参数, 每个高斯仅携带附加的局部特征 $\mathbf{f} \in \mathbb{R}^{24}$ 作为 MLP 的输入:

$$\Theta(\mathbf{f}) \rightarrow \{\lambda, \mu, \xi\}_N. \quad (5)$$

为了更好地区分高频和低频信息, 并进一步帮助 ASG 拟合高频的镜面细节, 我们将颜色 c 分解为漫反射和镜面反射成分:

$$c = c_d + c_s, \quad (6)$$

其中 c_d 表示漫反射颜色, 使用 SH 的前三阶来建模, c_s 是通过 ASG 计算得到的镜面颜色。我们将这种综合的外观建模方法称为 ASG 外观场。

尽管 ASG 理论上增强了 SH 建模各向异性的能力, 但直接使用 ASG 来表示每个 3D 高斯的镜面颜色仍然不足以准确建模各向异性和镜面反射成分, 如图 5 所示。受 [15] 的启发, 我们不直接使用 ASG 来表示颜色, 而是利用 ASG 来建模每个 3D 高斯的潜在特征。这个潜在特征包含各向异性信息, 然后输入到一个小型特征解耦 MLP Ψ 中, 以确定最终的镜面颜色:

$$\begin{aligned} \Psi(\kappa, \gamma(\mathbf{d}), \langle n, -\mathbf{d} \rangle) &\rightarrow c_s, \\ \kappa &= \bigoplus_{i=1}^N ASG(\omega_r | [\mathbf{x}, \mathbf{y}, \mathbf{z}], [\lambda_i, \mu_i], \xi_i) \end{aligned} \quad (7)$$

其中, κ 是由 ASG 得到的潜在特征, \bigoplus 表示连接操作, γ 表示位置编码, \mathbf{d} 是从摄像机指向每个 3D 高斯的单位视线方向, n 是每个 3D 高斯的法线, 将在第 3.2.2 节中讨论, ω_r 是单位反射方向。这种策略显著增强了 3D-GS 模型处理具有复杂光学现象的场景的能力, 而纯 ASG 和纯 MLP 都无法像我们的方法那样有效地实现各向异性外观建模。

3.2.2 Normal Estimation

直接估计 3D 高斯的法线存在挑战, 因为 3D-GS 由一系列离散实体组成, 每个实体代表一定范围内的局部空间, 而没有形成连续的表面。法线的计算通常需要连续的表面, 而 3D-GS 中每个实体的各向异性形状进一步复杂化了法线的确定。我们采用了 [20, 41] 中的方法, 将每个高斯的最短轴作为其法线。这种方法基于观察到在优化过程中, 3D 高斯往往逐渐变平, 使得最短轴可以作为法线的合理近似。

反射方向 ω_r 可以使用视线方向和局部法线向量 n 来推导:

$$\omega_r = 2(\omega_o \cdot n) \cdot n - \omega_o, \quad (8)$$

其中, $\omega_o = -\mathbf{d}$ 是从每个 3D 高斯指向摄像机的单位视线方向。我们使用反射方向 ω_r 来查询 ASG, 从而更好地插值包含各向异性信息的潜在特征。实验结果表明, 尽管这种无监督法线估计不能生成与真实世界对齐的物理上准确的法线, 但足以产生相对准确的反射方向, 以帮助 ASG 拟合高频信息。

3.3. Anchor-Based Gaussian Splatting

3.3.1 Neural Gaussian Derivation with ASG Appearance Field.

虽然 ASG 外观场显著提高了 3D-GS 模型建模镜面和各向异性特性的能力, 但与纯 SH 相比, 它引入了额外的存储和计算开销, 这是由于每个高斯都关联有附加的局部特征 \mathbf{f} 。虽然在有界场景中仍然可以实现超过 100 帧每秒的实时渲染, 但在现实世界中无界场景中由 ASG 引起的存储开销大幅增加和渲染速度降低是不可接受的。受 [30] 的启发, 我们采用基于锚点的高斯溅射来减少存储开销和渲染所需的 3D 高斯数量, 从而加速渲染。

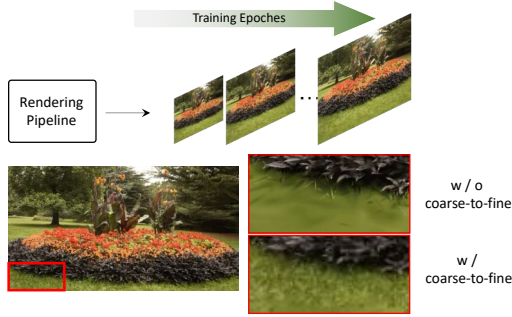


Figure 3. Using a coarse-to-fine strategy, our approach is able to optimize the scene in a progressive manner and eliminate the floaters efficiently.

与3D-GS中每个实体携带的属性不同，每个锚定高斯都携带一个位置坐标 $\mathbf{P}_v \in \mathbb{R}^3$ ，一个局部特征 $\mathbf{f}_v \in \mathbb{R}^{32}$ ，一个位移因子 $\eta_v \in \mathbb{R}^3$ ，以及 k 个可学习的偏移 $\mathbf{O}_v \in \mathbb{R}^{k \times 3}$ 。我们使用从COLMAP [40] 获得的稀疏点云来初始化每个锚定3D高斯，作为体素中心以指导神经高斯的生成。锚定高斯的位置 \mathbf{P}_v 初始化为：

$$\mathbf{P}_v = \left\{ \left\lfloor \frac{\mathbf{P}}{\epsilon} + 0.5 \right\rfloor \right\} \cdot \epsilon, \quad (9)$$

其中 \mathbf{P} 是点云的位置， ϵ 表示体素大小， $\{\cdot\}$ 表示去除重复的锚定点。

然后我们使用锚定高斯来指导神经高斯的生成，后者具有与普通3D-GS相同的属性。对于视锥体内每个可见的锚定高斯，我们生成 k 个神经高斯并预测它们的属性（见图2）。神经高斯的位置 \mathbf{x} 计算如下：

$$\{\mathbf{x}_0, \dots, \mathbf{x}_{k-1}\} = \mathbf{P}_v + \{\mathbf{O}_0, \dots, \mathbf{O}_{k-1}\} \cdot \eta_v, \quad (10)$$

其中 \mathbf{P}_v 代表锚定高斯的位置，对应于 k 个神经高斯。不透明度 σ 通过一个小型MLP计算：

$$\{\sigma_0, \dots, \sigma_{k-1}\} = \mathcal{F}_\sigma(\mathbf{f}_v, \delta_{cv}, \mathbf{d}_{cv}), \quad (11)$$

其中 δ_{cv} 表示锚定高斯与相机之间的距离， \mathbf{d}_{cv} 是从相机指向锚定高斯的单位方向。每个神经高斯的旋转 r 和缩放 s 类似地通过对应的小型MLP \mathcal{F}_r 和 \mathcal{F}_s 推导得出。

