arXiv:2512.14352v1 [cs.CV] 16 Dec 2025

HGS：用于紧凑型动态视图合成的静态-动态分解混合高斯泼溅

Kaizhe Zhang* Yijie Zhou Weizhan Zhang Xuanyu Wang 西安交通大学 西安交通大学 西安交通大学 西安交通大学 Caixia Yan Haipeng Du yugui xie Yu-Hui Wen Yong-Jin Liu 西安交通大学 西安交通大学 chinamobile.com 北京交通大学 清华大学



(a)高质量结果 (b)实时4K，128每秒帧数 (c)质量、速度和大小比较

图1：与现有最先进水平的性能比较 [21, 10, 39, 1, 41, 2, 43]。我们的方法在 (a) 时间复杂场景中实现了更高质量的重建，(b) 同时保持实时渲染，(c) 并提升了性能。

摘要

动态新视角合成（NVS）对于打造沉浸式体验至关重要。现有方法通过引入带有隐式变形场的3D高斯泼溅（3DGS），或不加区分地分配时变参数，推动了动态NVS的发展，并超越了基于NeRF的方法。然而，由于模型复杂度过高且参数冗余，这些方法导致模型大小庞大、渲染速度缓慢，难以胜任实时应用，尤其是在资源受限的设备上。为获得冗余参数更少、效率更高的模型，本文提出混合高斯泼溅（HGS），这是一种紧凑高效的框架，专为在统一表征中明确分离场景的静态区域和动态区域而设计。HGS的核心创新在于静态-动态分解（SDD）策略，该策略采用径向基函数（RBF）对高斯基元进行建模。具体而言，对于动态区域，我们使用时变径向基函数有效捕获时间变化并处理突发场景变化；对于静态区域，则通过共享时间不变参数来减少冗余。此外，我们提出了一种专为显式模型设计的两阶段训练策略，以增强静态-动态边界处的时间连贯性。实验结果表明，我们的方法可将模型大小最多缩减98%，并在单张RTX 3090 GPU上以4K分辨率实现最高每秒125帧的实时渲染。在RTX 3050上，模型在 1352×1014时仍可保持每秒160帧，并已集成到VR系统中。此外，HGS在渲染质量上可与最先进的方法相媲美，同时在高频细节和突发场景变化方面显著提升了视觉保真度。

索引词：实时渲染，高斯泼溅，动态场景建模，虚拟现实。

*电子邮箱：zkz1081@stu.xjtu.edu.cn

1 引言

动态新视角合成（NVS）旨在从新视角重建动态场景的照片级真实感图像。由于其在增强现实、虚拟现实、远程临场和沉浸式媒体流传输等领域具有广泛应用，这项任务一直是计算机视觉和图形学领域的研究重点。

隐式神经表示（尤其是神经辐射场（NeRF））的最新进展显著提升了静态场景建模的保真度与便捷性。然而，基于NeRF的方法通常训练时间较长，渲染速度也较慢。尽管已有大量研究致力于缓解这些局限，但要同时实现高质量与快速渲染仍然颇具挑战。

为解决NeRF因密集体采样而产生的计算瓶颈，3D高斯泼溅（3DGS）等显式方法应运而生。3DGS使用可微3D高斯基元对场景进行显式建模，并可将其高效光栅化到图像平面上。这种光栅化策略在保持高视觉保真度的同时，显著提升了渲染速度，解决了NeRF在计算效率方面的局限。尽管3DGS在静态场景建模中取得了成功，但将其扩展到动态场景仍然面临挑战。已有研究证明了其在静态场景下的有效性，但在动态场景中会出现帧间不连续。近期方法采用隐式表示来捕捉动态变化，但这会带来较高的计算成本和较慢的渲染速度，且无法应对突发的运动变化。为提高效率，研究者探索了时间相关模块或4D高斯扩展等显式技术。然而，这些方法会为静态区域分配动态参数，造成参数冗余、复杂度过高和效率降低。

总而言之，现有的动态 3DGS 方法面临两大挑战：(1) 为静态内容分配了过多时间参数，导致显著的计算和内存开销；(2) 时间伪影以及静态区域高频细节的丢失会造成视觉质量下降。此外，常用的时间建模方法

难以捕捉场景中的突变，限制了动态场景重建的保真度和鲁棒性。

为解决上述局限性，我们提出了一种新颖的混合高斯泼溅（HGS）框架，用于动态场景表征，旨在降低存储成本、提升渲染速度并实现高质量效果。为降低存储成本，我们采用静态-动态分解（SDD）策略，分别部署静态和动态高斯基元，从而明确分离静态区域与动态区域。对于静态区域，我们共享参数进行建模，避免承担不必要的动态参数开销，从而降低计算开销。对于动态区域，我们仅使用时间径向基函数（RBF） [21]，避免采用变形场等隐式运动建模方式。该方案进一步降低了计算开销，并提升了渲染速度。

