



安徽大学
人工智能学院
School of Artificial Intelligence
Anhui University

《专业英语》课程作业

实时辐射场渲染的 3D 高斯泼溅

学 院: 人工智能

专 业: 人工智能

班 级: 二班

姓 名: 杨跃浙

学 号: WA2214014

指导老师: 王辉

提交日期: 24.12

目 录

1 背景介绍	3
2 摘要	3
3 引言	4
4 相关工作	6
4.1 传统场景重建与渲染	6
4.2 神经渲染和辐射场	7
4.3 基于点云的渲染和辐射场	7
5 概述	8
6 可微 3D 高斯泼溅	9
7 三维高斯自适应密度优化	10
7.1 优化	10
7.2 自适应控制高斯	11
8 高斯快速可微光栅化器	13
9 实现、结果和评估	14
9.1 实现	14
9.2 结果与评估	15
9.3 消融	17
9.4 局限性	19
10 讨论与结论	20
11 致谢	21
12 附录	21
12.1 梯度计算细节	21
12.2 优化与致密化算法	24
12.3 光栅化器细节	24
12.4 数值稳定性	25
12.5 逐场景误差指标	25
13 最后总结	28

1 背景介绍

《3D Gaussian Splatting for Real-Time Radiance Field Rendering》¹ 由 Bernhard Kerbl、Georgios Kopanas、Thomas Leimkühler 和 George Drettakis 撰写，发表于 2023 年 7 月的《ACM Transactions on Graphics》（第 42 卷，第 4 期），并在同年 8 月发布于 arXiv 平台^{kerbl20233d}。这篇论文提出了一种名为 3D 高斯点绘（3D Gaussian Splatting, 3DGS）的创新方法，用于实时渲染辐射场。这一方法凭借其实时性、高效率和简洁的设计理念，在计算机视觉领域引起了广泛关注。

3DGS 从相机校准生成的稀疏点云入手，采用 3D 高斯函数对场景进行显式表示。这种表示方法不仅保留了连续体积辐射场的优点，还避免了对空白区域的无效计算。通过交替优化 3D 高斯的分布密度和各向异性协方差，3DGS 实现了对场景的精确建模。此外，作者还开发了一种支持各向异性点绘的快速可见性感知渲染算法，使得 3DGS 能够在 1080p 分辨率下实现每秒 30 帧或更高的实时渲染。凭借这一显著的性能提升，3DGS 在多个数据集上的实验结果均显示出其在视觉质量和渲染速度方面的领先优势²。

3DGS 与传统的神经辐射场方法（如 NeRF）形成鲜明对比。NeRF 自 2020 年提出以来，凭借神经网络实现空间坐标到颜色和密度的映射，在多视角合成领域取得了重大突破。然而，NeRF 计算密集，通常需要长时间的训练和大量的渲染资源，尤其是在高分辨率输出的情况下。3DGS 以其显式表示和高度并行化的工作流程，有效解决了这些痛点。通过可学习的 3D 高斯函数表示场景，它保留了 NeRF 的高质量图像合成能力，同时避免了空白空间渲染的开销，使实时渲染成为可能^{chen2024survey}。

3DGS 的设计还体现了对经典计算机视觉方法的回归与创新。其思路受启发于 2001 年的“Surface Splatting”，没有采用复杂的神经网络或大型模型，而是以一种简单、明了的方式解决了问题。在当下人工智能领域，充满了基于数十亿参数的模型研究，这种方法以其独特的简洁性显得格外突出³。3DGS 不仅让人耳目一新，也再次证明了传统方法与现代计算技术的融合能够激发出强大的解决方案。正因如此，它在不到一年内就达到了超过 2500 次引用，成为计算机视觉领域的一项重要成果。

文章中 **□** 部分为我自己对文章的理解和解释，不是原文部分。

2 摘要

通过多张照片或视频捕捉的场景进行新视角合成的任务，辐射场方法已引发重大变革。然而，要实现高质量的视觉感知图像，依旧需要借助训练和渲染成本高昂的神经网络，并且近期出现的速度较快的方法往往在质量上有所妥协。对于无界且完整的场景（并非孤立物体），在 1080p 分辨率下进行渲染时，当前尚无方法能够实现实时显示。在此背景下，我们引入了三个关键要素，这使得我们在保持具有竞争力的训练时间的同时，能够达成最先进的视觉质量，尤为重要的是，能够在 1080p 分辨率下实现高质量的实时（ $\geq 30 \text{ fps}$ ）新视角合成。

首先，我们以相机校准过程中产生的稀疏点为起点，【相机校准是一种估计相机镜头和图像传感器参数的技术。校准的目的是了解相机捕捉的图像如何对应现实世界的尺寸。“稀疏点”指的是在进行相机校准时用到的一些特定点，这些点可以是自然环境中的特征点或者特别设计的标定图案上的点。这些点在相机视角中的分布通常是稀疏的，而不是均匀覆盖整个视场。相机校准通过这些稀疏点来估计相机的内部参数（如焦距、镜头畸变）和外部参数（如相机位置和朝向）。这些参数对于后续的图形渲染是非常关键的。】运用 3D 高斯来表示场景。在场景优化过程中，3D 高斯保留了连续体辐射场的理想特性，同时避免了在空白区域进行不必要的计算。

其次，我们对 3D 高斯执行交错优化和密度控制操作，尤其针对各向异性协方差进行优化，以此实现对场景的精确表示。

最后，我们开发了一种快速的、具有可见性感知能力的渲染算法，该算法支持各向异性的绘制操作，既能加速训练过程，又能实现实时渲染。我们在多个已有的数据集上展示了顶尖的视觉质量和实时渲染效果。

关键词：新视角合成；辐射场；3D 高斯；实时渲染

3 引言

在 3D 场景表示领域，网格和点云是常见的形式，因其具有直观明确的特性，与基于 GPU/CUDA 的快速光栅化技术相得益彰。然而，近期兴起的神经辐射场（NeRF）方法却独树一帜，它基于连续场景表征构建，通常利用体光线行进技术优化多层感知器（MLP），以此实现采集场景的新视角合成。同样地，当下许多高效的辐射场解决方案也依赖连续表示，通过对体素、哈希网格或点云中存储的值进行插值运算来达成目的。尽管这些方法的连续性有助于优化，但渲染时所需的随机采样操作成本颇高，还容易引发噪声问题。【神经辐射场（Neural Radiance Fields, NeRF）是一种用于三维重建和渲染的先进技术，它通过深度学习来模拟复杂场景的光照和颜色。NeRF 在 2020 年被提出，迅速成为计算机视觉和图形学领域的热门话题。NeRF 使用一种称为场景表示的方法，它将一个场景编码为一个连续的神经网络。这个网络将给定的三维位置和观察方向作为输入，输出该位置和方向上的颜色和体积密度。这允许 NeRF 模拟光线在场景中的传播，包括光线的吸收、散射和反射等现象。^{4】}

为此，我们提出了一种创新方法，巧妙融合了两者的优势。在该方法中，3D 高斯表示既能保证优化达到顶尖的视觉质量，又能维持极具竞争力的训练时长。同时，我们基于瓦片的绘制方案能够确保在 1080p 分辨率下，针对多个已发布数据集实现实时渲染，且渲染质量达到一流水平。（见图1



图 1：我们的方法实现了辐射场的实时渲染，其质量可与之前质量最佳的方法相媲美。在优化时间上，与此前速度最快的方法相当。此性能的关键在于全新的 3D 高斯场景表示与实时可微渲染器相结合，这极大地加快了场景优化和新视角合成的速度。需要注意的是，在与同类方法相近的训练时间内，我们能达到相似质量，且经过 51 分钟训练后，可实现领先的质量，甚至比以往最优方法的质量还要略胜一筹。

我们的主要目标是实现多照片采集场景的实时渲染，并且在典型真实场景中，使创建场景表示的优化速度超越以往最有效的方法。当前，虽然一些新方法在训练速度上有所进步，但在视觉质量方面却难以企及当前最优的 NeRF 方法——Mip-NeRF360，其训练时长长达 48 小时。那些速度较快但质量欠佳的辐射场方法，虽能根据场景实现每秒 10-15 帧的交互渲染速率，却无法在高分辨率下达成实时渲染效果。

我们的解决方案主要包含三个关键部分：

- 首先，引入 3D 高斯作为一种极为灵活且富有表现力的场景表示手段。我们的初始输入与先前的 NeRF 类方法相同，即采用运动恢复结构 (SfM) 校准的相机，并利用 SfM 过程中生成的稀疏点云对 3D 高斯集进行初始化。相较于大多数依赖多视图立体 (MVS) 数据的基于点云的解决方案，我们仅用 SfM 点云作为输入就能获得高质量的结果。特别在处理 NeRF 合成数据集时，即便采用随机初始化，我们的方法也能输出高质量成果。经研究发现，3D 高斯不仅是一种可微的体表示形式，还能通过投影到二维平面，并应用标准的 α 混合技术，借助与 NeRF 等效的图像形成模型实现高效光栅化。【运动恢复结构 (Structure-from-Motion,SfM) 是一种从一系列带有视角变化的二维图像中恢复场景的三维结构的计算机视觉技术，它通过估计拍摄图像的相机位置和方向来重建三维点云。而多视图立体视觉 (Multi-View Stereo,MVS) 则使用已知的相机参数，通过图像间的视差来产生更高密度的三维点云或表面模型，通常在 SfM 的基础上进行，以提高三维重建的精度和细节。^{5】}
- 其次，对 3D 高斯的属性优化与自适应密度控制步骤相互交织。具体而言，我们对 3D 位置、不透明度 α 、各向异性协方差以及球谐 (spherical harmonic,SH) 系数等属性进行优化，在优化过程中适时添加或移除 3D 高斯。通过这一优化程序，最终生成的场景表示具有相当的紧凑性、非结构化特性和精确性，经测试，所有场景的高斯数量大致处于 100-500 万区间。【球谐函数是数学中的一类特殊函数，定义在单位球面上的一组正交函数。它们可以被视为二维傅里叶级数在球面坐标系中的等价物，用于表示球面上的任何平方可积函数。球谐函数通常表示为 $Y_l^m(\theta, \varphi)$ ，其中 l 是度 (degree)， m 是阶 (order)。球谐函数的形式为：

$$Y_l^m(\theta, \varphi) = K_l^m P_l^m(\cos \theta) e^{im\varphi}$$

其中：

- $K_l^m = \sqrt{\frac{(2l+1)}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}}$ 是一个归一化因子。
- $P_l^m(x)$ 是关联勒让德多项式，表示为 $P_l^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m}(P_l(x))$ ， $P_n(x) = \frac{1}{2^n \cdot n!} \frac{d^n}{dx^n} [(x^2 - 1)^n]$ 。
- $e^{im\varphi}$ 表示复指数函数，提供了方位角 φ 的周期性。

球谐函数具有以下特性：

- 正交性：不同 l 或 m 值的球谐函数之间是正交的。
- 完备性：所有可能的球谐函数构成一个完备集。
- 对称性：具有不同的对称性质，取决于它们的 m 和 l 值。