由于ASG建模的各向异性在空间中是连续的，因此可以将其压缩到低维空间。得益于锚定高斯的指导，锚定特征 \mathbf{f}_v 可以直接用于压缩 N 个ASG，从而进一步减小存储压力。为了使神经高斯的ASG具有位置感知能力，我们引入单位视角方向来解压ASG参数。因此，式(5) 中的ASG参数预测被修改如下：

$$\Theta(\mathbf{f}_v, \mathbf{d}_{cn}) \rightarrow \{\lambda, \mu, \xi\}_N, \quad (12)$$

其中 \mathbf{d}_{cn} 表示从相机到每个神经高斯的单位视角方向。此外，我们将神经高斯的漫反射部分 $c_d = \phi(\mathbf{f}_v)$ 设置为通过MLP ϕ 直接预测，以确保漫反射组件的平滑性并降低收敛难度。

3.3.2 锚定高斯的自适应控制

为了使3D-GS能够表示场景细节同时去除冗余实体，我们根据神经高斯的梯度和不透明度自适应地调整锚定高斯的数量。按照[21, 30] 的方法，我们每100次训练迭代计算每个锚定高斯生成的 k 个神经高斯的平均梯度，记作 ∇_v 。对于梯度大于 τ_g 的锚定高斯，我们将其密化。在实际操作中，我们遵循[30] 的方法，将空间量化为多分辨率体系，以允许在不同粒度添加新的锚定高斯：

$$\epsilon^{(l)} = \epsilon \cdot \beta / 4^l, \quad \tau_g^{(l)} = \tau_g \cdot 2^l, \quad (13)$$

其中 l 表示新锚定高斯的级别， $\epsilon^{(l)}$ 是第 l 级新生成锚定高斯的体素大小， β 是一个增长因子。与[30] 不同，为了减少由于锚定点过度密化而导致的过拟合，我们引入了层次选择。只有梯度大于 $\text{Quantile}(\nabla_v, 2^{-(l+1)})$ 的锚定高斯才会在相应体素中心的第 l 级被密化。

为了消除无关紧要的锚点，我们每100次训练迭代累积其相关神经高斯的不透明度值，记作 $\bar{\sigma}$ 。如果一个锚定高斯未能生成具有满意不透明度水平的神经高斯，即 $\bar{\sigma} < \tau_o$ ，我们将其移除。

3.4. Coarse-to-fine Training

我们观察到，在许多现实场景中，3D-GS往往会过拟合训练数据，导致从新视点渲染图像时出现大量的浮动物。在现实世界数据集中，一个常见的挑战是相机位姿估计的不准确，特别是在大场景中尤为明显。Scaffold-GS [30] 通过锚定3D-GS，对几何结构施加稀疏体素约束，创建分层3D-GS表示。虽然这种分层方法提高了3D-GS建模复杂几何形状的能力，但它并没有解决过拟合问题，而且在许多情况下，加剧了场景背景中浮动物的出现。

为了减轻现实场景中浮动物的出现，我们提出了一种由粗到细的训练机制。我们认为3D-GS过拟合的倾向源于过分关注每个3D高斯对特定像素及其邻近像素的贡献，而不是考虑更广泛的全局信息。因此，我们决定逐步从低分辨率到高分辨率训练3D-GS：

$$r(i) = \min(\lfloor r_s + (r_e - r_s) \cdot i / \tau \rfloor, r_e), \quad (14)$$

其中 $r(i)$ 是第 i 次训练迭代时的图像分辨率， r_s 是起始图像分辨率， r_e 是目标渲染的最终图像分辨率， τ 是阈值迭代次数，经验值设定为20k。

这种训练方法使3D-GS在训练的早期阶段能够从图像中学习全局信息，从而减少对训练图像局部区域的过拟合，并消除大量在新视点渲染时的浮动物。此外，由于初始阶段采用较低分辨率训练，这种机制使训练时间减少了约20

3.5. 损失函数

除了3D-GS中的颜色损失[21] 外，我们还引入了一个正则化损失，以鼓励神经高斯保持较小且尽量不重叠。

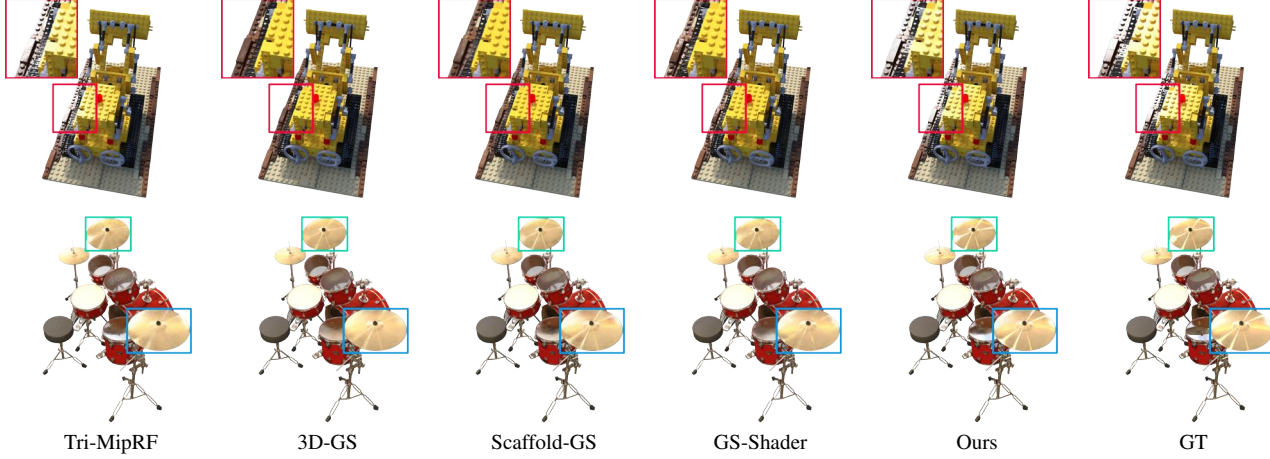


Figure 4. NeRF数据集上的可视化结果。我们的方法成功实现了局部镜面高光建模，这是其他基于3D-GS的方法无法做到的，同时保持了快速渲染速度。与基于NeRF的方法Tri-MipRF相比，我们显著增强了各向异性材料的建模能力。

Dataset Method — Metrics	Mip-NeRF360					Tanks&Temples					Deep Blending				
	PSNR ↑	SSIM ↑	LPIPS ↓	FPS	Mem	PSNR ↑	SSIM ↑	LPIPS ↓	FPS	Mem	PSNR ↑	SSIM ↑	LPIPS ↓	FPS	Mem
Mip-NeRF360	27.69	0.792	0.237	0.06	8.6MB	22.22	0.759	0.257	0.14	8.6MB	29.40	0.901	0.245	0.09	8.6MB
iNGP	25.59	0.699	0.331	9.43	48MB	21.72	0.723	0.330	14.4	48MB	23.62	0.797	0.423	2.79	48MB
Plenoxels	23.08	0.626	0.463	6.79	2.1GB	21.08	0.719	0.379	13.0	2.3GB	23.06	0.795	0.510	11.2	2.7GB
3D-GS	27.47	0.812	0.222	115	748MB	23.71	0.844	0.178	169	432MB	29.65	0.899	0.247	130	662MB
Scaffold-GS	27.66	0.807	0.236	96	203MB	23.96	0.853	0.177	143	89.5MB	30.21	0.906	0.254	179	63.5MB
Ours-w/o anchor	27.81	0.810	0.223	25	1.02GB	23.94	0.846	0.181	37	563MB	29.71	0.901	0.250	32	793MB
Ours	28.01	0.812	0.222	70	245MB	24.58	0.855	0.174	111	96.5MB	30.45	0.906	0.252	132	68MB

Table 1. Quantitative evaluation of our method compared to previous work on real-world datasets. We report PSNR, SSIM, LPIPS(VGG) and color each cell as **best**, **second best** and **third best**. Our method has overall achieved the best rendering quality, while also striking a good balance between FPS and the storage memory of 3D Gaussians.