为抑制时间伪影，我们提出的SDD策略将时间径向基函数 [21] 限制在动态区域内。此约束有效减少了跨区域时间干扰，保留了静态内容中的高频细节和视觉保真度。我们通过显式分离将动态运动与静态区域解耦，不仅提高了运动精度，也显著提升了整体渲染质量。为进一步增强静态区域与动态区域边界处的一致性，我们针对显式高斯表示提出了两阶段训练策略。在每个优化周期内，训练过程交替优化静态和动态基元。这种迭代优化策略有助于静态与动态组件相互适应。结合使用时间径向基函数对动态区域进行显式建模，我们的方法能够有效应对突发场景变化，并大幅减少静态-动态边界处的伪影。

得益于显式的 SDD策略，我们的 HGS 在单张 RTX 3090 上以 1080p 分辨率实现了超过 300 FPS，同时所需参数显著更少。用于动态区域的时间径向基函数以及我们提出的两阶段训练策略有效减少了时间伪影。因此，尽管参数数量减少，我们的 HGS 在峰值信噪比和结构相似性指数方面仍优于或媲美近期最先进的方法 [47]。在大多数场景中，都能观察到视觉质量的提升，这归功于时间伪影的抑制以及静态区域中高频细节的保留。我们的主要贡献总结如下：

- 我们提出了一种新颖的 HGS 框架，用于紧凑、高效的动态场景重建，在统一的渲染与优化流程中显式分离静态和动态组件。
- 我们提出了 SDD策略，通过为所有静态基元分配一组紧凑且随时间不变的参数来实现参数共享。结合显式建模，SDD 显著降低了模型复杂度，从而实现更快的渲染，同时保持与动态建模的兼容性。
- 我们针对显式高斯表示设计了一种两阶段训练策略。这种迭代方法通过相互适应增强时间一致性，并有效减少静态-动态边界处的伪影。
- 如图所示。 1，我们的方法在将模型大小最多缩减 98% 的同时，达到了可与最先进方法媲美的渲染质量，形成了更加紧凑、高效的表征。此外，该方法能更好地保留静态区域中的高频细节，并抑制突发场景变化下的时间伪影。

2 相关工作

2.1 DynamicNeRF

近期动态 NeRF [28] 领域的研究主要聚焦于扩展静态场景方法，使其能够处理时间动态。许多研究采用基于变形的方法 [12, 22, 32, 33, 34, 38,42], , 引入神经变形场，将动态场景扭曲至规范静态帧。例如，D-NeRF [34], 、Nerfies [32], 和 HyperNeRF [33]等方法通过学习形变有效捕捉复杂的场景动态，但由于需要进行密集体积采样，计算成本高昂。为缓解这些计算问题，结构化表示方法 [5, 9, 10, 35, 39, 40, 36, 1] 应运而生。K-Planes [10]和 MixVoxels [39] 将时空域分解为结构化平面或自适应体素网格，大幅降低了计算复杂度。NeRFPlayer [36] 采用显式时间分解，以更好地处理动态变化；HyperReel [1] 则提出低维时间嵌入，以提升效率。然而，这些结构化方法在准确表示快速或高度复杂的运动时仍可能面临困难。

2.2 动态 3DGS

3DGS 在渲染质量和效率方面取得的显著进展，推动后续研究 [13, 14, 18,23, 41, 45, 24, 21, 8, 46, 27, 2], 致力于将原本静态的 3DGS 框架扩展至动态场景重建。早期工作 [27, 41] 采用多层感知机回归高斯参数的逐帧变形，提升了灵活性，但也带来了显著的计算开销。ED3DGS [2] 通过引入用于变形的逐高斯潜在嵌入，并将运动分解为粗细层级，改进了这一方法。RoDyGS [14] 引入运动基，并在变形场训练过程中分离静态点和动态点，以提升鲁棒性和效率。然而，这些方法都依赖隐式运动建模，因而限制了渲染速度并增加了模型复杂度。

为克服隐式建模的局限性，4DGS [46] 通过为每个基元附加时间特征，将三维高斯扩展到四维空间；而 STGS [21] 则通过多项式函数和时间径向基函数对位置和方向的时间演化进行建模。这些方法提高了运行时效率，并避免了对隐式变形场的需求。然而，它们通常将所有高斯基元都视为动态基元，包括属于静态区域的基元，从而导致参数冗余和潜在的时间模糊。

不同于依赖隐式变形场来建模动态运动的先前工作，我们的方法采用时间径向基函数，从而避免了此类计算开销。此外，我们引入了 SSD策略，仅为动态区域分配与时间相关的参数，同时使用共享的、与时间无关的表征对静态区域进行建模。这种设计减少了参数冗余，避免了静态区域中的时间伪影，并在保持与最先进方法相当的视觉质量的同时，实现了显著更快的渲染。

3 预备知识：3D高斯泼溅

3DGS [16]是一种显式表示方法，使用可微各向异性三维高斯对场景进行建模。给定一组从多个视角拍摄的已标定图像，3DGS 通过可微光栅化优化这些高斯的高斯参数，从而有效地表征场景并高效地渲染新视角。

具体而言，场景中的每个高斯G_i 都由若干属性参数化：空间位置 $\mu_i \in \mathbb{R}^3$、定义其形状和朝向的协方差矩阵 $\Sigma_i \in \mathbb{R}^{3\times3}$ 、控制透明度的不透明度标量$\sigma_i \in \mathbb{R}$ ，以及用于编码视角相关外观的球谐函数（SH）系数$\mathbf{h}_i$ 。