^{6】}

- 最后，我们构建了实时渲染解决方案，它借助快速 GPU 排序算法，并从近期基于瓦片的光栅化研究成果中获取灵感。得益于 3D 高斯表示形式，我们能够执行各向异性绘制操作，依据排序和 α 混合确保可见性顺序，并通过追踪尽可能多的已排序绘制的遍历过程，实现快速且精确的反向传播。【瓦片 (Tile) 瓦片技术是一种将大型图像或纹理分解成较小块（称为“瓦片”）的处理方法，每块可以单独处理和管理。这种技术主要用于优化图像处理，减少对内存的需求，并提高渲染效率。在实际应用中，瓦片可以并行处理，每个瓦片可以在不同的处理器

核心或 GPU 上独立渲染，最终将所有瓦片组合成完整的图像。瓦片的应用非常广泛，例如在地图应用中，大尺度的地图会被切分成许多小瓦片，用户在缩放或移动地图时只会加载视图中需要显示的瓦片。这种方法减少了数据传输和处理的负担，提高了应用的响应速度和效率。光栅化（Rasterization）光栅化是将矢量图形（如三维模型的线框）转换为像素阵列（光栅图像）的过程。在三维图形渲染中，这一步骤至关重要，因为它负责将 3D 模型的几何信息转换成可以在二维屏幕上显示的像素。光栅化过程包括确定哪些像素代表物体的表面，以及如何应用纹理和光照效果。这里我想提及，NeRF 使用的是光追，3DGS 使用的是光栅化，有学者认为之所以 3DGS 会比 NeRF 快这么多，就是因为这一点的改进，所以我想单独介绍一下这两个。光栅化和光追在原理上的区别主要体现在它们处理光线和场景交互的方式。光栅化通过将 3D 模型转换为 2D 屏幕坐标并直接在像素层面进行颜色和纹理计算，快速有效地生成图像，适合实时应用。相比之下，光追通过模拟光线如何在场景中传播，包括反射、折射和阴影效果，能够生成更为真实的视觉效果。光追的这种方法虽然能够提供高度的真实性，但由于其对光线路程的复杂计算，通常速度较慢，计算成本高，不常用于需要快速渲染的实时应用。这些原理上的差异导致光栅化和光追各自在特定领域内更为适用。】

综上所述，我们的主要贡献如下：

- 成功引入各向异性 3D 高斯，将其作为辐射场的高质量、非结构化表示形式。
- 提出 3D 高斯属性的优化方法，并与自适应密度控制有机结合，为捕获场景创建高质量表示。
- 研发针对 GPU 的快速可微渲染方法，该方法具备视觉感知能力，可实现各向异性绘制与快速反向传播，进而达成高质量的新视角合成效果。

在先前发布的数据集上进行的实验结果有力地表明，我们能够基于多视角捕获数据对 3D 高斯进行优化，其质量与之前最优的隐式辐射场方法相当，甚至在某些情况下更胜一筹。同时，我们在训练速度和质量方面与最快的方法相近，更为重要的是，首次实现了新视角合成的高质量实时渲染。

4 相关工作

我们首先对传统重建方法予以简要介绍，接着探讨基于点云的渲染及辐射场相关工作，并分析它们之间的相似之处。鉴于辐射场领域极为宽泛，在此我们仅聚焦于直接相关的研究内容。若要全面了解该领域，请查阅近期的出色综述文献。

4.1 传统场景重建与渲染

最早的新视角合成途径基于光场技术，先是进行密集采样，随后实现非结构化捕获。运动恢复结构的出现开辟了全新的局面，它能够利用一组照片合成新视角。【光场技术通过捕捉场景中光线的完整路径信息，使得在捕获之后可以生成新的视角图像。光场相机记录的是光线的方向和位置，这允许软件重建光在三维空间中的行为，从而创建出从不同角度观察场景的视图。这种技术主要利用了光场数据的深度信息，通过计算预测光线从新视角发散时的相互作用，从而合成出全新的图像。】运动恢复结构在相机标定期间估算出稀疏点云，最初用于 3D 空间的简单可视化。此后，多视图立体技术历经多年发展，催生出令人瞩目的全 3D 重建算法，进而推动了一系列视图合成算法的进步。

所有这些传统方法均是将输入图像重投影并混合至新视角相机内，同时借助几何信息引导重投影过程。在许多情形下，它们能取得优异成果，但通常难以妥善处理未重建区域或多视图立体生成

不存在几何结构时的“过度重建”问题。近期的神经渲染算法在很大程度上减少了此类瑕疵，并避免了在 GPU 上存储全部输入图像所产生的高额成本，在诸多方面超越了传统方法。

4.2 神经渲染和辐射场

深度学习技术很早就被应用于新视角合成领域。例如，曾有研究利用卷积神经网络（Convolutional Neural Network, CNN）估计混合权重或求解纹理空间问题，但使用基于 MVS 的几何方法是这些早期研究的主要缺陷，而且利用 CNN 进行最终渲染常常引发时域频闪现象。

Soft3D 率先提出了新视角合成的体积表示方法，随后深度学习技术与体积光线行进相结合，基于连续可微密度场来表征几何结构。不过，使用体积光线步进来渲染的成本颇高，因为查询体积需要大量样本。

神经辐射场引入了重要性采样和位置编码技术提升质量，但因其使用大型多层感知器（Multi-layer Perceptron, MLP）而导致速度受限。NeRF 的成功引发了大量后续研究，这些研究通常借助正则化策略来解决质量和速度问题。当前，新视角合成图像质量方面的最先进方法是 Mip-NeRF360，其渲染质量虽高，但训练和渲染耗时极长。与之相比，我们的方法能够在实现快速训练和实时渲染的同时，达到甚至超越其质量水平。

最新的方法主要致力于加快训练和/或渲染速度，主要通过以下三种设计思路：利用空间数据结构存储神经特征，并在体积光线行进期间进行插值；采用不同的编码方式；调整 MLP 规模。这些方法涵盖了空间离散化的多种变体、码本以及哈希表等编码形式，甚至包括改用较小的 MLP 或完全摒弃神经网络的方法。

其中，InstantNGP 使用哈希网格和占据网格加速计算，并采用小容量 MLP 表征密度和外观；Plenoxels 则利用稀疏体素网格对连续密度场进行插值，从而完全避开了神经网络。这两种方法都依赖于球面谐波，前者直接表示方向效应，后者将其输入编码到颜色网络。尽管它们都取得了出色的结果，但在有效表示空区域方面仍存在困难，且图像质量在很大程度上受限于加速所用的结构化网格，同时渲染速度也因在光线行进步骤中需要查询大量样本而受到影响。而我们采用的非结构化、对 GPU 友好的 3D 高斯方法，无需神经网络即可实现更快的渲染速度和更高的质量。

4.3 基于点云的渲染和辐射场

基于点云的方法能够有效渲染不连续且非结构化的几何样本。最简单的点样本渲染方式是将一组固定大小的非结构化点进行光栅化，在此过程中可利用图形 API 支持的点类型或 GPU 并行光栅化技术。但这种方法存在明显缺陷，会产生孔洞、走样问题且严格不连续。**【在计算机图形学中，基于点云的渲染是一种使用三维点集来生成图像的方法，这些点包括位置以及可能的颜色和法线信息，适用于渲染复杂或详细的场景。】**

后续高质量的基于点云渲染研究通过对范围大于像素的点基元（如圆盘、椭圆盘、椭圆体或面元等）进行泼溅 (Splatting) 操作，有效解决了这些问题。**【点基元 (Point Primitives)，是在点云渲染中用来表示三维空间中单个点的几何形状⁶。每种点基元有其独特的特点，适用于不同的渲染需求：**

1. 圆盘 (Disks)：这些在屏幕上表现为二维圆形的点基元，用于模拟表面散射效果。圆盘是最简单的点基元形状，易于计算和渲染，适合模拟平滑或轻微曲面的物体。
2. 椭圆盘 (Elliptical Disks)：类似于圆盘，但是它们在不同的轴向有不同的半径。这使得椭圆盘可以更好地适应于局部点密度不均或在特定视角下的点云数据，提供更自然的覆盖效果。
3. 椭圆体 (Ellipsoids)：三维空间中的点基元，具有三个不同的轴，可以更精确地模拟点的空间扩展和深度，适合复杂和详细的结构表现。
- 4.

面元 (Surface Elements)：通常是小面片或多边形，可以更精确地覆盖和模拟复杂的表面。面元因其能精确匹配局部表面的形状而被广泛用于高精度的三维重建和渲染。

泼溅 (Splatting) 是一种点云渲染技术，其中点基元被扩展到覆盖多个像素的大小，以填补点云的视觉空隙和提高渲染的连续性和真实感。通过泼溅操作，点基元不仅覆盖更大的屏幕区域，而且在它们重叠的地方可以通过颜色和光照的混合来实现更平滑的视觉效果。这种方法特别适用于处理大规模或细节密集的点云数据，有效提升了视觉质量和场景的真实感。¹ 近期，可微点云渲染技术引发了关注。通过为点添加神经特征并利用 CNN 进行渲染，实现了快速甚至实时的视角合成。然而，这些方法仍依赖 MVS 获取初始几何形状，因此会继承其相关缺陷，在无特征、闪亮区域或薄结构等复杂情况下易出现过重建或欠重建问题。

基于点云的 α -混合与 NeRF 风格的体渲染本质上采用相同的成像模型。具体而言，颜色 C 通过沿射线的体渲染得出：

$$C = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) c_i, \quad T_i = \exp \left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right), \quad (1)$$

该式可改写为：

$$C = \sum_{i=1}^N T_i \alpha_i c_i, \quad \alpha_i = 1 - \exp(-\sigma_i \delta_i), \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (2)$$

一种典型的基于神经点云的方法计算像素颜色 C 的方式为：

$$C = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (3)$$

其中 c_i 是每个点的颜色， α_i 是通过评估协方差为 Σ 的二维高斯并乘以每个点的学习不透明度得出。

从式 (1) 和式 (3) 可以看出，两种成像模型相同，但渲染算法差异显著。NeRF 是一种连续表示，隐含地表示空/占用空间，需要昂贵的随机采样来获取样本，从而产生噪声和高计算开销。相比之下，点云是一种非结构化离散表示，具有足够的灵活性，可通过优化不透明度和位置来创建、破坏和移动几何结构，避免了全体积表示的缺点。

Pulsar 实现的快速球体光栅化启发了我们的基于瓦片和排序的渲染器。我们希望在排序的高斯椭球上保持（近似）传统的 α -混合，以保留体积表示的优势，即我们的光栅化遵循可见性顺序，而不像某些方法与顺序无关。此外，我们在像素中的所有高斯椭球上反向传播梯度，并光栅化各向异性高斯椭球，这些都有助于提高结果的视觉质量。尽管先前一些方法使用 CNN 进行渲染会导致时域不稳定，但 Pulsar 和 ADOP 的渲染速度促使我们开发快速渲染解决方案。

还有一种近期方法使用点和径向基函数表示辐射场，在优化过程中采用点修剪和密集化技术，但因使用体积光线行进而无法实现实时显示速率。

5 概述

我们的方法以一组静态场景的图像和经运动恢复结构校准的对应相机为输入，该过程会生成一个稀疏点云。依据这些点，我们构建了一组 3D 高斯（在第6章详述），其由位置（均值）、协方差矩阵和不透明度 α 来定义，这种定义方式使得优化机制十分灵活，进而能形成对 3D 场景较为紧凑的表示。这得益于高度各向异性的体素块可紧凑地呈现精细结构。对于辐射场的方向外观分量（即颜色），我们按照常规做法，采用球谐函数予以表示。