Dataset Method — Metrics	NeRF Synthetic				
	PSNR ↑	SSIM ↑	LPIPS ↓	FPS	Mem
iNGP-Base	33.18	0.963	0.045	~10	~13MB
Mip-NeRF	33.09	0.961	0.043	~1	~10MB
Tri-MipRF	33.65	0.963	0.042	~5	~60MB
3D-GS	33.32	0.970	0.031	415	69MB
GS-Shader	33.38	0.968	0.029	97	29MB
Scaffold-GS	33.68	0.967	0.034	240	19MB
Ours-w anchor	33.96	0.969	0.032	162	20MB
Ours	34.12	0.971	0.028	105	79MB

Table 2. Quantitative results on NeRF synthetic dataset. Our method achieves a rendering quality that surpasses NeRF-based methods, without excessively reducing FPS.

因此，所有可学习参数和MLP的总损失函数如下：

$$\mathcal{L} = (1 - \lambda_{D-SSIM})\mathcal{L}_1 + \lambda_{D-SSIM}\mathcal{L}_{D-SSIM} + \lambda_{reg}\mathcal{L}_{reg},$$

$$\mathcal{L}_{reg} = \frac{1}{N} \sum_{i=1}^{N_n} \text{Prod}(s_i), \quad (15)$$

其中 N_n 是神经高斯的数量， $\text{Prod}(\cdot)$ 计算每个神经高斯的尺度 s_i 的乘积。我们在实验中统一使用 $\lambda_{D-SSIM} = 0.2$ 和 $\lambda_{reg} = 0.01$ 。

4. Experiments

在本节中，我们展示了我们方法的定量和定性结果。为了评估其有效性，我们将其与几种最新的先进方法在不同数据集上进行了比较。我们将每个单元格着色为**最佳**、**次佳**和**第三佳**。我们的方法在建模复杂镜面和各向异性特征方面表现出色，这在NeRF、NSVF和我们的“各向异性合成”数据集上的比较中得到了证实。此外，我们通过在3D-GS中的所有场景下比较其性能，展示了其多功能性，进一步证明了我们方法的鲁棒性。

4.1. Implementation Details

我们使用PyTorch [37] 实现了我们的框架，并修改了可微高斯光栅化以包含深度可视化。对于ASG外观场，特征解耦MLP Ψ 由3层组成，每层有64个隐藏单元，视角方向的位置编码阶数为2。在基于锚点的高斯散点方面，我们为锚点高斯的密化设立了三个级别，密化阈值 τ_g 设置为0.0002，修剪阈值 τ_o 设置为0.005，每个锚点的神经高斯数量 k 设置为10。在有界场景中，我们的体素大小 ϵ 设置为0.001，增长因子 β 为4。对于Mip-360场景，体素大小仍为0.001，但增长因子 β 增加到16。关于由粗到细的训练，我们从一个8倍下采样的分辨率 r_s 开始。为了进一步加快渲染速度，

Dataset Method — Metrics	NSVF Synthetic				
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	FPS	Mem
TensorRF	36.52	0.982	0.026	~ 1.5	~ 65 MB
Tri-MipRF	34.58	0.973	0.030	~ 5	~ 60 MB
NeuRBF	37.80	0.986	0.019	~ 1	~ 580 MB
3D-GS	37.07	0.987	0.015	403	66MB
GS-Shader	33.85	0.981	0.020	68	33MB
Scaffold-GS	36.43	0.984	0.017	218	17MB
Ours-w anchor	37.71	0.987	0.015	142	18MB
Ours	38.35	0.988	0.013	91	99MB

Table 3. **Quantitative results on NSVF synthetic dataset.** Our method achieved significantly higher rendering quality than 3D-GS, and it also surpassed NeRF-based methods.

Dataset Method — Metrics	Anisotropic Synthetic				
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	FPS	Mem
3D-GS	33.82	0.966	0.062	345	47MB
Scaffold-GS	35.34	0.972	0.052	234	27MB
Ours-w anchor	36.76	0.976	0.046	180	28MB
Ours	37.42	0.977	0.047	119	59MB

Table 4. **Quantitative results on our "Anisotropic Synthetic" dataset.**

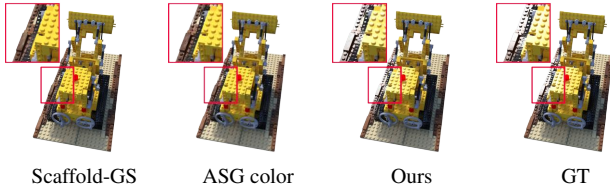


Figure 5. **Ablation on ASG feature decoupling MLP.** We show that directly using ASG to model color leads to the failure in modeling anisotropy and specular highlights. By decoupling the ASG features through MLP, we can realistically model complex optical phenomena.

我们预过滤可见的锚点高斯，并只允许那些不透明度 $\sigma_n > 0$ 的神经高斯通过ASG外观场和高斯光栅化管道。所有实验均在Tesla V100上进行，FPS测量是在具有24GB内存的NVIDIA RTX 3090上进行的。

4.2. Results and Comparisons

Synthetic Bounded Scenes. We used the NeRF, NSVF, and our "Anisotropic Synthetic" datasets as the experimental datasets for synthetic scenes. Our comparisons were made with the most relevant state-of-the-art methods, including 3D-GS [21], Scaffold-GS [30], GaussianShader [20], and several NeRF-based methods such as NSVF [28], TensorRF [6], NeuRBF [9], and Tri-MipRF [17]. To ensure a fair comparison, we used the rendering metrics of the NeRF and NSVF datasets as reported in the baseline papers. For scenes not reported in the baseline papers, we trained the baselines from scratch using the released codes and their default configurations.

As shown in Fig. 4, Figs. 7-8, and Tabs. 2-4, our method

achieved the highest performance in terms of PSNR, SSIM, and LPIPS. It also significantly improved upon the issues that 3D-GS faced in modeling high-frequency specular highlights and complex anisotropy. See per-scene results in the supplementary materials.