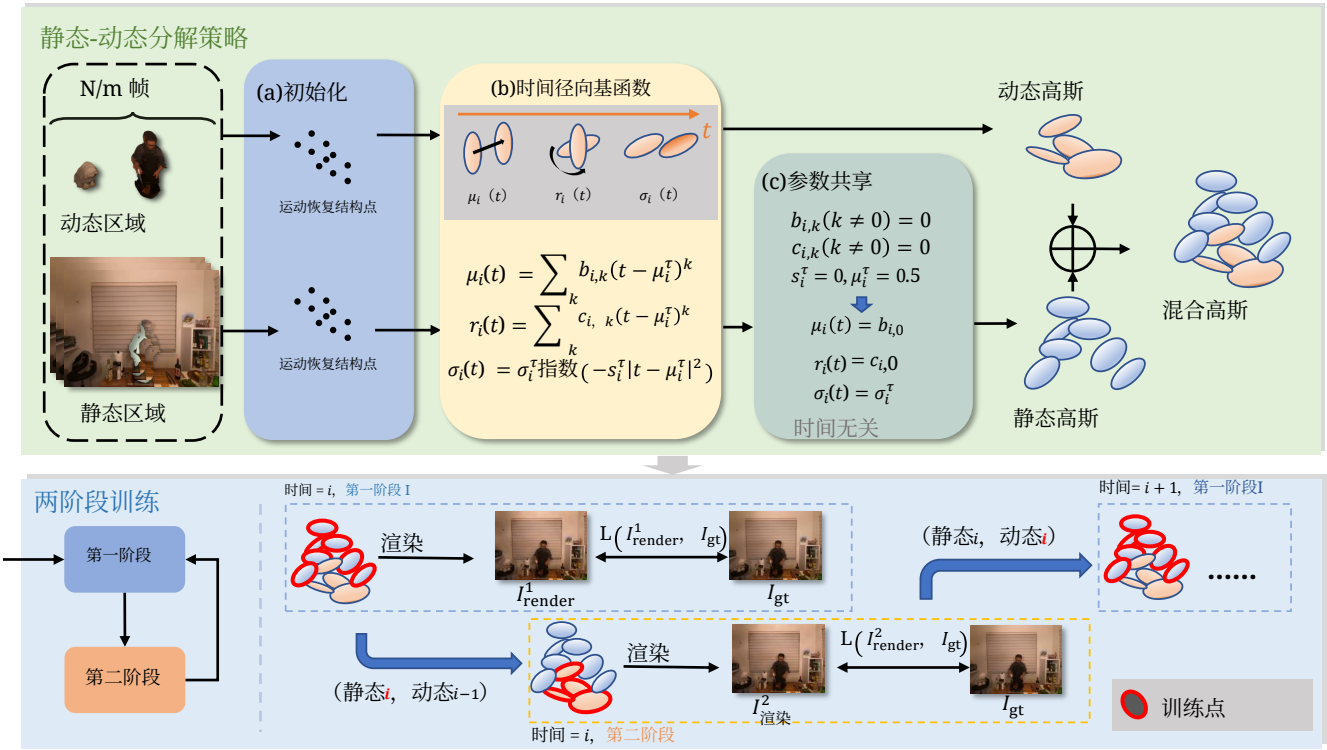

图 2：所提出的 HGS 流程概览，该流程由两个部分组成：（上）静动态分解策略，包括 (a) 初始化：每隔 m 帧选取一帧进行初始化，以减少运动恢复结构点的数量；(b) 时间径向基函数：为动态高斯基元添加与时间相关的参数；以及 (c) 参数共享：对静态高斯基元施加与时间无关的参数约束。（下）两阶段训练：第一阶段更新静态高斯基元，而第二阶段利用第一阶段优化后的静态基元联合优化动态高斯基元。

任意空间位置$x \in \mathbb{R}^3$ 处的三维高斯核可形式化表示为：

$$\alpha_i(x) = \sigma_i \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right), \tag{1}$$

其中，协方差矩阵 Σ_i 是对称半正定的。为确保这些约束并实现直观的几何解释，Σ_i 通过缩放和旋转分量进行分解：

$$\Sigma_i = R_i S_i S_i^T R_i^T, \tag{2}$$

其中，S_i 为由缩放向量 $s_i \in \mathbb{R}^3$ 参数化的对角缩放矩阵，$R_i \in \mathbb{R}^{3\times3}$ 为由四元数表示 $q_i \in \mathbb{R}^4$导出的旋转矩阵，并施加单位范数约束以确保稳定性。