我们的算法通过一系列针对 3D 高斯参数的优化步骤来创建辐射场表示（见第7章），这些参数包含位置、协方差、 α 以及 SH 系数等，并且在优化进程中穿插着对高斯密度的自适应控制操作。我们方法高效的关键在于基于瓦片的光栅化器（见第8章），它允许对各向异性斑点进行 α 混合操作，并通过快速排序保证遵循可见性顺序。此快速光栅化器还能通过跟踪累积的 α 值实现快速反向传播，且对接收梯度的高斯数量没有限制。

我们的方法概览如图2所示。

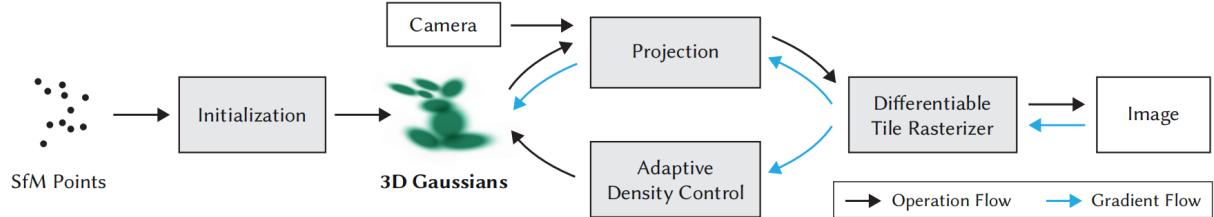


图 2：优化流程起始于稀疏的 SfM 点云，基于此创建一组 3D 高斯点云。随后，对这组高斯的密度展开优化与自适应调控操作。在整个优化进程中，运用基于瓦片的快速渲染器，相较于当前最优的快速辐射场方法，其训练时长更具优势。一旦完成训练，我们的渲染器便能针对各类场景实现实时的视角导航与观测。

6 可微 3D 高斯泼溅

我们的目标是优化一种场景表示方式，以便从没有法线信息的稀疏 SfM 点集出发，实现高质量的新视图合成。为此，我们需要一种基元，它既能继承可微体积表示的特性，又具有非结构化和明确的特点，从而实现快速渲染。我们选定了 3D 高斯，它具有可微性，能够轻易地投影到二维斑块上，进而实现快速的 α 混合渲染。

我们的表示方法与之前采用二维点的方法有相似之处，那些方法假定每个点都是带有法线的平面小圆。然而，由于 SfM 点极其稀疏，很难估算法线，并且从这种估计中优化噪声很大的法线极具挑战性。因此，我们将几何建模为一组无需法线的 3D 高斯。我们的高斯由以点（均值） μ 为中心的全三维协方差矩阵 Σ 定义：

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}, \quad (4)$$

在混合过程中，这个高斯乘以 $a\alpha$ 。

为了进行渲染，我们需要将 3D 高斯投影到二维空间。Zwicker 等人展示了如何将其投影到图像空间。给定视图变换 W ，相机坐标中的协方差矩阵 Σ' 如下所示：

$$\Sigma' = JW\Sigma W^T J^T, \quad (5)$$

其中 J 是投影变换仿射近似的雅可比。Zwicker 等人还表明，如果跳过 Σ' 的第三行和第三列，就会得到一个 2×2 方差矩阵，其结构和性质与之前从有法线的平面点出发的工作中得到的矩阵相同。

一种直接的方法是直接优化协方差矩阵 Σ ，以获得代表辐射场的 3D 高斯。但协方差矩阵只有在半正定时才有物理意义。在对所有参数进行优化时，我们采用梯度下降法，然而这种方法难以约束产生有效矩阵，并且更新步骤和梯度很容易产生无效协方差矩阵。

因此，我们选择了一种更直观但同样有效的优化表示方法。3D 高斯的协方差矩阵 Σ 类似于描述一个椭球体的构型。给定缩放矩阵 S 和旋转矩阵 R ，可以找到相应的 Σ :

$$\Sigma = RSS^T R^T. \quad (6)$$

为了对这两个因素进行独立优化，我们将它们分开存储：用一个三维向量 s 表示缩放，用一个四元数 q 表示旋转。可以将它们转换为各自的矩阵并组合，同时确保对 q 进行归一化，以获得有效的单位四元数。

为避免在训练过程中因自动微分产生大量开销，我们明确推导出所有参数的梯度。精确导数计算的详情见附录 12.1。

这种各向异性协方差的表示方法适合优化，使我们能够优化 3D 高斯以适应拍摄场景中不同形状的几何图形，从而获得相当紧凑的表示。图 3 展示了这种情况。



图 3: 展示了我们对优化后的 3D 高斯进行可视化的结果，具体是将其缩小 60%（位于最右侧）。从中能够清晰地看出 3D 高斯的各向异性形态，经过优化后，它可以紧凑地呈现复杂的几何结构。而实际渲染的图像则显示在左侧。

7 三维高斯自适应密度优化

我们方法的核心在于优化步骤，其目的是创建一组密集的三维高斯，以便精准地表示用于自由视角合成的场景。除了位置 P 、不透明度 α 以及协方差 Σ 之外，我们还对代表每个高斯颜色 c 的球谐系数进行优化，从而准确捕捉场景随视角变化的外观。这些参数的优化过程与控制高斯密度的步骤相互交错，以此更好地呈现场景。

7.1 优化

优化过程基于连续的迭代渲染，并将生成的图像与捕获数据集中的训练视图进行对比。由于三维到二维投影存在模糊性，几何体的位置很可能会出现错误。所以，我们的优化需要能够创建几何图形，并且在几何图形位置错误时能够将其销毁或移动。三维高斯协方差参数的质量对于表示的紧凑性至关重要，因为少量的大型各向异性高斯就能捕捉大面积的同质区域。

我们运用随机梯度下降技术进行优化，充分利用标准的 GPU 加速框架，并且按照近期的最佳实践，为某些操作添加定制的 CUDA 计算核。特别地，我们的快速光栅化（见第??章）在优化效率方面极为关键，因为它是优化的主要计算瓶颈。【随机梯度下降 (Stochastic Gradient Descent, SGD) 用于优化 3D 图形场景渲染的过程，尤其是在光栅化阶段。这里，利用了 GPU 的加速能力和定制的 CUDA 计算核心来提高计算效率。这种方法的创新之处在于它结合了传统图形处理技术与现代的并行计算框架，针对性地解决了光栅化这一计算瓶颈问题。】

我和某大学的老师及同学在讨论时，他们指出，虽然 3DGS 技术本身可能主要是现有技术的应用和结合，但其创新在于如何将这些技术融合在一起，以适应当前的硬件能力和市场需求。这种结合并优化传统技术的方法，尽管在学术上可能看起来不是完全新颖的，但在工业应用中却极具价值，特别是在解决实时处理和端到端系统优化这两大工业界难题时显得尤为重要。】我们使用 sigmoid 激活函数将 α 限制在 $[0, 1)$ 区间内，以获得平滑的梯度；出于类似的原因，对于协方差的尺度，我们采用指数激活函数。

初始协方差矩阵估计为各向同性高斯矩阵，其轴等于到最近三个点的距离的平均值。我们采用标准的指数衰减调度技术，但仅针对位置。损失函数为：

$$L = (1 - \lambda)L_1 + \lambda L_{D-SSIM}, \quad (7)$$

其中 $\lambda = 0.2$ ，在所有测试中均保持该值。关于学习调度及其他元素的详细信息将在第??节介绍。

7.2 自适应控制高斯

我们从运动结构恢复生成的初始稀疏点集开始，接着应用我们的方法自适应地控制单位体积内的高斯数量和密度。这样我们就能从初始的稀疏高斯集转变为能更好地表示场景且参数正确的更密集的高斯集。在优化预热（见第??节）之后，我们每迭代 100 次就进行一次高斯致密化，并移除那些基本透明的高斯，即 α 小于阈值 ϵ_α 的高斯。

我们对高斯的自适应控制需要填充空白区域，重点关注几何特征缺失的区域（“重建不足”）以及高斯覆盖场景大面积的区域（通常对应“重建过度”）。我们观察到，这两种情况都有较大的视空间位置梯度。直观来说，这可能是因为它们对应的区域还没有很好地重构，而优化会尝试移动高斯来纠正。

由于这两种情况都适合进行致密化，我们会对视图空间位置梯度的平均值高于阈值 τ_{pos} 的高斯进行致密化，在测试中，我们把这个阈值设为 0.0002。下面详细介绍这个过程，如图4所示：

对于处于未完全重建区域的小高斯，我们需要覆盖必须创建的新几何体。为此，最好是克隆高斯，即创建一个相同大小的副本，并沿位置梯度方向移动它。

另一方面，高方差区域的大高斯需要分割成小高斯。我们用两个新的高斯替换这些高斯，并按照实验确定的系数 $\phi = 1.6$ 来划分它们的比例。我们还使用原始三维高斯作为采样的概率密度函数 (Probability Density Function, PDF) 来初始化它们的位置。【使用高斯概率密度函数意味着点的初始位置将根据高斯分布来随机分布，这种分布以其均值为中心，展现出特定的方差，从而在空间中形成一种自然的聚集趋势。这种方法可以确保点的初始分布既有随机性也有一定的统计规律性，便于后续的处理和优化。选择高斯分布作为初始化方法是因为它的数学属性良好，易于计算且广泛应用于需要考虑空间位置或自然聚集特性的场景。】在第一种情况下，我们会检测并处理增加系统总体积和高斯数量的需求；而在第二种情况下，我们会保留总体积，但增加高斯数量。与其他体积表示法类似，我们的优化可能会陷入浮点靠近输入摄像头的情况；在我们的案例中，这可能会导致高斯密度不合理地增加。缓和高斯数量增加的有效方法是，每 $N = 3000$ 次迭代后，将 α 数值设置

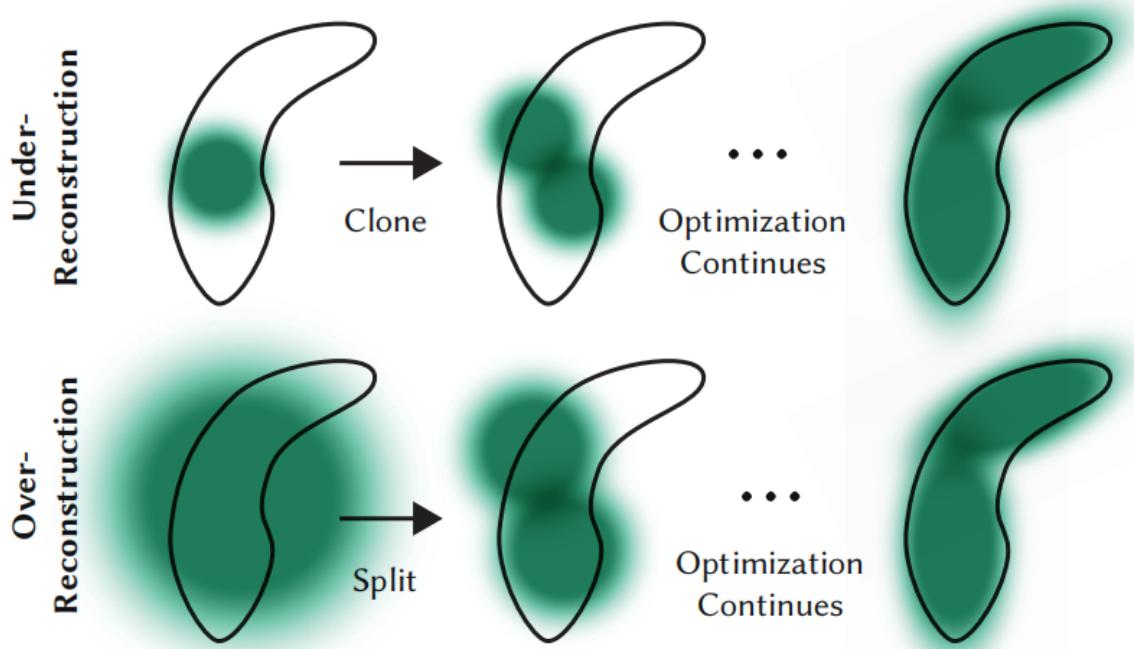


图 4: 我们的自适应高斯致密化方案。顶行 (重建不足): 当小尺度几何图形 (黑色轮廓) 覆盖不足时, 我们会克隆相应的高斯。下一行 (过度重构): 如果小尺度几何图形由一个大拼接块表示, 我们将其一分为二。