Real-world Unbounded Scenes. To verify the versatility of our method in real-world scenarios, we used the same real-world dataset as in 3D-GS [21]. As shown in Tab. 1, our method achieves rendering results comparable to state-of-the-art methods on the Deep Blending dataset and surpasses them on Mip-NeRF 360 and Tanks&Temples. Furthermore, our method effectively balances FPS, storage overhead, and rendering quality. It enhances rendering quality without excessively increasing storage requirements or significantly reducing FPS. As illustrated in Fig. 6, our method has also significantly improved the visual effect. It removes a large number of floaters in outdoor scenes and successfully models the high-frequency specular highlights in indoor scenes. This demonstrates that our approach is not only adept at modeling complex specular scenes but also effectively improves rendering quality in general scenarios.

4.3. Ablation Study

4.3.1 ASG特征解耦MLP

我们进行了消融研究，以评估使用ASG输出特征的有效性，这些特征通过MLP Ψ 解耦，得出最终的镜面颜色。如图5所示，直接使用ASG输出颜色无法建模镜面和各向异性组件。相比直接使用MLP进行颜色建模（如Scaffold-GS [30]），ASG可以编码更高频率的各向异性特征。这一能力帮助MLP学习复杂的光学现象，从而带来更准确和详细的渲染结果。

4.3.2 由粗到细的训练

我们进行了消融研究，以评估由粗到细（c2f）训练的影响。如图10所示，3D-GS和Scaffold-GS在新视点合成中都表现出大量浮动物。由粗到细的训练有效地减少了浮动物的数量，缓解了3D-GS在现实场景中常遇到的过拟合问题。

5. Conclusion

在本研究中，我们介绍了*Spec-Gaussian*，这是一种新颖的3D高斯分割方法，具有各向异性视图依赖的外观。利用ASG的强大能力，我们的方法有效克服了原始3D-GS在渲染带有镜面高光和各向异性的场景时遇到的挑战。此外，我们创新性地实现了一种由粗到细的训练机制，以消除现实场景中的浮动物。定量和定性实验均表明，我们的方法不仅赋予了3D-GS建模镜面高光和各向异性特征的能力，还增强了3D-GS在一般场景中的整体渲染质量，同时并未显著影响FPS和存储开销。

局限性 尽管我们的方法使3D-GS能够建模复杂的镜面和各向异性特征，但在处理反射方面仍面临挑战。镜面和各向异性效应主要受材料属性影响，而反射则与环境与几何形状密切相关。由于3D-GS缺乏显式几何，我们无法像Ref-NeRF [45] 和NeRO [29] 中使用法线等约束来区分反射和材质纹理。在我们的实验中，我们还观察到，当提供真实的几何信息时，3D-GS在严格约束下与预期结果更加一致，但这会导致渲染质量的某些下降。我们计划在未来的工作中探索3D-GS建模反射的解决方案。

References

- [1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021. 2
- [2] 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.
- [3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 2
- [4] Mario Botsch, Alexander Hornung, Matthias Zwicker, and Leif Kobbelt. High-quality surface splatting on today’s gpus. In *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005.*, pages 17–141. IEEE, 2005. 1
- [5] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2
- [6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022. 1, 7
- [7] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting, 2023. 3
- [8] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. *arXiv preprint arXiv:2208.00277*, 2022. 2
- [9] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4182–4194, 2023. 2, 7
- [10] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *arXiv preprint arXiv:2309.16585*, 2023. 3
- [11] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023. 3
- [12] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2
- [13] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021. 2
- [14] Markus Gross and Hanspeter Pfister. *Point-based graphics*. Elsevier, 2011. 2
- [15] Kang Han and Wei Xiang. Multiscale tensor decomposition and rendering equation encoding for view synthesis. In *The IEEE / CVF Computer Vision and Pattern Recognition Conference*, pages 4232–4241, 2023. 2, 4
- [16] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021. 2
- [17] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19774–19783, 2023. 2, 7, 10
- [18] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. pages 1–11, 2023. 3
- [19] Yuheng Jiang, Zhehao Shen, Penghao Wang, Zhuo Su, Yu Hong, Yingliang Zhang, Jingyi Yu, and Lan Xu. Hifi4g: High-fidelity human performance rendering via compact gaussian splatting. *arXiv preprint arXiv:2312.03461*, 2023. 3
- [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. *arXiv preprint arXiv:2311.17977*, 2023. 4, 7
- [21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1, 2, 3, 5, 7, 10
- [22] Leonid Keselman and Martial Hebert. Approximate differentiable rendering with algebraic surfaces. In *European Conference on Computer Vision*, pages 596–614. Springer, 2022. 2
- [23] Leonid Keselman and Martial Hebert. Flexible techniques for differentiable rendering with 3d gaussians. *arXiv preprint arXiv:2308.14737*, 2023. 2
- [24] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. *arXiv preprint arXiv:2312.16812*, 2023. 3
- [25] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

- [26] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching, 2023. 3
- [27] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2
- [28] Lingjie Liu, Jiatuo Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 2, 7, 10
- [29] 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. In *SIGGRAPH*, 2023. 2, 8
- [30] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. *arXiv preprint arXiv:2312.00109*, 2023. 4, 5, 7
- [31] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 3
- [32] 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 *ECCV*, 2020. 1, 2
- [33] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multi-resolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 1, 2
- [34] 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. 1
- [35] Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. Ash: Animatable gaussian splats for efficient and photoreal human rendering. 2023. 3
- [36] Keunhong Park, Philipp Henzler, Ben Mildenhall, Jonathan T Barron, and Ricardo Martin-Brualla. Camp: Camera preconditioning for neural radiance fields. *ACM Transactions on Graphics (TOG)*, 42(6):1–11, 2023. 2
- [37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. 6
- [38] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 2
- [39] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. 2023. 3
- [40] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 4104–4113, 2016. 3, 5
- [41] Yahao Shi, Yanmin Wu, Chenming Wu, Xing Liu, Chen Zhao, Haocheng Feng, Jingtuo Liu, Liangjun Zhang, Jian Zhang, Bin Zhou, et al. Gir: 3d gaussian inverse rendering for relightable scene factorization. *arXiv preprint arXiv:2312.05133*, 2023. 4
- [42] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021. 2
- [43] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 2
- [44] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 3
- [45] 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. *CVPR*, 2022. 8
- [46] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 2
- [47] Peng Wang, Lingzhe Zhao, Ruijie Ma, and Peidong Liu. BAD-NeRF: Bundle Adjusted Deblur Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4170–4179, 2023. 2
- [48] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF—: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2
- [49] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 12
- [50] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Wang Xinggang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 3
- [51] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. Reconfusion: 3d reconstruction with diffusion priors. *arXiv*, 2023. 2
- [52] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. In *International Conference on Learning Representations (ICLR)*, 2023. 2