为了从特定视点渲染图像，每个三维高斯都会通过近似透视投影，从三维世界坐标系投影到二维图像平面。给定相机外参（视图矩阵）$W \in \mathbb{R}^{4\times4}$ 和内参（投影矩阵）$K \in \mathbb{R}^{3\times4}$，将三维位置 μ_i 和协方差 Σ_i 投影后，可得到相应的二维高斯分布，其均值为 μ_i^{2D}，协方差矩阵为 Σ_i^{2D}：

$$\mu_i^{2D} = \left(K\frac{W\mu_i}{(W\mu_i)_z}\right)_{1:2}, \tag{3}$$

$$\Sigma_i^{2D} = (JW\Sigma_i W^T J^T)_{1:2,1:2}, \tag{4}$$

其中，$(W\mu_i)_z$ 表示经过视图变换后的深度，而J是透视投影的雅可比矩阵，体现了非线性投影变换的局部线性化。

获得投影后的高斯后，沿相机观察方向通过阿尔法合成进行渲染。首先根据高斯相对于相机的深度，按降序对其排序。通过对影响该像素的高斯进行阿尔法混合，计算像素坐标处的最终像素颜色I：

$$I = \sum_{i\in N} c_i \alpha_i^{2D} \prod_{j=1}^{i-1}(1-\alpha_j^{2D}), \tag{5}$$

其中，N 表示影响该像素的有序高斯集合，像素位置x_{2D} 处的不透明度 α_i^{2D}按三维高斯核的二维对应形式计算：

$$\alpha_i^{2D}(x_{2D}) = \sigma_i \exp\left(-\frac{1}{2}(x_{2D}-\mu_i^{2D})^T(\Sigma_i^{2D})^{-1}(x_{2D}-\mu_i^{2D})\right), \tag{6}$$

并且，颜色c_i 是根据与相应观察方向d_v对应的球谐函数（SH）系数 h_i 计算得出的：

$$c_i = \text{SH}(h_i, d_v), \tag{7}$$

其中，SH(·) 表示球谐函数的求值。

4 方法

在本节中，我们介绍 HGS 框架，用于以更低的存储成本、更快的渲染速度和更高的质量来表征动态场景。我们通过引入 SDD

策略和两阶段训练策略来实现这一目标。这种设计不仅能够实现更紧凑的表征，还支持高效、高质量的渲染。我们的方法概述如图2所示。第4.1 节介绍混合高斯基元。第4.2 节介绍用于精细建模的 SDD策略。第4.3 节介绍两阶段训练策略，随后在第4.4节详述优化。

4.1 混合高斯

为高效且紧凑地表示动态4D场景，我们在统一的渲染与优化框架中显式地区分静态基元和动态基元。受近期关于 STGS 的研究启发 [21], ，我们通过显式建模高斯基元的时间演化，将传统的3D高斯基元扩展至4D时空域。具体而言，每个高斯基元 G_i 由随时间变化的空间位置 $\mu_i(t)$、不透明度 $\sigma_i(t)$、旋转四元数 $r_i(t)$，以及随时间保持不变的各向异性尺度向量 s_i 和颜色系数 c_i进行参数化。这些参数的具体表达式分别见式（1）、式（2）和式（7）。在此建模框架下，我们提出一种新颖方法，利用二元指示变量区分静态基元与动态基元，并针对每类基元采用不同的参数化策略。

4.1.1 动态基元

对于动态基元，每个高斯都独立维护自身随时间变化的参数。每个动态基元的时间演化均通过显式参数化表示。具体而言，每个高斯参数均采用多项式插值建模。详细表达式见第 4.2.2 节。

4.1.2 静态基元

所有静态高斯基元共享一组统一且时间不变的参数。借助这种共享参数化方式，我们的方法显著减少了冗余，从而为静态场景组件提供紧凑而高效的表征。参数共享策略的详细说明见第 4.2.3 节。

4.2 静态-动态分解策略

为更有效地实现这一思路，我们提出 SDD策略，在混合高斯框架中具体实现对动态区域和静态区域的分别建模。我们首先进行初始化，以获得这两类区域对应的高斯基元。对于动态区域，我们应用时间径向基函数 [21] 来捕捉随时间变化的行为。相比之下，静态区域共享一组时间不变参数，从而无需承担冗余动态高斯带来的开销即可实现高效建模。

4.2.1 初始化

为实现紧凑高效的场景表征，我们采用交错帧初始化方案。我们不再依赖所有帧的稠密 SfM点云，而是使用按时间下采样帧的重建结果来初始化混合高斯基元。这种方法限制了初始高斯基元的数量，在计算效率与捕捉场景所需的保真度之间取得了良好平衡。

4.2.2 径向基函数

对于动态基元，我们对 STGS 的时间径向基函数进行调整 [21]，以显式参数化时间演化。其公式如下：

$$\mu_i(t) = \sum_{k=0}^{3} b_{i,k}(t-\mu_i^\tau)^k, \tag{8}$$

$$r_i(t) = \sum_{k=0}^{1} c_{i,k}(t-\mu_i^\tau)^k, \tag{9}$$

$$\sigma_i(t) = \sigma_i^\tau \exp\left(-s_i^\tau |t-\mu_i^\tau|^2\right), \tag{10}$$