为接近于零。优化后, 我们会在需要的地方增加高斯的 α , 同时允许我们的剔除方法去除 α 小于上述阈值的高斯。高斯可能会缩小或增大, 并与其他高斯有相当程度的重叠, 但我们会定期移除世界空间中非常大的高斯和视图空间中足迹较大的高斯。这种策略能很好地控制高斯总数。与其他方法不同, 我们的模型中的高斯始终保持在欧几里得空间中的原始状态, 不需要对遥远或大型高斯进行空间压缩、扭曲或投影等操作。【这章的贡献在于提出了一种基于三维高斯分布的自适应密度优化方法, 用以精准表示复杂场景, 实现自由视角下的图像合成。这种方法通过直接优化高斯分布的参数, 包括位置、颜色、不透明度以及协方差矩阵, 而非依赖深度学习模型, 从而实现了对场景外观变化的精确捕捉。文章特别强调了各向异性高斯分布在捕捉方向性和区域特征中的重要性, 通过动态调整高斯的密度和数量, 自适应地解决了重建不足和重建过度的问题。优化过程中结合了随机梯度下降 (SGD) 和定制化的 CUDA 计算核, 使得渲染和优化在 GPU 上能高效并行运行, 从而显著提高了效率。】

这种方法无需深度学习的原因主要在于高斯分布本身的参数化和可微性。通过直接优化高斯分布的几何参数和颜色参数, 它能够在三维空间和二维投影之间建立直接的联系, 并利用损失函数对投影结果进行精细对比。这种优化过程不依赖神经网络中的特征提取, 而是通过显式建模捕捉场景的几何和外观特性。此外, 结合光栅化技术和参数的自适应调整, 这种方法可以在无监督的情况下, 动态优化稀疏点集到致密高斯分布的转变。

这种方法的主要优势在于效率高、计算资源需求低且对数据依赖较小。它避免了深度学习模型需要的大量训练数据和长时间的训练过程, 直接通过几何优化实现场景重建。此外, 由于不使用深度网络, 其计算过程透明、易解释, 参数的物理意义明确, 便于调试和扩展。同时, 这种方法特别适合实时渲染任务, 通过结合现代 GPU 架构和自定义优化技术, 能够高效处理动态场景。

相比之下，深度学习的优势在于其强大的特征学习能力和非线性建模能力，能够捕捉场景中的细腻细节和复杂纹理，具有更好的泛化性能。深度学习擅长利用大规模数据集，通过训练获得丰富的先验知识，从而在复杂光照、材质变化等高维场景中表现出色。然而，与本方法相比，直接应用深度学习需要更多的计算资源和训练时间，且模型的解释性较低，对工业实时应用的适应性较差。为此，我们现在正在尝试，将一些全新的深度学习技术与 3DGS 技术结合，希望能在尽量不增加代价的前提下，增强三维高斯点的表征能力。^{7】}

8 高斯快速可微光栅化器

我们的目标是达成快速的整体渲染与快速排序效果，进而实现近似的 α 混合（涵盖各向异性泼溅），同时避免如先前工作中对可接收梯度的泼溅数量所设置的严格限制。

为达成这些目标，我们设计了一种基于瓦片的高斯泼溅光栅化器，其灵感源自近期的软件光栅化方法。此光栅化器能够一次性对整个图像的基元进行预排序，有效规避了先前 α 混合方案中按每个像素进行排序所产生的高昂开销。我们的快速光栅化器能够针对任意数量的混合高斯高效地执行反向传播，且额外内存消耗较低，仅需每个像素维持恒定的开销。我们的光栅化流水线具备完全可微的特性，并且鉴于其向二维的投影（见第 6 章），能够像之前的二维泼溅方法一样对各向异性泼溅进行光栅化处理。【各向异性在图形处理和渲染领域中指的是物体属性在不同方向上表现出不同的行为或特性。各向异性泼溅特别指的是点基元在屏幕上渲染时，其形状和扩散程度根据不同方向而异。

具体来说，如果在泼溅操作中应用各向异性，那么这些点基元在屏幕上的展现可能不是完全对称的圆形或者均匀分布的形状，而是根据其在三维空间中的方向、光照条件或视角差异，显示为椭圆形或其他非对称形状。这种方法允许更准确地模拟如光照影响下的反射和阴影效果，或是物体表面的纹理特性，从而在视觉上提供更为丰富和真实的渲染效果。

在光栅化处理中，各向异性泼溅的实现确保了渲染的点基元能够根据其在空间中的具体位置和方向属性进行适当的调整，优化视觉表现和性能，这在模拟复杂材质和照明条件下的视觉效果时尤为重要。这种技术的应用提升了渲染质量，使得数字图像更加逼真，更能符合人眼对现实世界的感知。自然界中的物体和场景通常在不同的方向上显示出不同的视觉特性和行为。这些差异可以是光照条件、表面纹理、或者物体自身的几何形态导致的。在处理这类图像时，各向异性的方法能够更精确地模拟和表达这些方向性的变化，从而增强图像的视觉真实感和深度感。通过各向异性的处理方法，RGB 自然图像的渲染能够更好地适应这些复杂多变的自然特征。但是这也就意味着，这一方法是不适用于各向同性的医学图像的⁸，我们正在尝试 3DGS 在医学图像/深度图上的应用。】我们的方法首先将屏幕划分为 16×16 的瓦片。接着，依据视图锥和每个瓦片对三维高斯进行剔除操作。具体而言，我们仅保留 99% 置信区间与视图锥相交的高斯。此外，我们运用保护带简单剔除处于极端位置的高斯（即均值接近近平面且远离视图锥的高斯），因为计算此类高斯的投影二维协方差会导致不稳定情况出现。随后，我们依据每个高斯所重叠的瓦片数量对其进行实例化，并为每个实例分配一个结合了视图空间深度和瓦片 ID 的键。之后，利用快速 GPU 基数排序，依据这些键对高斯进行排序。需要注意的是，在此过程中不存在额外的按像素对高斯进行排序的操作，混合操作基于初始排序结果开展。因此，在某些配置下，我们的 α 混合是近似的。然而，当泼溅接近单个像素大小时，这些近似所产生的影响可忽略不计。我们发现这种方式极大地提升了训练和渲染性能，并且在融合场景中不会产生明显的瑕疵。

在完成对高斯的排序后，我们针对每个瓦片生成一个列表，该列表标记出深度排序中首个和最

后一个会溅射到给定瓦片上的条目。在光栅化阶段，我们为每个瓦片启动一个线程块。每个线程块首先协同将高斯数据包加载至共享内存，然后针对给定像素，通过从前至后遍历列表来累积颜色和 α 值，以此最大程度地提升数据加载、共享及处理的并行性。当像素达到目标饱和度时，相应的线程便会停止。每隔一定时间，会对瓦片中的线程进行查询，当所有像素均达到饱和（即 α 达到1）时，整个瓦片的处理流程便会终止。关于排序的详细信息以及整个光栅化方法的高级概述可查阅附录12.3。

在光栅化过程中， α 饱和度是唯一的停止准则。与先前的工作有所不同，我们不限制接收梯度更新的混合基元数量。之所以强制执行这一特性，是为了使我们的方法能够处理具有任意且不同深度复杂度的场景，并对其进行精确学习，而无需针对特定场景进行超参数调整。因此，在反向传播过程中，我们必须恢复前向传播时每个像素的完整混合点序列。一种解决方案是在全局内存中存储每个像素任意长的混合点列表，但为避免由此产生的动态内存管理开销，我们选择再次遍历每个瓦片的列表，在此过程中可以重复利用前向传播中的高斯排序数组和瓦片范围。为便于梯度计算，我们采用从后向前的方式进行遍历。

遍历从影响瓦片中任何像素的最后一个点开始，并再次以协同方式将点加载至共享内存。此外，每个像素仅在其深度小于或等于前向传播过程中对其颜色产生影响的最后一个点的深度时，才会启动（昂贵的）重叠测试和点处理流程。第6章中所描述的梯度计算需要原始混合过程中每一步所累积的不透明度值。我们无需在后向传播中遍历一个逐渐缩小的不透明度显式列表，只需存储前向传播结束时的总累积不透明度，便能恢复这些中间不透明度。具体而言，每个点在前向过程中存储最终累积的不透明度 α ，在从后向前的遍历中，将其除以每个点的 α ，即可获取梯度计算所需的系数。

9 实现、结果和评估

接下来，我们将探讨实现过程中的一些细节、所取得的结果，以及对我们算法的评估情况，包括与先前工作的对比和消融研究。

9.1 实现

我们在Python中利用PyTorch框架实现了我们的方法，并编写了用于光栅化的自定义CUDA内核，这些内核是对先前方法的扩展。同时，我们使用英伟达CUB排序例程进行快速基数排序。此外，我们借助开源的SIBR构建了一个交互式查看器，用于测量帧率。源代码及所有数据可在以下网址获取：<https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>。

优化细节：为确保稳定性，我们在较低分辨率下进行计算“预热”。具体而言，开始优化时采用的图像分辨率是原始分辨率的四分之一，并在迭代250次和500次后分别进行两次上采样。**【预热是一种分阶段的训练策略。在深度学习和优化中，预热通常指在优化的初始阶段以更简单或低分辨率的任务开始，让模型逐步适应优化过程，从而避免初期的剧烈变化导致优化陷入局部最优或不稳定。这里在训练开始时使用较低的分辨率（原始分辨率的四分之一）进行优化，然后随着优化过程的进行逐步提高分辨率。】**球谐系数优化对角度信息的缺失较为敏感。对于典型的“类NeRF”捕捉，即中心物体被其周围整个半球拍摄的照片所观测时，优化效果良好。然而，若捕捉存在角度区域缺失的情况（例如捕捉场景角落或进行“内向外”捕捉时），优化可能会产生完全错误的SH零阶分量值（即基色或漫反射色）。为解决此问题，我们首先仅优化零阶成分，随后每迭代1000次引入一个SH波段，直至所有4个SH波段均得以体现。**【球谐函数的每个波段 l 包含 $2l+1$ 个独立的基函数，**

对应于 m 的取值范围 $-l$ 到 l 。这些基函数的每一个都对应一个可学习的参数。因此，对于每个波段 l ，可学习参数的数量是 $2l + 1^5$ 。对于球谐函数的每个波段 l :