- [53] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. *arXiv preprint arXiv:2311.12198*, 2023. 3
- [54] Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical gaussians. *ACM Transactions on Graphics*, 32(6):209:1–209:11, 2013. 2, 4
- [55] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2
- [56] Ziyi Yang, Yanzhen Chen, Xinyu Gao, Yazhen Yuan, Yu Wu, Xiaowei Zhou, and Xiaogang Jin. Sire-ir: Inverse rendering for brdf reconstruction with shadow and illumination removal in high-illuminance scenes. *arXiv preprint arXiv:2310.13030*, 2023. 2
- [57] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 3
- [58] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv 2310.10642*, 2023. 3
- [59] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjuan Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. *arXiv preprint arXiv:2310.08529*, 2023. 3
- [60] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019. 1, 2
- [61] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 2
- [62] Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. Differentiable point-based radiance fields for efficient view synthesis. *arXiv preprint arXiv:2205.14330*, 2022. 2
- [63] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)*, 40(6):1–18, 2021. 2
- [64] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. *arXiv*, 2023. 3
- [65] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars. 2023. 3
- [66] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS'01.*, pages 29–538. IEEE, 2001. 3

A. More Results

In this section, we present the complete quantitative results of our experiments. We report PSNR, SSIM, LPIPS (VGG), and color each cell as **best**, **second best** and **third best**.

A.1. NeRF Synthetic Scenes

As shown in Tabs. 5-7, our method demonstrates the best rendering quality metrics in almost every scene. It’s important to note that the experimental setup for Tri-MipRF [17] differs from other methods. It uses both the training and validation sets as training data, expanding the scale of the model’s data. When its training data is limited to the training set, its metrics suffer a noticeable drop. Nevertheless, to ensure that the experimental results fully reflect the highest performance of each method, and to prevent significant drops in metrics due to differences in experimental environments, we still present the metrics from the Tri-MipRF official paper. Our method achieved more prominent metrics in scenes with notable specular reflection and anisotropy, such as Drums, Lego, and Ship. This demonstrates that our method not only improves the overall rendering quality but also has a more significant advantage in complex specular scenarios.

A.2. NSVF Synthetic Scenes

The NSVF [28] dataset, in comparison to NeRF, features more noticeable metallic specular reflection, as presented in the Wineholder, Steamtrain, and Spaceship scenes. It also includes more complex transmission scenarios, such as Lifestyle. It is important to note that Tri-MipRF fails to converge in the Steam scene with the official code, so we did not report metrics for that scenario. As shown in Tabs. 8-10, we present the per-scene experimental results of PSNR, SSIM, and LPIPS in the supplementary material. The experimental results indicate that compared to other methods based on 3D-GS [21], our method has significant advantages in metallic highlights and complex transmission scenarios. Additionally, we compared it with the SOTA NeRF-based methods based on NeRF. Our approach enables 3D-GS to surpass the latest SOTA of NeRF, achieving high-frequency highlight modeling that 3D-GS couldn’t realize but NeRF could, thereby achieving truly high-quality rendering.

A.3. Anisotropic Synthetic Scenes

“Anisotropic Synthetic” is a synthetic dataset we rendered ourselves, which includes 8 scenes with significant anisotropy. We tested some existing 3D-GS-based methods on “Anisotropic Synthetic.” As shown in Tabs. 11-13, our method achieved a very significant improvement in rendering metrics. Qualitative experiments also demonstrate the significant visual advantages of our method, highlighting the substantial improvement our method brings to

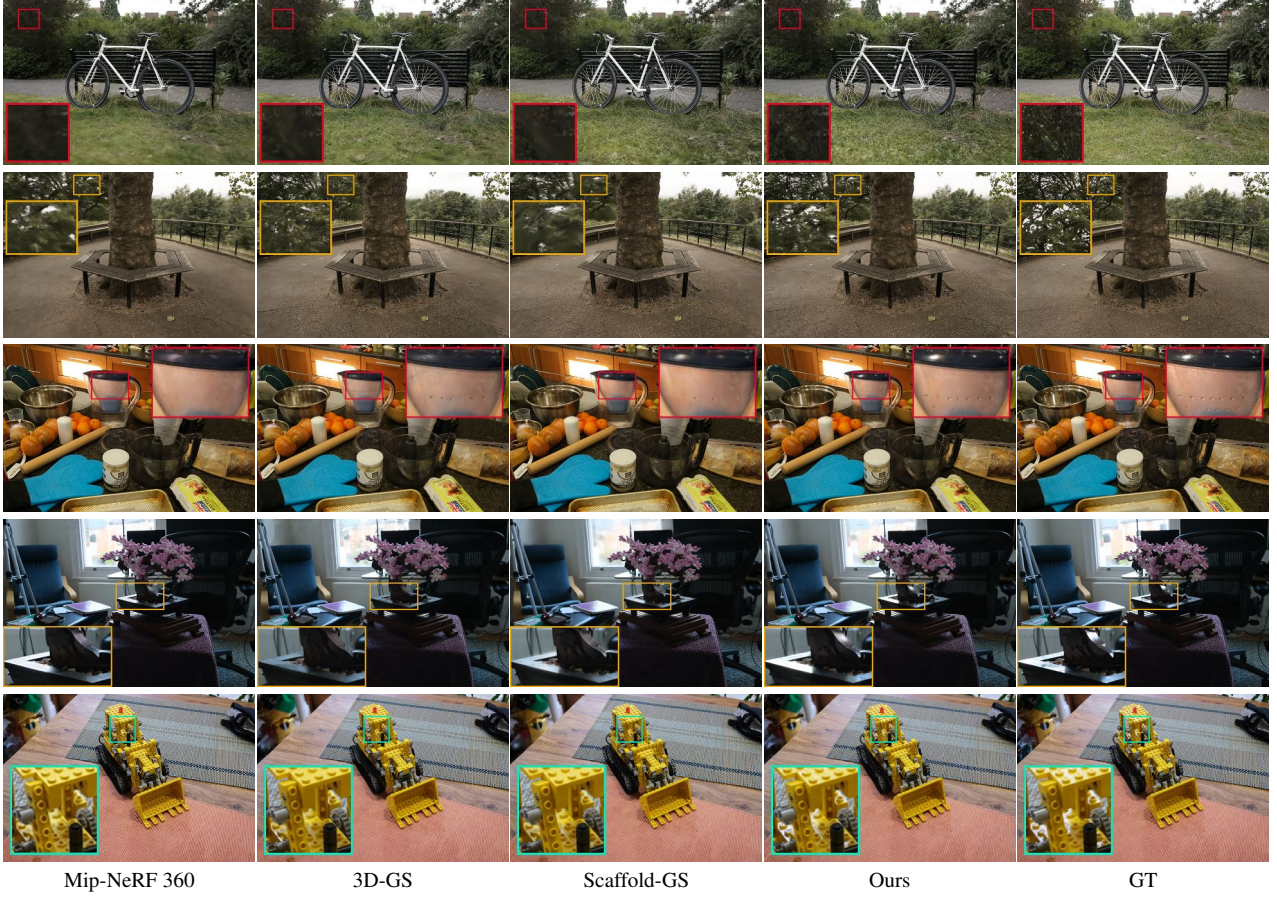


Figure 6. **Visualization on Mip-NeRF 360 dataset.** This clearly demonstrates that our method is capable of modeling complex specular highlights and effectively removing floaters, outperforming other methods in these aspects.