其中， $\mu_i(t)$ 和 $r_i(t)$ 通过多项式插值显式编码位置和旋转的轨迹。这里，$b_{i,k}$ 和 $c_{i,k}$ 表示训练过程中优化的多项式系数，而时间中心 μ_i^τ则定义了对每个高斯轨迹进行建模所围绕的中心时刻。不透明度 $\sigma_i(t)$ 被表示为高斯形时间函数，由空间不透明度因子 σ_i^τ、时间缩放因子 s_i^τ和时间中心 μ_i^τ参数化。为简化模型并提高计算效率，受 [19]启发，我们将尺度 s_i 和颜色 c_i 均视为不随时间变化。

与依赖隐式运动建模（例如形变场）不同，后者会带来沉重的参数负担 [2, 14], ，我们采用的显式建模方法使用更少的参数，同时确保渲染结果相当。

4.2.3 参数共享

为减少参数冗余并提高模型紧凑性，我们采用了一种专门的参数共享策略。具体而言，在初始化期间，直接根据 SfM点云生成静态高斯基元。对于场景中的每个静态基元 G_i ，我们将时间多项式系数 $b_{i,k}$（$k=1,2,3$）、一阶旋转系数 $c_{i,}1$ 以及时间缩放因子 s_i^τ显式设为零，并将时间中心 μ_i^τ设为二分之一。因此，尽管静态基元在形式上与对应的动态基元共享相同的参数结构，但实际上会退化为时间不变基元。

由于这些参数在所有静态基元中都相同，因此在模型序列化期间无需逐个存储。由此，我们的参数共享策略显著减少了参数冗余，在保持表征精度和渲染效率的同时，大幅缩减了模型大小。

通过将静态基元的时间固定参数共享与动态基元的灵活时间建模相结合，我们的 SDD策略在保留表示复杂场景动态灵活性的同时，显著减少了参数量。静态基元的固定参数共享还消除了跨区域的时间干扰，从而保留静态内容中的高频细节和视觉保真度。

4.3 两阶段训练

尽管 SDD策略提供了紧凑高效的表示，但对静态和动态区域进行显式分离可能会在其边界附近引入伪影。为解决这一问题，我们提出了一种新颖的两阶段训练策略，专门用于实现更平滑的过渡，并增强静态-动态边界处的一致性。我们的方法促使动态基元更好地拟合并适应静态区域，从而显著减轻边界伪影，确保整体场景连贯性。

训练期间，我们将 HybridGaussians 分解为静态和动态基元。在优化过程中拆分或生成高斯基元时，它们会继承父基元的静态/动态标签，从而确保整个训练过程中的一致性。

在每个优化周期中，我们的两阶段训练流程如下：

4.3.1 阶段 1：静态优化

我们首先仅优化静态基元的参数。具体而言，我们渲染由上一轮优化得到的混合高斯，并计算渲染图像与真实图像之间的渲染损失。该损失仅用于更新静态高斯基元的参数，从而稳定静态表示，并细化场景几何结构与外观。

4.3.2 阶段 2：动态优化

随后，我们将阶段 1 中更新后的静态基元与先前优化得到的动态基元合并，形成更新后的混合高斯。再次渲染这些混合高斯后，我们计算其与真实图像之间的损失，并利用该损失仅更新动态高斯基元的参数。至关重要的是，在此阶段，动态基元会适应经过优化的静态表示，进一步提升静态-动态边界处的一致性。

与孤立地分别训练静态基元和动态基元相比，我们依次进行的两阶段训练方法能够有效缓解静态区域与动态区域交界处的边界不连续性和时间模糊伪影。尽管这种迭代优化会略微增加总训练时间，但它显著增强了静态基元与动态基元之间的无缝整合，从而实现对动态场景连贯且高质量的重建。

4.4 优化

在建立建模框架和两阶段训练策略后，我们现在介绍 HGS 参数的优化，该过程通过可微光栅化和基于梯度的优化来实现。由于框架中的静态基元和动态基元性质不同，它们的参数集也相应有所差异。具体而言，每个静态基元的优化参数包括 σ_i^τ、$b_{i,}0$、$c_{i,}0$、s_i 和 c_{i0}。对于每个动态基元，我们优化用于控制时间变化的参数，即 σ_i^τ、s_i^τ, μ_i^τ、多项式轨迹系数 $\{b_{i,k}\}_{k=0}^{3}$、旋转系数$\{c_{i,k}\}_{k=0}^{1}$，以及各向异性尺度向量 s_i 和颜色系数c_{i0}。

优化过程交替进行基于梯度的参数更新和高斯密度控制策略（如分裂和剪枝），确保模型在整个训练过程中兼具紧凑性与表达能力。

遵循先前方法 [16], ，我们将渲染损失定义为 L_1 光度项与结构相似性指数（D-SSIM）项的组合，该损失能够稳健地捕捉渲染图像与真实图像之间的视觉差异。给定渲染图像 I_{render} 及其对应的真实图像 I^{gt}，渲染损失 $\mathscr{L}$ 定义如下：

$$\mathscr{L}(I_{\text{render}}, I_{\text{gt}}) = \|I_{\text{render}} - I_{\text{gt}}\|_1 + \lambda \cdot \mathscr{L}_{\text{D-SSIM}}(I_{\text{render}}, I_{\text{gt}}), \quad (11)$$