- 波段 0: 仅有一个基函数（即 $m = 0$ ），对应 1 个可学习参数。
- 波段 1: 包含 3 个基函数 ($m = -1, 0, 1$)，对应 3 个可学习参数。
- 波段 2: 包含 5 个基函数 ($m = -2, -1, 0, 1, 2$)，对应 5 个可学习参数。
- 波段 3: 包含 7 个基函数 ($m = -3, -2, -1, 0, 1, 2, 3$)，对应 7 个可学习参数。

以此类推，每个波段的可学习参数数量为 $2l + 1$ 。

累计参数数量

如果考虑所有波段从 0 到 L ，总的可学习参数数量为:

$$\text{总参数数} = \sum_{l=0}^L (2l + 1)$$

例如:

- 当 $L = 0$ (仅零阶波段): 总参数数 = 1。
- 当 $L = 1$: 总参数数 = $1 + 3 = 4$ 。
- 当 $L = 2$: 总参数数 = $1 + 3 + 5 = 9$ 。
- 当 $L = 3$: 总参数数 = $1 + 3 + 5 + 7 = 16$ 。

在“类 NeRF”优化中，通过分阶段引入球谐波的不同波段，逐步增加模型复杂性，从零阶的基色优化到高阶的角度相关光照优化。这种策略能够避免因不完整的视角捕捉导致高阶成分学习过早、过度拟合的问题。

因此，每次引入一个波段时，增加的参数数量为该波段的 $2l + 1$ 。这里我的理解是，可以简单的理解为模型在学习（拟合）一个复杂的函数，随着迭代次数的上升学习的参数增加，拟合效果也进一步变好。这里其实 SH 只是一个选择，但是不是一定只有这个选择，说不定其他表示也能有很好的效果。】

9.2 结果与评估

结果：我们在先前发布的数据集以及合成 Blender 数据集中的共计 13 个真实场景上对算法进行了测试。特别地，我们在 Mip - NeRF360 所展示的全套场景（其代表了当前 NeRF 渲染质量的最高水平）、Tanks&Temples 数据集 (2017) 中的两个场景以及 Hedman 等人提供的两个场景上测试了我们的方法。这些场景的捕捉风格各异，涵盖有界室内场景和大型无界室外环境。在评估过程中，所有实验均采用相同的超参数配置。除 Mip - NeRF360 方法外，所有结果均在 A6000 GPU 上运行。【我真是去复现过 Mip-NeRF 的代码，巨慢。】在补充材料中，我们展示了一些场景的渲染视频路径，这些场景的视角与输入照片相差甚远。

真实世界场景：在质量方面，当前最先进的方法是 Mip - NeRF360。我们将该方法作为质量基准进行对比，并与两种最新的快速 NeRF 方法 InstantNGP 和 Plenoxels 进行比较。

我们按照 Mip - NeRF360 所建议的方法对数据集进行训练/测试划分，每隔 8 张照片选取一张用于测试，以此确保比较的一致性和有效性，进而生成误差指标。所采用的误差指标为文献中常用的标准 PSNR、L - PIPS 和 SSIM；具体数据见表 1。表中除 Mip - NeRF360 在其自身数据集上的结果直接引用自原始出版物（以避免与当前最优水平混淆）外，其余数字均来自我们自行运行的先前方法的作者代码。对于文中图像，我们使用了自己运行的 Mip - NeRF360 数据，这些数据的详细信息见附录 D。此外，我们还展示了平均训练时间、渲染速度以及用于存储优化参数的内存。我们报告了 InstantNGP 基本配置运行 35K 次迭代以及作者建议的稍大网络（Big）的结果，同时也给出了我们方法的两种配置（7K 和 30K 次迭代）的结果。两种配置在视觉质量上的差异如图 6 所示，在许多情况下，7K 次迭代时的质量已相当不错。【1. PSNR (Peak Signal-to-Noise Ratio) 定义：PSNR 是衡量图像重建质量的一种传统指标，用于比较重建图像与参考图像之间的差异。它基于像素级误差计算，反映图像重建中的信噪比。

计算公式：

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right)$$

其中：

- MAX_I 是图像像素的最大可能值（对于 8 位图像通常为 255）。
- MSE 是均方误差，计算如下：

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i, j) - K(i, j))^2$$

- I 和 K 分别是参考图像和重建图像的像素值。

特点：1. 数值越高，代表重建图像与参考图像越接近，质量越高。2. 对细节感知较差，只能反映像素级误差，难以反映人眼感知的视觉质量。

2. L-PIPS (Learned Perceptual Image Patch Similarity)

定义：L-PIPS 是基于深度学习的图像感知质量评价指标，专注于衡量图像的感知相似度。它通过比较图像在预训练神经网络（如 VGG）的特征空间中的差异，来评估图像的质量。

计算方式：

$$L\text{-PIPS}(I, K) = \sum \text{特征图差异}$$

其中特征图是从预训练模型（如 VGG16）提取的。

特点：1. 数值越低，代表图像的感知质量越高。2. 关注人类视觉系统的特性，能更好地捕捉细节、纹理和感知上的差异。3. 对几何失真和局部噪声敏感，适合复杂图像评估任务。

3. SSIM (Structural Similarity Index)

定义：

SSIM 是一种结构相似性评价指标，旨在模拟人眼对图像结构信息的感知能力。它同时考虑亮度、对比度和结构信息三个因素。

计算公式：

$$\text{SSIM}(I, K) = \frac{(2\mu_I\mu_K + c_1)(2\sigma_{IK} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)}$$

其中：

- μ_I, μ_K 是图像 I, K 的均值。

- σ_I^2, σ_K^2 是图像 I, K 的方差。
- σ_{IK} 是图像 I 和 K 的协方差。
- c_1, c_2 是防止分母为零的常数。

特点：1. 值域在 $[0, 1]$ ，越接近 1 表示图像质量越高。2. 同时考虑了图像的亮度、对比度和结构特征，能够更贴合人眼的视觉感知。

指标	定义	优点	局限性
PSNR	像素级信噪比	计算简单，直观	无法反映视觉感知质量
L-PIPS	基于深度学习的感知相似度	贴近人眼感知，关注纹理与细节	依赖预训练模型，计算较耗时
SSIM	结构相似性指标	考虑图像结构，对比度和亮度	对局部几何失真敏感

】

不同数据集的训练时间各不相同，我们将分别予以报告。需要注意的是，不同数据集的图像分辨率也存在差异。在项目网站上，我们提供了所有测试视图的渲染图，用于计算所有方法（我们的方法和先前工作）在所有场景下的统计数据，且所有渲染均保留原始输入分辨率。

从表中可以看出，我们完全收敛的模型在质量上与最先进的 (State Of The Art, SOTA) Mip - NeRF360 方法相当，有时甚至略胜一筹。在相同硬件条件下，Mip - NeRF360 的平均训练时间为 48 小时，而我们仅需 35 - 45 分钟，其渲染时间为 10 秒/帧。经过 5 - 10 分钟的训练，我们的方法可达到与 InstantNGP 和 Plenoxels 相当的质量，但额外的训练时间使我们能够获得 SOTA 质量，而其他快速方法则无法实现这一点。对于 Tanks&Temples 数据集，我们在相似的训练时间（约 7 分钟）内达到了与基本 InstantNGP 相似的质量。

我们在图 5 中展示了我们的方法与先前渲染方法在测试视图中的对比结果，我们的方法结果基于 30K 次迭代训练。可以看到，在某些情况下，即使是 Mip - NeRF360 也会出现我们的方法能够避免的残留瑕疵（例如，在自行车和树桩场景中植被的模糊，或房间场景中墙壁的模糊）。在补充视频和网页中，我们提供了远距离路径的比较，我们的方法倾向于保留覆盖良好区域的视觉细节，即便从很远的地方观察也是如此，而先前的方法并非总是如此。

合成边界场景：除了现实场景外，我们还在合成 Blender 数据集上对我们的方法进行了评估。这些场景提供了详尽的视图集，尺寸有限且具备精确的摄像机参数。在这种情况下，即使采用随机初始化，我们也能获得最先进的结果：从包围场景边界的体积内的 100K 个均匀随机高斯开始训练，我们的方法能够快速自动地将其修剪为大约 6 - 10K 个有意义的高斯。经过 30K 次迭代后，训练模型的最终规模达到每个场景约 200 - 500K 个高斯。为确保兼容性，我们在表 2 中使用白色背景对所取得的 PSNR 分数与先前方法进行了报告和比较，示例可参见图 12（左起第二张图片）和补充材料。训练后的合成场景能够以 180 - 300 FPS 的速度进行渲染。

表 1：将我们的方法与以往工作进行的定量评估，是基于三个数据集计算得出的。带有剑号 (\dagger) 标记的结果是直接从原始论文中引用的，其余所有结果均通过我们自己的实验获得。

Dataset Method Metric	Mip-NeRF360						Tanks&Temples						Deep Blending					
	SSIM †	PSNR †	LPIPS †	Train	FPS	Mem	SSIM †	PSNR †	LPIPS †	Train	FPS	Mem	SSIM †	PSNR †	LPIPS †	Train	FPS	Mem
Plenoxels	0.626	23.08	0.463	25m49s	6.79	2.1GB	0.719	21.08	0.379	25m5s	13.0	2.3GB	0.795	23.06	0.510	27m49s	11.2	2.7GB
INGP-Base	0.671	25.30	0.371	5m37s	11.7	13MB	0.723	21.72	0.330	5m26s	17.1	13MB	0.797	23.62	0.423	6m31s	3.26	13MB
INGP-Big	0.699	25.59	0.331	7m30s	9.43	48MB	0.745	21.92	0.305	6m59s	14.4	48MB	0.817	24.96	0.390	8m	2.79	48MB
M-NeRF360	0.792 †	27.69 †	0.237 †	48h	0.06	8.6MB	0.759	22.22	0.257	48h	0.14	8.6MB	0.901	29.40	0.245	48h	0.09	8.6MB
Ours-7K	0.770	25.60	0.279	6m25s	160	523MB	0.767	21.20	0.280	6m55s	197	270MB	0.875	27.78	0.317	4m35s	172	386MB
Ours-30K	0.815	27.21	0.214	41m33s	134	734MB	0.841	23.14	0.183	26m54s	154	411MB	0.903	29.41	0.243	36m2s	137	676MB

9.3 消融

我们分离出所做的不同贡献和算法选择，并构建了一组实验来衡量它们的效果。具体测试了算法的以下方面：从 SfM 初始化、致密化策略、各向异性协方差、允许无限数量的斑块具有梯度以及球谐波的使用。每种选择的量化效果总结于表 3。