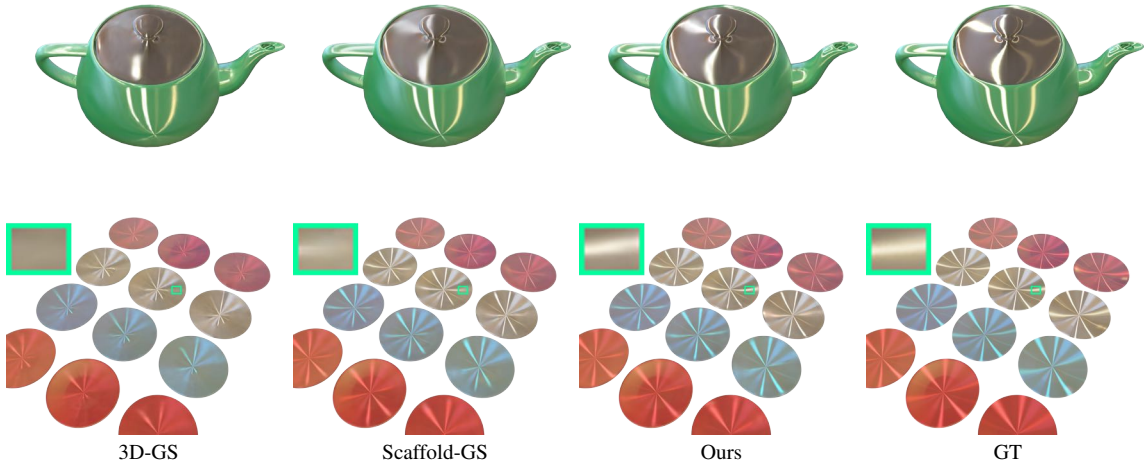


Figure 7. **Visualization on our anisotropic dataset.** We have demonstrated the superiority of our method compared to 3D-GS and scaffold-GS, which models color based on MLP. With the help of ASG, we can model specular highlights and anisotropic parts of the scene more effectively.

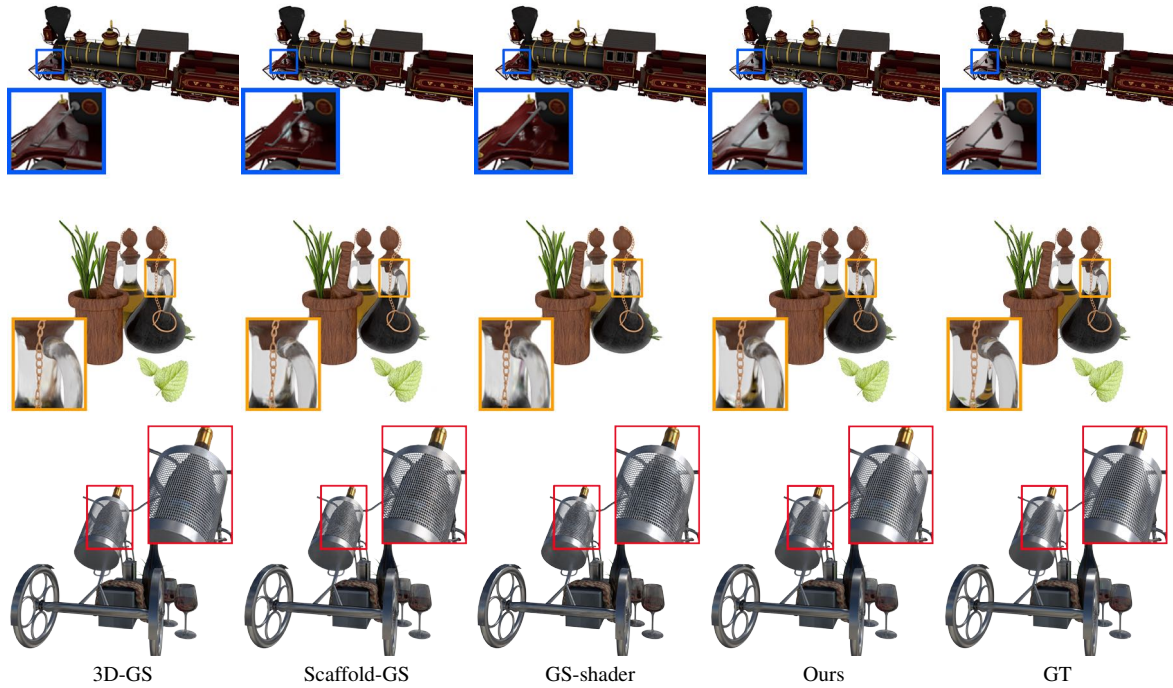


Figure 8. **Visualization on NSVF dataset.** Our method significantly improves the ability to model metallic materials compared to other GS-based methods. At the same time, our method also demonstrates the capability to model refractive parts, reflecting the powerful fitting ability of our method.

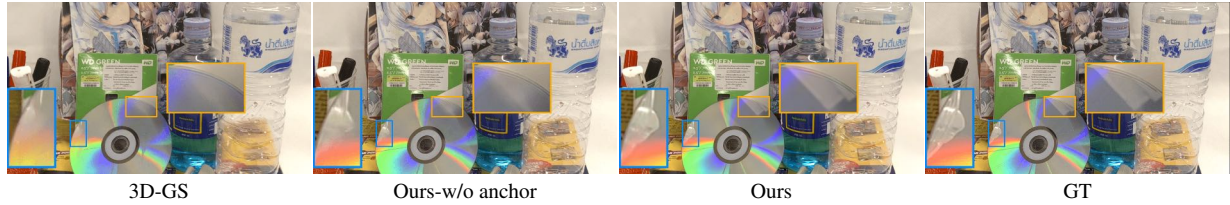


Figure 9. **Ablation on anchor Gaussians.** The shiny scene is borrowed from Nex [49]. This clearly demonstrates that anchor Gaussians can improve the geometry of 3D-GS. Consequently, this enhancement aids in its ability to learn the reflective parts of the scene, as highlighted in the orange and blue boxes.