其中 λ 用于平衡两项的贡献，$\mathscr{L}_{\text{D-SSIM}}$ 表示 D-SSIM 损失项。

训练期间，我们遵循所提出的两阶段优化策略（详见第4.3节），交替优化静态和动态基元，从而有效建模场景各组成部分之间的动态交互，并提升静态-动态边界处的一致性。

表 1：神经三维视频数据集上的定量比较。

方法	峰值信噪比↑	结构相似性指数↑	存储空间(MB)↓	训练时间↓	FPS↑
K-Planes [10]	31.39	0.944	311	-	0.3
MixVoxels [39]	30.55	0.924	508.07	1小时22分钟	16.7
HyperReel [1]	31.65	0.934	169.47	1小时41分钟	2.00
4DGS [41]	32.02	0.946	29.36	31m	114
ED3DGS [2]	32.35	0.949	31.41	1小时35分钟	75
STGS [21]	32.32	0.953	18.19	29m	140
SaRO-GS [43]	33.06	0.955	236.44	1小时31分钟	40
Ours	32.36	0.952	6.87	18m	300+

5 实验

5.1 实验设置

5.1.1 实现细节

我们使用 Track Anything 模型（TAM） [44] 对输入视频中的静态区域和动态区域进行显式分离。我们保存 TAM 生成的分割掩码，并利用它们为训练准备数据集。根据 [31],，我们使用不含颜色谐波分量的 3DGS，以进一步减少参数数量并实现高质量渲染。优化时，我们采用 Adam 优化器 [17]。除实时渲染在 RTX 3090 GPU（24GB）上进行外，所有实验和训练均在单块 NVIDIA RTX 4090 GPU（24GB）上完成。

5.1.2 数据集。

我们在两个公开数据集上对所提出的方法进行了定量和定性评估：神经三维视频数据集 [20] 和 Google沉浸式数据集 [4]。神经三维视频数据集使用 19–21 台摄像机采集， 2704×2028分辨率为 30 每秒帧数，包含具有复杂运动和多变物体外观的动态场景，给一致性重建带来了挑战。Google沉浸式数据集使用由 44–46 台摄像机组成的鱼眼相机阵列录制，包含室内和室外场景，且存在突如其来的过渡——例如大型火源突然出现——这进一步增加了多视图重建的难度。由于 Google沉浸式数据集本身具有较大挑战性，只有少数研究尝试在该数据集上进行比较评估。

5.1.3 对比方法。

我们将本方法与近期基于动态 NeRF 和高斯的方法进行比较，以评估其从多视角视频重建动态场景的效果。基于 NeRF 的基线方法包括 K-Planes [10], MixVoxels [39], HyperReel [1], 和 Nerf-Player [36], , 而基于高斯的基线方法包括 4DGS [41], ED3DGS [2], STGS [21] 和 SaRO-GS [43]。所有比较均采用官方实现，并保持原始超参数不变。

5.2 结果

5.2.1 定量比较

如表 1 所示，我们的方法峰值信噪比达到 32.36 dB，结构相似性指数达到 0.952，优于表现出色的基线方法 [1, 10, 39, 36, 41, 2] ，同时实现了与 STGS 相当的视觉质量 [21] ，并显著提升了效率。尽管渲染质量略逊于 SaRO-GS [43], ，我们的方法在模型紧凑性、训练效率和渲染速度方面具有显著优势。具体而言，我们的模型仅需 6.87 MB 存储空间，相较于 STGS 减少了 63% [21] ，相较于 MixVoxels 则最多减少了 98% [39]。这些结果证明，我们的设计能够在不牺牲视觉质量的前提下，实现紧凑而高效的动态场景表示。此外，我们的方法在所有测试场景中均表现出色，详情请参阅补充材料。