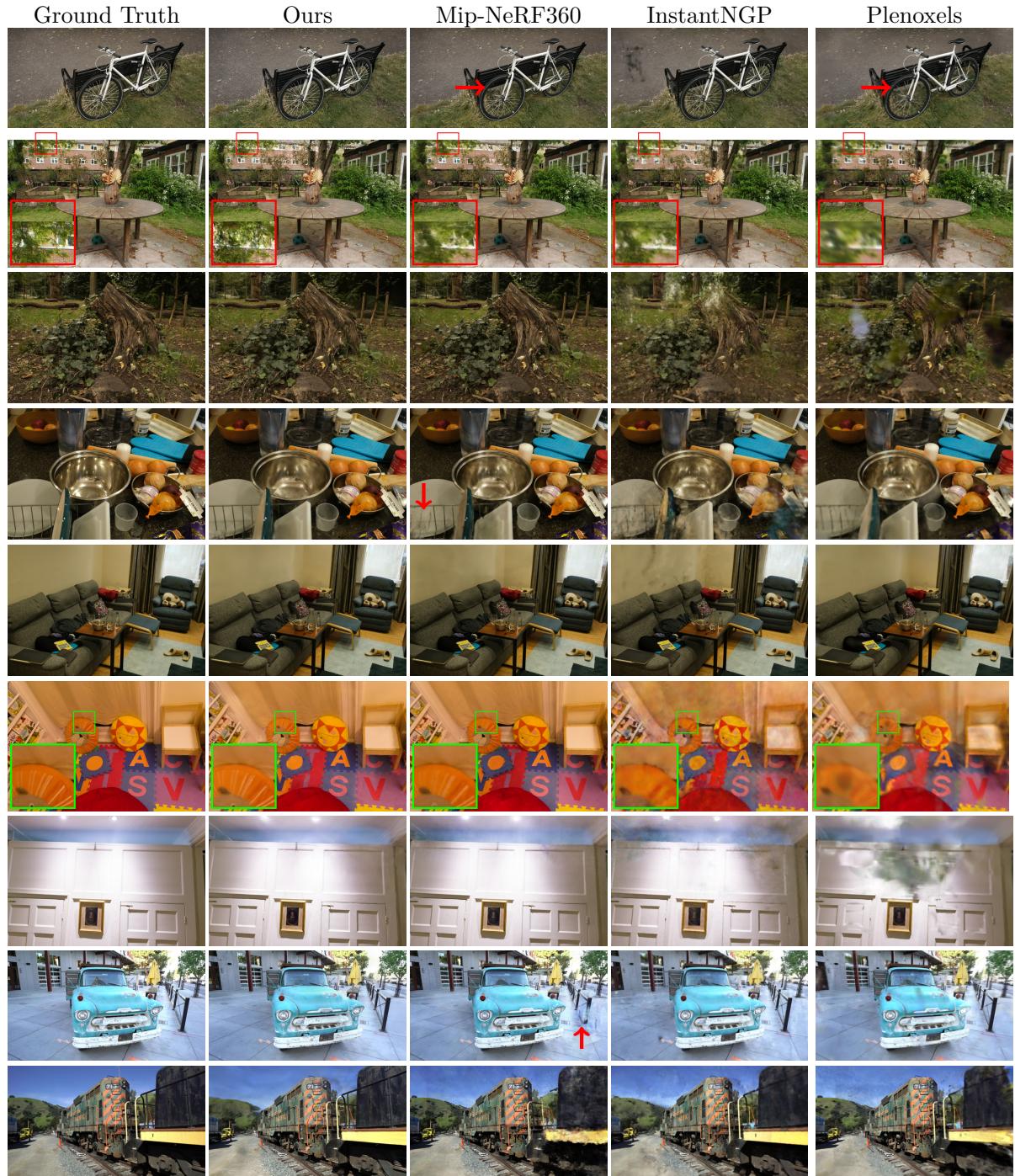


图 5：我们展示了我们的方法与先前方法的对比情况，以及来自预留测试视图的相应真实图像。场景从上到下依次为：来自 Mip-NeRF360 数据集的 BICYCLE（自行车）、GARDEN（花园）、STUMP（树桩）、COUNTER（柜台）和 ROOM（房间）；来自 Deep Blending 数据集的 PLAYROOM（游戏室）和 DRJOHNSON（约翰逊医生），以及来自 TanksTemples 的 TRUCK（卡车）和 TRAIN（火车）。质量上的细微差异通过箭头 / 插图进行了突出显示。【这里的 DrJohnson 是一个场景，可以看结果图，其实就是墙上有张 Dr. Johnson 的照片】

从 SfM 进行初始化：我们评估了从 SfM 点云初始化 3D 高斯的重要性。为此，我们对一个立方体进行均匀采样，该立方体的大小为输入相机边界框范围的三倍。结果发现，我们的方法表现相对较好，即使没有 SfM 点也能避



图 6：对于某些场景（上方所示），我们可以看到，即便在进行 7000 次迭代时（对于该场景大约耗时 5 分钟），我们的方法也已经很好地捕捉到了火车的画面。而在进行 30000 次迭代时（大约耗时 35 分钟），背景伪影已显著减少。对于其他场景（下方所示），差异几乎难以察觉；进行 7000 次迭代时（大约耗时 8 分钟）画面质量就已经很高了。

免完全失败。不过，该方法主要在背景区域表现出一定程度的退化，如图 7 所示。此外，在训练视图覆盖不佳的区域，随机初始化方法似乎会产生更多无法通过优化去除的浮点。另一方面，合成 NeRF 数据集不存在这种现象，因为其没有背景且受输入摄像机的良好约束。

致密化：接下来评估我们的两种致密化方法，即第7节中描述的克隆和拆分策略。分别禁用这两种方法，并在其余方法不变的情况下进行优化。结果表明，如图8所示，拆分大高斯对于良好的背景重建至关重要，而克隆小高斯而非拆分大高斯则可实现更好、更快的收敛，尤其是在场景中出现较薄结构时。

带有梯度的溅射斑的无限深度复杂性：我们评估了是否在最前端的点之后跳过梯度计算能够在不牺牲质量的前提下提高速度，正如 Pulsar 所建议的那样。在本次测试中，我们选择 $N = 10$ ，比 Pulsar 的默认值高出两倍，但由于梯度计算中的严重近似，导致优化不稳定。对于卡车场景，PSNR 的质量下降了 11dB（见表 3，Limited - BW），花园场景的视觉结果如图 9 所示。

各向异性协方差：我们方法中的一个重要算法选择是优化三维高斯的全协方差矩阵。为展示这一选择的效果，我们进行了一次控制实验，通过优化控制所有三个轴上三维高斯半径的单一标量值来消除各向异性。图 10 直观展示了这一优化的结果，可以观察到各向异性显著提高了三维高斯与曲面对齐的能力，从而在保持相同点数的情况下，大幅提高了渲染质量。

球谐函数：最后，球谐函数的使用提高了我们的 PSNR 总分，因为它们能够补偿视角效应（表 3）。

9.4 局限性

我们的方法并非完美无缺。在场景观察不佳的区域，会产生伪影，在这些区域其他方法也面临同样的问题（例如图 11 中的 Mip - NeRF360）。尽管各向异性高斯具有诸多优点，但我们的方法有时也会产生拉长的伪影或“斑点状”高斯（见图 12），先前的方法在这种情况下同样表现不佳。【其实真实效果没有想象的这么好，3DGS 出现伪影

表 2: 合成神经辐射场 (Synthetic NeRF) 的峰值信噪比 (PSNR) 分数，我们从 10 万个随机初始化的点开始。对比指标从相应论文中提取。

	Mic	Chair	Ship	Materials	Lego	Drums	Ficus	Hotdog	Avg.
Plenoxels	33.26	33.98	29.62	29.14	34.10	25.35	31.83	36.81	31.76
INGP-Base	36.22	35.00	31.10	29.78	36.39	26.02	33.51	37.40	33.18
Mip-NeRF	36.51	35.14	30.41	30.71	35.70	25.48	33.29	37.48	33.09
Point-NeRF	35.95	35.40	30.97	29.61	35.04	26.06	36.13	37.30	33.30
Ours-30K	35.36	35.83	30.80	30.00	35.78	26.15	34.87	37.72	33.32

表 3: 消融实验的峰值信噪比 (PSNR) 分数。在本次实验中，我们将每个场景输入图像的高分辨率版本手动下采样至我们其他实验所设定的渲染分辨率。这样做能够减少随机伪影（例如，因预缩放前的 Mip-NeRF360 输入图像存在 JPEG 压缩而产生的伪影）。

	Truck-5K	Garden-5K	Bicycle-5K	Truck-30K	Garden-30K	Bicycle-30K	Average-5K	Average-30K
Limited-BW	14.66	22.07	20.77	13.84	22.88	20.87	19.16	19.19
Random Init	16.75	20.90	19.86	18.02	22.19	21.05	19.17	20.42
No-Split	18.31	23.98	22.21	20.59	26.11	25.02	21.50	23.90
No-SH	22.36	25.22	22.88	24.39	26.59	25.08	23.48	25.35
No-Clone	22.29	25.61	22.15	24.82	27.47	25.46	23.35	25.91
Isotropic	22.40	25.49	22.81	23.89	27.00	24.81	23.56	25.23
Full	22.71	25.82	23.18	24.81	27.70	25.65	23.90	26.05

的概率和状况都会比 NeRF 严重很多，但是这个方法确实足够快，而且指标上看似乎不低】

当优化产生大高斯时，偶尔会出现“弹出”伪影，这种情况通常发生在视图相关的区域。产生这些“弹出伪影”的原因之一是光栅化器中的保护带简单地剔除了高斯，若采用更合理的剔除方法将有助于减少此类伪影。另一个因素是我们的可见度算法较为简单，可能导致高斯突然切换深度/混合顺序，这可通过抗锯齿技术解决，我们将其作为未来的研究方向。此外，目前我们没有对优化进行任何正则化处理，若进行正则化将有助于解决未见区域和弹出伪影问题。

虽然在全面评估中使用了相同的超参数，但早期实验表明，在超大型场景（如城市数据集）中，降低位置学习率可能是实现收敛的必要条件。【现在，已经有很多发表在 ECCV, CVPR, SIGGRAPH, NeurIPS 等等顶会上的文章解决了这个问题】尽管与先前基于点的方法相比，我们的方案较为紧凑，但内存消耗明显高于基于 NeRF 的解决方案。在大型场景的训练过程中，未优化的原型的 GPU 内存消耗峰值可能超过 20GB。不过，通过对优化逻辑进行细致的底层实现（类似于 InstantNGP），这一数字有望大幅降低。渲染训练好的场景需要足够的 GPU 内存来存储完整模型（大型场景需要几百兆），光栅化器还需要 30 – 500MB，具体取决于场景大小和图像分辨率。值得注意的是，有很多机会可以进一步降低我们方法的内存消耗，点云压缩技术是一个值得深入研究的领域。

10 讨论与结论

我们首次提出了一种能够在多种场景及捕捉方式下，真正实现实时、高质量辐射场渲染的方法，并且其所需的训练时间可与之前速度最快的方法相匹敌。

我们所选用的 3D 高斯基元，在保留体积渲染优化特性的同时，能够直接实现基于泼溅的快速光栅化。我们的研究成果表明，与普遍看法相悖，连续表示并非实现快速、高质量辐射场训练的必备条件。

在训练过程中，大约 80% 的时间耗费在 Python 代码上，这是因为我们基于 PyTorch 构建解决方案，旨在方便他人使用我们的方法。其中，仅有光栅化程序是以优化的 CUDA 内核形式实现。我们期望能像 InstantNGP 那样，将剩余的优化工作全部移植到 CUDA 中，如此一来，对于对性能要求极高的应用，速度将得到大幅提升。

我们还证实了基于实时渲染原理，充分发挥 GPU 强大功能以及利用软件光栅化流水线架构速度的重要性。这些设计选择是提升训练和实时渲染性能的关键所在，相较于以往的体积光线行进技术，在性能方面更具竞争优势。