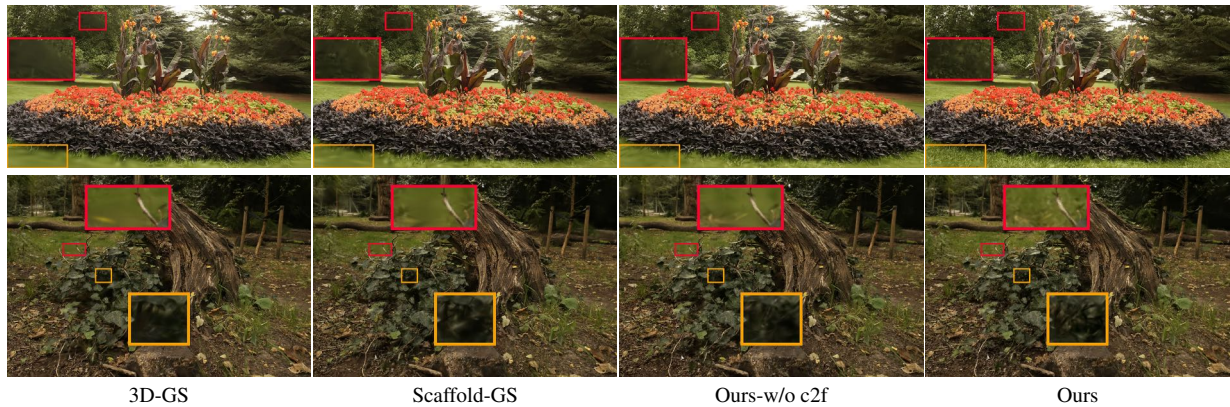


Figure 10. **Ablation on coarse-to-fine training mechanism.** Experimental results demonstrate that our simple yet effective training mechanism can effectively remove floaters in both the background and foreground, thereby alleviating the overfitting problem prevalent in 3D-GS-based methods.

anisotropic parts, thereby enhancing the overall rendering quality.

A.4. Mip-360 Scenes

The MipNeRF-360 scenes include five outdoor and four indoor scenarios. There are several scenes rich in specular reflections, such as bonsai, room, and kitchen. As shown in Tabs. 14-16, our method achieved significant advantages in the four indoor scenes. This reflects our method’s strengths in modeling specular reflections and anisotropy. In outdoor scenes, our method also achieved rendering metrics comparable to the SOTA methods. Furthermore, with the help of the coarse-to-fine training mechanism, our method significantly reduced the number of floaters, resulting in a substantial improvement in visual effects.

A.5. Tanks & Temples and Deep Blending Scenes

To more comprehensively demonstrate the superiority of our method over 3D-GS in terms of rendering quality, we also compared it with the Deep Blending and Tanks & Temples datasets, which were also used in the original 3D-GS paper. As shown in Tab. 17, our method achieved the best metrics in almost all scenes. This also showcases the versatility of our method, indicating that it is not limited to modeling anisotropic and specular scenarios.

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg.
iNGP-Base	35.00	26.02	33.51	37.40	36.39	29.78	36.22	31.10	33.18
Mip-NeRF	35.14	25.48	33.29	37.48	35.70	30.71	36.51	30.41	33.09
Tri-MipRF	36.10	26.59	34.51	38.54	36.15	30.73	37.75	28.78	33.65
GS-Shader	35.83	26.36	34.97	37.85	35.87	30.07	35.23	30.82	33.38
3D-GS	35.36	26.15	34.87	37.72	35.78	30.00	35.36	30.80	33.32
Scaffold-GS	35.28	26.44	35.21	37.73	35.69	30.65	37.25	31.17	33.68
Ours-w anchor	35.57	26.58	35.71	38.12	36.62	30.66	36.81	31.63	33.96
Ours	35.68	26.92	36.14	38.28	36.07	30.85	37.12	31.89	34.12

Table 5. Per-scene PSNR comparison on the NeRF dataset.

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg.
iNGP-Base	0.979	0.937	0.981	0.982	0.982	0.951	0.990	0.896	0.963
Mip-NeRF	0.981	0.932	0.980	0.982	0.978	0.959	0.991	0.882	0.961
Tri-MipRF	0.985	0.939	0.983	0.984	0.982	0.953	0.992	0.879	0.963
GS-Shader	0.987	0.949	0.985	0.985	0.983	0.960	0.991	0.905	0.968
3D-GS	0.988	0.955	0.987	0.985	0.983	0.960	0.992	0.907	0.970
Scaffold-GS	0.985	0.950	0.985	0.983	0.980	0.960	0.992	0.898	0.967
Ours-w anchor	0.986	0.953	0.987	0.985	0.982	0.962	0.992	0.904	0.969
Ours	0.987	0.958	0.988	0.985	0.982	0.963	0.993	0.909	0.971

Table 6. Per-scene SSIM comparison on the NeRF dataset.

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg.
iNGP-Base	0.022	0.071	0.023	0.027	0.017	0.060	0.010	0.132	0.045
Mip-NeRF	0.021	0.065	0.020	0.027	0.021	0.040	0.009	0.138	0.043
Tri-MipRF	0.016	0.066	0.020	0.021	0.016	0.052	0.008	0.136	0.042
GS-Shader	0.012	0.040	0.013	0.019	0.014	0.033	0.006	0.098	0.029
3D-GS	0.011	0.037	0.012	0.020	0.016	0.037	0.006	0.106	0.031
Scaffold-GS	0.013	0.042	0.013	0.023	0.019	0.040	0.008	0.114	0.034
Ours-w anchor	0.013	0.038	0.012	0.022	0.016	0.037	0.007	0.112	0.032
Ours	0.011	0.032	0.011	0.019	0.014	0.032	0.006	0.104	0.028

Table 7. Per-scene LPIPS (VGG) comparison on the NeRF dataset.

	Bike	Life	Palace	Robot	Space	Steam	Toad	Wine	Avg.
NeRF	31.77	31.08	31.76	28.69	34.66	30.84	29.42	28.23	30.81
NSVF	37.75	34.60	34.05	35.24	39.00	35.13	33.25	32.04	35.13
TensorRF	39.23	34.51	37.56	38.26	38.60	37.87	34.85	31.32	36.52
Tri-MipRF	36.98	33.98	36.55	33.49	37.60	-	33.48	29.97	34.58
NeuRBF	40.71	36.08	38.93	39.13	40.44	38.35	35.73	32.99	37.80
3D-GS	40.76	33.19	38.89	39.16	36.80	37.67	37.33	32.76	37.07
Scaffold-GS	39.87	35.00	38.53	37.92	34.36	37.12	36.29	32.32	36.43
GS-Shader	37.38	27.36	36.55	37.00	32.61	35.27	34.50	30.16	33.85
Ours-w anchor	40.63	35.56	38.95	38.52	39.47	37.98	36.55	34.04	37.71
Ours	41.52	36.13	39.10	39.60	40.02	38.28	37.43	34.73	38.35

Table 8. Per-scene PSNR comparison on the NSVF dataset.

	Bike	Life	Palace	Robot	Space	Steam	Toad	Wine	Avg.
NeRF	0.970	0.946	0.950	0.960	0.980	0.966	0.920	0.920	0.952
NSVF	0.991	0.971	0.969	0.988	0.991	0.986	0.968	0.965	0.979
TensoRF	0.993	0.968	0.979	0.994	0.989	0.991	0.978	0.961	0.982
Tri-MipRF	0.990	0.962	0.973	0.985	0.986	-	0.968	0.945	0.973
NeuRBF	0.995	0.977	0.985	0.995	0.993	0.993	0.983	0.972	0.986
3D-GS	0.994	0.979	0.983	0.994	0.991	0.993	0.985	0.975	0.987
Scaffold-GS	0.993	0.979	0.981	0.995	0.985	0.992	0.982	0.971	0.984
GS-Shader	0.992	0.964	0.979	0.994	0.985	0.990	0.980	0.966	0.981
Ours-w anchor	0.994	0.979	0.982	0.994	0.993	0.992	0.984	0.975	0.987
Ours	0.995	0.982	0.984	0.995	0.994	0.994	0.985	0.978	0.988

Table 9. Per-scene SSIM comparison on the NSVF dataset.