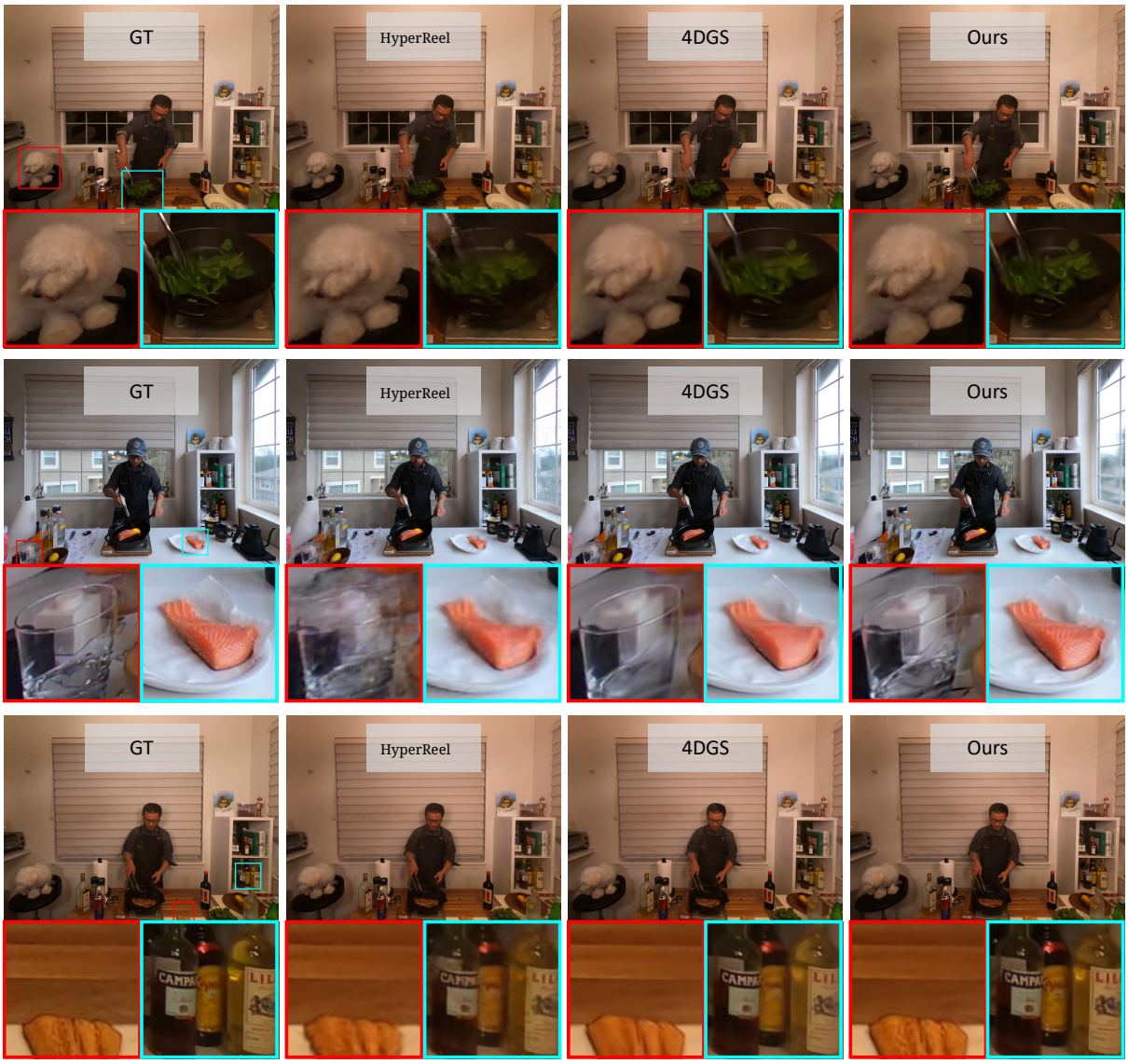

图 3：神经三维视频数据集上的定性结果对比。红色和蓝色区域突出显示了所提出方法在视觉质量方面取得显著改进的区域。

表 2：Google沉浸式数据集上的定量对比。

方法	峰值信噪比↑	结构相似性指数↑	存储空间（MB）↓	FPS↑
NeRFPlayer [36]	27.23	0.887	5130	0.12
HyperReel [1]	29.09	0.899	190.68	4
STGS [21]	28.87	0.943	19.74	99
Ours	29.60	0.925	12.73	300+

在计算效率方面，大量实验表明，HGS 具有卓越的实时性能，在分辨率为 1352×1014 时，使用单块 NVIDIA RTX 3090 GPU 可实现超过 300 FPS 的速度，远超 STGS [21]（140 FPS）和 ED3DGS [2]（75 FPS）。此外，HGS 仅用 18 分钟即可完成训练，显著提升了训练效率，远快于 SaRO-GS 等竞争方法 [43], ，后者需要 1 小时 31 分钟。这一高效流程进一步凸显了我们的方法在以下方面的实用性和可扩展性

真实世界中的动态场景应用。

表。2 总结了 Google沉浸式数据集上的定量结果。我们的场景具有突变的特点，我们的方法取得了具有竞争力的峰值信噪比（29.60 dB）和结构相似性指数（0.925），优于 HyperReel [1] 和 STGS [21]等基线方法。为减少突变产生的伪影，我们的方法补充了高斯，使模型大小略有增加。定性比较中提供了详细的视觉结果，以供进一步了解。

5.2.2 定性比较。

为了定性评估所提方法的视觉保真度，我们在前述两个数据集上进行了比较。在神经三维视频数据集上，我们展示了两个具有代表性的场景的重建结果，重点关注高频纹理区域的保真度。如图 3所示，与两种基线方法 [1, 41], 相比，我们的方法生成的重建结果明显更接近真实值，尤其能够保留精细纹理，例如肉表面的细节和复杂的