若我们的高斯模型能够应用于捕捉场景的网格重建，那将会是十分有趣的探索。鉴于网格的广泛应用，这不仅



图 7：利用运动恢复结构（SfM）点进行初始化是有帮助的。上方：使用随机点云进行初始化。下方：利用运动恢复结构（SfM）点进行初始化。

具有实际意义，还能使我们更深入地了解我们的方法在体积表示和表面表示之间的具体定位。

综上所述，我们成功提出了首个辐射场实时渲染解决方案，其渲染质量可与之前成本高昂的方法相媲美，训练时间也能与现有最快的解决方案相抗衡。

11 致谢

本研究由欧洲研究理事会高级基金 FUNGRAPH (编号 788065, 网址: <http://fungraph.inria.fr>) 资助。作者对 Adobe 公司的慷慨捐赠、蔚蓝海岸大学提供的 OPAL 基础设施以及 GENCI-IDRIS 公司提供的高性能计算 (HPC) 资源 (拨款编号 2022-AD011013409) 表示感谢。同时，也感谢匿名审稿人提出的宝贵意见，感谢 P. Hedman 和 A. Tewari 对早期草稿进行校对，以及感谢 T. Müller、A. Yu 和 S. Fridovich-Keil 在比较工作方面提供的帮助。

12 附录

12.1 梯度计算细节

回想一下， Σ / Σ' 分别是高斯的世界空间/视角空间协方差矩阵， q 表示旋转， s 表示缩放， W 是视角变换， J 是射影变换仿射近似的雅可比矩阵。



图 8：针对“克隆”和“拆分”这两种情况（第 7 节）的致密化策略消融实验。

我们可以应用链式法则来求关于缩放和旋转的导数：

$$\frac{d\Sigma'}{ds} = \frac{d\Sigma'}{d\Sigma} \frac{d\Sigma}{ds} \quad (8)$$

以及

$$\frac{d\Sigma'}{dq} = \frac{d\Sigma'}{d\Sigma} \frac{d\Sigma}{dq} \quad (9)$$



图 9：如果我们限制接收梯度的点的数量，那么对视觉质量的影响是很显著的。左图：限制为 10 个接收梯度的高斯点。右图：我们的完整方法。

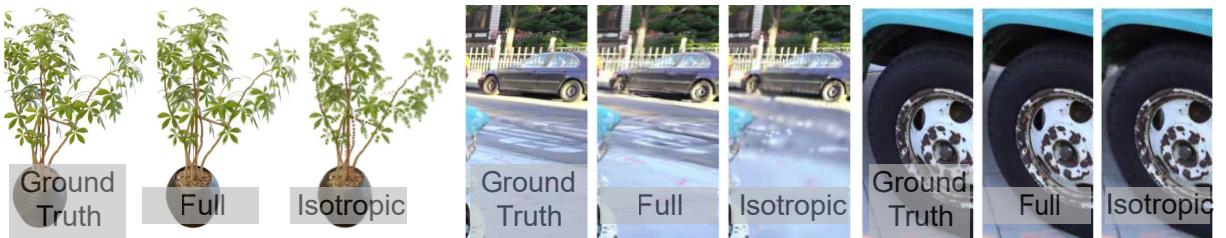


图 10：我们对禁用和启用高斯各向异性的场景进行训练。使用各向异性体素泼溅（Volumetric Splats）能够对精细结构进行建模，并且对视觉质量有着重大影响。请注意，出于演示目的，在两种配置下，我们都将“榕树”（Ficus）场景限制为使用不超过 5000 个高斯。【各向异性体素：将场景分割成许多体素（体积像素），然后通过对体素的渲染计算，生成整个场景的视觉效果。属性在不同方向上有差异，例如一个拉伸的椭球体。】



图 11：失败伪影对比：Mip-NeRF360 存在“悬浮物”以及颗粒状外观（左图，前景部分），而我们的方法会产生粗糙的、各向异性的高斯，导致视觉细节较少（右图，背景部分）。火车场景。

使用 $U = JW$ 并将 Σ' 视作 $U\Sigma U^T$ 的左上角 2×2 对称矩阵（用下标表示矩阵元素）来简化公式，我们可以求得偏导数

$$\frac{\partial \Sigma'}{\partial \Sigma_{ij}} = \begin{pmatrix} U_{1,i}U_{1,j} & U_{1,i}U_{2,j} \\ U_{1,j}U_{2,i} & U_{2,i}U_{2,j} \end{pmatrix}$$

接下来，我们求导数 $\frac{d\Sigma}{ds}$ 和 $\frac{d\Sigma}{dq}$ 。由于 $\Sigma = RSS^T R^T$ ，我们可以计算 $M = RS$ 并将 Σ 重写为 $\Sigma = MM^T$ 。这样，我们就能写成 $\frac{d\Sigma}{ds} = \frac{d\Sigma}{dM} \frac{dM}{ds}$ 以及 $\frac{d\Sigma}{dq} = \frac{d\Sigma}{dM} \frac{dM}{dq}$ 。因为协方差矩阵 Σ （及其梯度）是对称的，所以共有的第一部



图 12：在与训练期间所见视图几乎没有重叠的视图中，我们的方法可能会产生伪影（右图）。同样，在这些情况下，Mip-NeRF360 也会出现伪影（左图）。约翰逊医生场景。

分可紧凑地由 $\frac{d\Sigma}{dM} = 2M^T$ 得出。对于缩放，我们进一步得到

$$\frac{\partial M_{i,j}}{\partial s_k} = \begin{cases} R_{i,k} & \text{如果 } j = k \\ 0 & \text{否则} \end{cases}$$

为推导旋转的梯度，我们回顾一下从实部为 q_r 、虚部为 q_i 、 q_j 、 q_k 的单位四元数 q 到旋转矩阵 R 的转换：

$$R(q) = 2 \begin{pmatrix} \frac{1}{2} - (q_j^2 + q_k^2) & (q_i q_j - q_r q_k) & (q_i q_k + q_r q_j) \\ (q_i q_j + q_r q_k) & \frac{1}{2} - (q_i^2 + q_k^2) & (q_j q_k - q_r q_i) \\ (q_i q_k - q_r q_j) & (q_j q_k + q_r q_i) & \frac{1}{2} - (q_i^2 + q_j^2) \end{pmatrix} \quad (10)$$

结果，我们得到 q 各分量的以下梯度：

$$\begin{aligned} \frac{\partial M}{\partial q_r} &= 2 \begin{pmatrix} 0 & -s_y q_k & s_z q_j \\ s_x q_k & 0 & -s_z q_i \\ -s_x q_j & s_y q_i & 0 \end{pmatrix}, \\ \frac{\partial M}{\partial q_i} &= 2 \begin{pmatrix} 0 & s_y q_j & s_z q_k \\ s_x q_j & -2s_y q_i & -s_z q_r \\ s_x q_k & s_y q_r & -2s_z q_i \end{pmatrix}, \\ \frac{\partial M}{\partial q_j} &= 2 \begin{pmatrix} -2s_x q_j & s_y q_i & s_z q_r \\ s_x q_i & 0 & s_z q_k \\ -s_x q_r & s_y q_k & -2s_z q_j \end{pmatrix}, \\ \frac{\partial M}{\partial q_k} &= 2 \begin{pmatrix} -2s_x q_k & -s_y q_r & s_z q_i \\ s_x q_r & -2s_y q_k & s_z q_j \\ s_x q_i & s_y q_j & 0 \end{pmatrix} \end{aligned} \quad (11)$$

推导四元数归一化的梯度是较为直接的。

12.2 优化与致密化算法

我们的优化和致密化算法在算法 1 中进行了总结。

12.3 光栅化器细节

排序 我们的设计基于大量小泼溅负载的假设，并通过在开始时使用基数排序对每帧的泼溅进行一次排序来对此进行优化。

我们将屏幕划分为 16×16 像素的瓦片。通过在每个 16×16 瓦片内对其重叠的每个泼溅进行实例化，我们为每个瓦片创建一个泼溅列表。这会使要处理的高斯数量适度增加，不过，通过更简单的控制流以及优化后的 GPU 基数排序的高度并行性，这种增加可以得到分摊。

我们为每个泼溅实例分配一个最多 64 位的键，其中低 32 位对其投影深度进行编码，高位对重叠瓦片的索引进行编码。索引的具体大小取决于当前分辨率下能容纳的瓦片数量。这样，通过单次基数排序，所有泼溅的深度排序就能直接并行解决。排序后，我们可以通过识别排序数组中具有相同瓦片 ID 的范围的起始和结束位置，高效地生成每

Algorithm 1 优化与密集化

w, h : 训练图像的宽度和高度

```
 $M \leftarrow$  SfM 点                                ▷ 位置
 $S, C, A \leftarrow$  初始化属性 ()                  ▷ 协方差, 颜色, 透明度
 $i \leftarrow 0$                                     ▷ 迭代计数
while 未收敛 do
     $V, \hat{I} \leftarrow$  抽样训练视图 ()                ▷ 相机  $V$  和图像
     $I \leftarrow$  光栅化 ( $M, S, C, A, V$ )            ▷ 算法 2
     $L \leftarrow$  损失 ( $I, \hat{I}$ )                      ▷ 损失
     $M, S, C, A \leftarrow$  Adam( $\nabla L$ )           ▷ 反向传播与步进
    if 是精细化迭代 ( $i$ ) then
        for all 高斯分布  $(\mu, \Sigma, c, \alpha)$  in  $(M, S, C, A)$  do
            if  $\alpha < \epsilon$  或过大  $(\mu, \Sigma)$  then          ▷ 修剪
                移除高斯分布 ()
            end if
            if  $\nabla_p L > \tau_p$  then                  ▷ 密集化
                if  $\|S\| > \tau_S$  then                  ▷ 过度重建
                    分割高斯分布  $(\mu, \Sigma, c, \alpha)$ 
                else                                ▷ 未充分重建
                    克隆高斯分布  $(\mu, \Sigma, c, \alpha)$ 
                end if
            end if
        end for
    end if
     $i \leftarrow i + 1$ 
end while
```

个瓦片要处理的高斯列表。这是并行完成的，为每个 64 位数组元素启动一个线程，将其高 32 位与其两个相邻元素进行比较。

与 Lassner 等人相比，我们的光栅化方法完全消除了顺序的图元处理步骤，并在正向传递过程中生成了更紧凑的每个瓦片要遍历的列表。

12.4 数值稳定性

在反向传播过程中，我们通过将正向传播中累积的不透明度除以每个高斯的 α 值来重建梯度计算所需的中间不透明度值。如果简单实现，这个过程容易出现数值不稳定的情况（例如，除以 0）。为解决此问题，在正向和反向传播中，我们跳过任何 $\alpha < \epsilon$ （我们选择 ϵ 为 $\frac{1}{255}$ ）的混合更新，并将 α 从上方限制为 0.99。最后，在将一个高斯纳入正向光栅化传递之前，我们计算如果将其纳入时累积的不透明度，并在其正向到反向混合之前确保其不会超过 0.9999。

12.5 逐场景误差指标

表4至表9列出了我们针对所有考虑的技术和真实世界场景进行评估时收集的各种误差指标。我们列出了复制的 Mip - NeRF360 的数值以及我们用于生成文中图像的运行结果数值；在整个 Mip - NeRF360 数据集上这些数值的平均值分别为：PSNR 为 27.58，SSIM 为 0.790，LPIPS 为 0.240。