	Bike	Life	Palace	Robot	Space	Steam	Toad	Wine	Avg.
TensoRF	0.010	0.048	0.022	0.010	0.020	0.017	0.031	0.051	0.026
Tri-MipRF	0.012	0.048	0.023	0.019	0.019	-	0.036	0.055	0.030
NeuRBF	0.006	0.036	0.016	0.009	0.011	0.011	0.025	0.036	0.019
3D-GS	0.005	0.028	0.017	0.006	0.009	0.007	0.018	0.025	0.015
Scaffold-GS	0.007	0.030	0.019	0.008	0.019	0.010	0.022	0.021	0.017
GS-Shader	0.007	0.051	0.020	0.008	0.016	0.010	0.023	0.029	0.020
Ours-w anchor	0.005	0.027	0.018	0.007	0.007	0.008	0.021	0.025	0.015
Ours	0.004	0.023	0.015	0.005	0.007	0.007	0.017	0.021	0.013

Table 10. Per-scene LPIPS (VGG) comparison on the NSVF dataset.

	Teapot	Plane	Record	Ashtray	Dishes	Headphone	Jupyter	Lock	Avg.
3D-GS	27.24	26.80	43.81	34.43	29.62	38.72	40.52	29.36	33.81
Scaffold-GS	30.64	29.14	47.79	35.66	32.12	37.19	40.04	30.13	35.34
Ours-w anchor	33.53	31.56	50.35	36.14	32.95	38.48	40.10	30.96	36.76
Ours	34.88	30.83	50.51	37.02	32.90	39.45	41.18	31.46	37.28

Table 11. Per-scene PSNR comparison on our "Anisotropic Synthetic" dataset.

	Teapot	Plane	Record	Ashtray	Dishes	Headphone	Jupyter	Lock	Avg.
3D-GS	0.968	0.946	0.994	0.969	0.947	0.989	0.985	0.932	0.966
Scaffold-GS	0.979	0.965	0.998	0.973	0.967	0.986	0.983	0.924	0.972
Ours-w anchor	0.985	0.973	0.999	0.974	0.973	0.988	0.984	0.930	0.976
Ours	0.987	0.967	0.998	0.977	0.967	0.990	0.986	0.943	0.977

Table 12. Per-scene SSIM comparison on our "Anisotropic Synthetic" dataset.

	Teapot	Plane	Record	Ashtray	Dishes	Headphone	Jupyter	Lock	Avg.
3D-GS	0.043	0.085	0.019	0.044	0.120	0.015	0.075	0.098	0.062
Scaffold-GS	0.029	0.057	0.006	0.038	0.082	0.021	0.086	0.099	0.052
Ours-w anchor	0.022	0.042	0.004	0.039	0.067	0.017	0.084	0.093	0.046
Ours	0.021	0.051	0.009	0.036	0.087	0.014	0.071	0.087	0.047

Table 13. Per-scene LPIPS (VGG) comparison on our "Anisotropic Synthetic" dataset.

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
Plenoxels	21.91	20.10	23.49	20.66	22.25	27.59	23.62	23.42	24.67
iNGP	22.17	20.65	25.07	23.47	22.37	29.69	26.69	29.48	30.69
Mip-NeRF360	24.37	21.73	26.98	26.40	22.87	31.63	29.55	32.23	33.46
3D-GS	25.08	21.41	27.26	26.62	22.68	31.54	29.04	31.44	32.16
Scaffold-GS	25.05	21.20	27.33	26.49	23.23	32.13	29.44	31.59	32.49
Ours-w/o anchor	25.11	21.31	27.48	26.59	22.63	31.84	30.05	31.91	33.38
Ours	25.12	21.63	27.50	26.61	23.19	32.14	30.11	32.10	33.68

Table 14. Per-scene PSNR comparison on the Mip-NeRF 360 dataset.

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
Plenoxels	0.496	0.431	0.606	0.523	0.509	0.842	0.759	0.648	0.814
iNGP	0.512	0.486	0.701	0.594	0.542	0.871	0.817	0.858	0.906
Mip-NeRF360	0.685	0.583	0.813	0.744	0.632	0.913	0.894	0.920	0.941
3D-GS	0.746	0.588	0.855	0.769	0.635	0.924	0.913	0.931	0.944
Scaffold-GS	0.738	0.568	0.846	0.754	0.641	0.927	0.914	0.929	0.946
Ours-w/o anchor	0.739	0.584	0.856	0.759	0.631	0.925	0.919	0.933	0.946
Ours	0.744	0.589	0.850	0.758	0.640	0.929	0.917	0.930	0.950

Table 15. SSIM Comparison on the Mip-NeRF 360 dataset.

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
Plenoxels	0.506	0.521	0.386	0.503	0.540	0.419	0.441	0.447	0.398
iNGP	0.446	0.441	0.257	0.421	0.450	0.261	0.306	0.195	0.205
Mip-NeRF360	0.301	0.344	0.170	0.261	0.339	0.211	0.204	0.127	0.176
3D-GS	0.245	0.359	0.123	0.242	0.347	0.199	0.184	0.117	0.182
Scaffold-GS	0.266	0.383	0.143	0.276	0.353	0.200	0.195	0.121	0.186
Ours-w/o anchor	0.247	0.361	0.121	0.246	0.349	0.203	0.180	0.115	0.184
Ours	0.251	0.346	0.137	0.258	0.341	0.192	0.184	0.120	0.174

Table 16. LPIPS Comparison on the Mip-NeRF 360 dataset.

Method	Truck			Train			Dr Johnson			Playroom		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Plenoxels	23.22	0.774	0.335	18.93	0.663	0.422	23.14	0.787	0.521	22.98	0.802	0.499
iNGP	23.38	0.800	0.249	20.46	0.689	0.360	28.26	0.854	0.352	21.67	0.779	0.428
Mip-NeRF360	24.91	0.857	0.159	19.52	0.660	0.354	29.14	0.901	0.237	29.66	0.900	0.252
3D-GS	25.42	0.878	0.147	22.01	0.811	0.209	29.21	0.900	0.247	30.09	0.898	0.247
Scaffold-GS	25.77	0.883	0.147	22.15	0.822	0.206	29.80	0.907	0.250	30.62	0.904	0.258
Ours-w/o anchor	25.50	0.878	0.150	22.38	0.813	0.211	29.24	0.902	0.252	30.17	0.900	0.246
Ours	26.25	0.885	0.144	22.90	0.825	0.204	29.89	0.906	0.251	31.00	0.906	0.253

Table 17. Quantitative comparison on Tanks&Temples and Deep Blending dataset.