Algorithm 2 GPU 软件光栅化 3D 高斯函数

w, h : 要光栅化的图像的宽度和高度

M, S : 世界空间中的高斯均值和协方差

C, A : 高斯颜色和透明度

V : 当前相机的视图配置

```
function 光栅化 ( $w, h, M, S, C, A, V$ )
    剔除高斯 ( $p, V$ )                               ▷ 视锥剔除
     $M', S' \leftarrow$  屏幕空间高斯 ( $M, S, V$ )         ▷ 变换
     $T \leftarrow$  创建瓦片 ( $w, h$ )
     $L, K \leftarrow$  重复并标记 ( $M', T$ )               ▷ 索引和键
    按键排序 ( $K, L$ )                               ▷ 全局排序
     $R \leftarrow$  标识瓦片范围 ( $T, K$ )
     $I \leftarrow \mathbf{0}$                                    ▷ 初始化画布
    for all 瓦片  $t$  in  $I$  do
        for all 像素  $i$  in  $t$  do
             $r \leftarrow$  获取瓦片范围 ( $R, t$ )
             $I[i] \leftarrow$  按序混合 ( $i, L, r, K, M', S', C, A$ )
        end for
    end for
    return  $I$ 
end function
```

表 4: Mip-NeRF360 场景的 SSIM 分数。† 从原始论文复制。

	自行车	花朵	花园	树桩	山丘树	房间	柜台	厨房	盆景
Plenoxels	0.496	0.431	0.6063	0.523	0.509	0.8417	0.759	0.648	0.814
INGP-Base	0.491	0.450	0.649	0.574	0.518	0.855	0.798	0.818	0.890
INGP-Big	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
Mip-NeRF360	0.685	0.584	0.809	0.745	0.631	0.910	0.892	0.917	0.938
Ours-7k	0.675	0.525	0.836	0.728	0.598	0.884	0.873	0.900	0.910
Ours-30k	0.771	0.605	0.868	0.775	0.638	0.914	0.905	0.922	0.938

表 5: Mip-NeRF360 场景的 PSNR 分数。† 从原始论文复制。

	自行车	花朵	花园	树桩	山丘树	房间	柜台	厨房	盆景
Plenoxels	21.912	20.097	23.4947	20.661	22.248	27.594	23.624	23.420	24.669
INGP-Base	22.193	20.348	24.599	23.626	22.364	29.269	26.439	28.548	30.337
INGP-Big	22.171	20.652	25.069	23.466	22.373	29.690	26.691	29.479	30.685
Mip-NeRF360 [†]	24.37	21.73	26.98	26.40	22.87	31.63	29.55	32.23	33.46
Mip-NeRF360	24.305	21.649	26.875	26.175	22.929	31.467	29.447	31.989	33.397
Ours-7k	23.604	20.515	26.245	25.709	22.085	28.139	26.705	28.546	28.850
Ours-30k	25.246	21.520	27.410	26.550	22.490	30.632	28.700	30.317	31.980

表 6: Mip-NeRF360 场景的 LPIPS 分数。† 从原始论文复制。

	自行车	花朵	花园	树桩	山丘树	房间	柜台	厨房	盆景
Plenoxels	0.506	0.521	0.3864	0.503	0.540	0.4186	0.441	0.447	0.398
INGP-Base	0.487	0.481	0.312	0.450	0.489	0.301	0.342	0.254	0.227
INGP-Big	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
Mip-NeRF360	0.305	0.346	0.171	0.265	0.347	0.213	0.207	0.128	0.179
Ours-7k	0.318	0.417	0.153	0.287	0.404	0.272	0.254	0.161	0.244
Ours-30k	0.205	0.336	0.103	0.210	0.317	0.220	0.204	0.129	0.205

表 7: Tanks&Temples 和 Deep Blending 场景的 SSIM 分数。

	卡车	火车	约翰逊医生	游戏室
Plenoxels	0.774	0.663	0.787	0.802
INGP-Base	0.779	0.666	0.839	0.754
INGP-Big	0.800	0.689	0.854	0.779
Mip-NeRF360	0.857	0.660	0.901	0.900
Ours-7k	0.840	0.694	0.853	0.896
Ours-30k	0.879	0.802	0.899	0.906

表 8: Tanks&Temples 和 Deep Blending 场景的 PSNR 分数。

	卡车	火车	约翰逊医生	游戏室
Plenoxels	23.221	18.927	23.142	22.980
INGP-Base	23.260	20.170	27.750	19.483
INGP-Big	23.383	20.456	28.257	21.665
Mip-NeRF360	24.912	19.523	29.140	29.657
Ours-7k	23.506	18.892	26.306	29.245
Ours-30k	25.187	21.097	28.766	30.044

表 9: Tanks&Temples 和 Deep Blending 场景的 LPIPS 分数。

	卡车	火车	约翰逊医生	游戏室
Plenoxels	0.335	0.422	0.521	0.499
INGP-Base	0.274	0.386	0.381	0.465
INGP-Big	0.249	0.360	0.352	0.428
Mip-NeRF360	0.159	0.354	0.237	0.252
Ours-7k	0.209	0.350	0.343	0.291
Ours-30k	0.148	0.218	0.244	0.241

13 最后总结

回顾这学期专业英语课程学习，我深刻感受到了这门课程对我个人成长的巨大帮助。这不仅仅是一门提升语言能力的课程，更是帮助我在学术发展中迈出关键一步的重要平台。从最初对专业英语表达感到陌生，到现在能够自如地进行学术交流和论文撰写，这其中的点滴进步让我对未来的发展充满了信心。尤其是作为一名有出国深造计划的学生，这门课程让我更加意识到，掌握好专业英语不仅是一个工具，更是一种通向国际学术舞台的桥梁。

在课程的学习中，我要特别感谢我们的授课老师王辉老师。他不仅具备深厚的专业素养，还以极大的耐心和责任心引导我们逐步提高语言能力。在课堂上，王老师通过严谨的教学方式，使我对专业术语的理解更加透彻，同时也让我更加清楚如何在国际学术环境中进行精准、严谨的表达。通过专业英语课程的学习，我可以初步了解到人工智能领域的相关知识，并发现到自己感兴趣的点。他还特别关注我们每一位同学的学习情况，课上的互动来激发我们的学习积极性。王老师的教学不仅让我提升了语言能力，更让我对如何在学术研究中保持严谨和创新有了更加深刻的理解。

这门课程的学习也为我的学术研究带来了巨大的帮助。我现在的研究方向是医学图像分析。目前，我已经完成了两篇学术论文，并以第一作者身份投稿。第一篇论文题为《EM-Net: Effective and Morphology-aware Network for Skin Lesion Segmentation》，目前已投稿至 *Expert Systems with Applications*，并处于修订阶段。该论文提出了一种高效且具有形态意识的网络，用于皮肤病变的分割，代码已开源于 EM-Net。第二篇论文题为《An Annotated Heterogeneous Ultrasound Database》，已投稿至 *Scientific Data*，同样处于修订阶段。这篇论文旨在为超声图像领域提供一个经过详细注释的异构数据库，相关资源已开源于 AHU-Database。

完成这些研究的过程中，我深刻体会到专业英语的学习对科研的重要性。无论是撰写论文摘要、讨论研究意义，还是与期刊编辑沟通，清晰准确的英语表达都是不可或缺的。在这方面，专业英语课程的学习让我能够更加自信地完成这些任务，从而使我的研究成果得以顺利传播到更广泛的学术群体中。

至于我为何选择翻译这篇论文，是因为这篇论文和我现在做的课题有着紧密的联系，我目前正在与金老师和董老师合作进行一项基于 3D Gaussian Splatting 的超声图像渲染方法研究。这项研究旨在通过先进的 3D 渲染技术提升超声图像的可视化效果，为临床诊断提供更加直观和准确的支持。同时，在中国国家留学基金委员会 (CSC) 的支持下，我也与加拿大阿尔伯塔大学程立教授的团队展开了深入合作。研究课题和这篇文章紧密相关。在这个过程

中，我需要与不同文化背景的学者进行频繁的交流，包括讨论研究进展、撰写合作计划、以及参加国际会议。专业英语课程为我提供了宝贵的语言支持，使我能够更加从容地与国际团队交流思想，展示我的研究成果。

Dear Yuezhe Yang,

Congratulations! Your application has been **selected** by University of Alberta to move forward in the China Scholarship Council-funded research internship program for Summer 2025.

This email serves as proof of your preliminary **selection** for the following research project at University of Alberta:

Project Name: **Motion Analysis from Images and Videos**
Host Professor: **Li Cheng**

图 13: CSC Intern

除了学术上的帮助，这门课程还让我意识到学习英语的最终目标并不仅仅是掌握一种语言，而是通过语言更好地表达思想、解决问题，并在更大的范围内与人沟通。每一次课程中的练习、每一篇论文中的推敲，都让我更加深刻地认识到，语言能力的提升不是一蹴而就的，而是需要不断积累、不断实践。这种学习的过程让我学会了如何在困难中坚持，也让我更加珍惜学习的机会。

总结这段学习经历，我不仅提升了专业英语能力，也更加明确了未来的方向。无论是在现阶段的科研工作中，还是在未来的国际学术交流中，这门课程都将成为我发展的重要基石。我衷心感谢王辉老师的悉心指导，以及他为我们提供的支持。

未来，我将继续以这段经历为动力，进一步提升自己的学术能力，努力在医学影像领域做出更多的贡献。我希望能够通过我的研究，帮助更多的人，同时也为推动学术进步尽自己的绵薄之力。

参考文献

- [1] *graphdeco-inria/gaussian-splatting: Original reference implementation of "3D Gaussian Splatting for Real-Time Radiance Field Rendering"*. URL: <https://github.com/graphdeco-inria/gaussian-splatting> (visited on 12/31/2024).
- [2] 国内首个面向科研 & 落地的 3DGS 全栈实战教程来啦! Weixin Official Accounts Platform. URL: <https://mp.weixin.qq.com/s/ONMaouwNk8Nwj8nm7kHDg> (visited on 12/30/2024).
- [3] Kedreamix. 万字长文详解什么是 3D Gaussian Splatting. Weixin Official Accounts Platform. URL: <https://mp.weixin.qq.com/s/VIIIF8jJNR75yM61TAhqrIA> (visited on 12/30/2024).
- [4] 3DGS 综述以及对 3DGS 的理解: A Survey on 3D Gaussian Splatting - 知乎. URL: <https://zhuanlan.zhihu.com/p/679809915> (visited on 12/31/2024).
- [5] 3D Gaussian Splatting 简明教程 - 知乎. URL: <https://zhuanlan.zhihu.com/p/665135145> (visited on 12/31/2024).
- [6] 3D 高斯点染简介 - 知乎. URL: <https://zhuanlan.zhihu.com/p/674652829> (visited on 12/31/2024).
- [7] Yanqi Bao et al. 3D Gaussian Splatting: Survey, Technologies, Challenges, and Opportunities. Dec. 17, 2024. DOI: 10.48550/arXiv.2407.17418. arXiv: 2407.17418[cs]. URL: <http://arxiv.org/abs/2407.17418> (visited on 12/31/2024).
- [8] 2D Gaussian Splatting 文章 + 代码串读 (无敌详细/引经据典/疯狂解读) - 知乎. URL: <https://zhuanlan.zhihu.com/p/708372232> (visited on 12/31/2024).