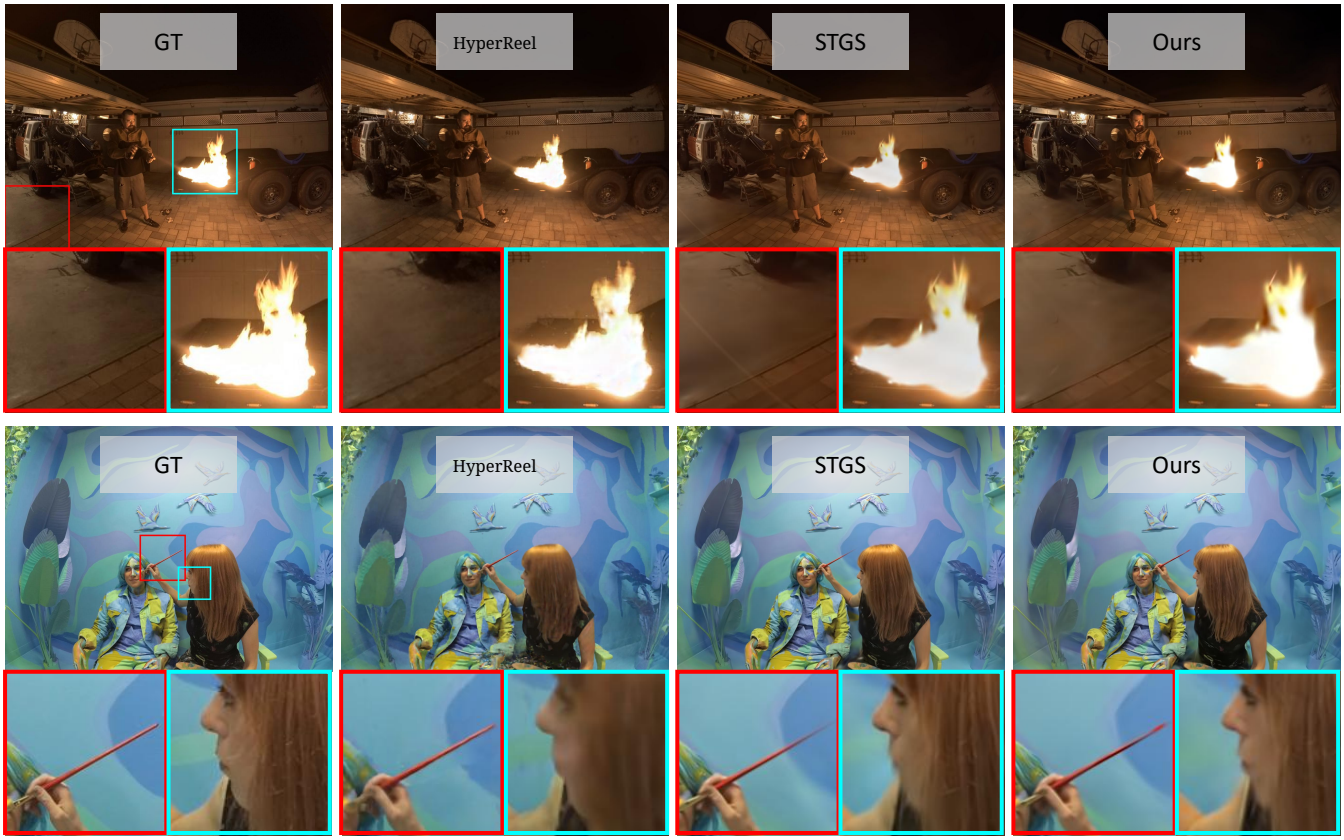

图4：Google沉浸式数据集上的定性结果对比。红色和蓝色区域突出显示了所提方法在视觉质量方面取得显著提升的区域。

表3：所提出框架的消融研究定量结果，报告为神经三维视频数据集上的平均值。

方法	峰值信噪比↑	结构相似性指数↑	存储空间(MB)↓
无初始化	32.5	0.954	13.51
无参数共享	32.22	0.951	11.56
无两阶段训练	25.76	0.910	5.88
Ours	32.36	0.952	6.87

蔬菜叶片分层。这证实了我们的方法能够有效解决静态区域中常见的高频细节缺失问题。

在 Google沉浸式数据集上，我们重点关注突发场景变化引发的伪影。如图4所示，当突然出现的火焰使场景变亮时，STGS [21] 无法准确捕捉亮度，导致静态区域出现伪影。在 STGS [21] 和 HyperReel [1]对运动物体进行重建时，也会出现类似问题。相比之下，我们的方法能够减轻这些伪影，应对突发的光照变化和内容变化。

总体而言，我们的定性分析表明，我们的方法能够保留高频纹理细节，并减轻场景突变带来的伪影，从而实现更优的视觉质量。有关与最先进方法的更多定性比较，请参阅附录。

5.3 消融研究

为探究我们框架中各个组件的有效性，我们进行了全面的消融研究，汇总结于表 3中。下面我们将介绍每个消融基线的配置和性能。

初始化：省略我们的交错帧初始化会使存储空间进一步增加至 13.51 MB。尽管峰值信噪比略有提升，但这一微小收益无法抵消模型复杂度的大幅增加。这表明，我们的初始化策略在渲染质量与模型紧凑性之间实现了最佳平衡。

参数共享：移除参数共享策略会使存储空间从 6.87 MB 增加至 11.56 MB，参数数量几乎翻倍，但视觉指标没有任何显著改善。这凸显了我们的参数共享策略在大幅减少冗余方面的有效性。

两阶段训练：移除我们的两阶段训练策略（即分别训练静态高斯和动态高斯，且不进行相互适应）会显著降低渲染质量，峰值信噪比从 32.36 dB 降至 25.76 dB，结构相似性指数则从 0.952 降至 0.910。这一下降凸显了静态区域与动态区域之间相互适应的关键作用；我们的两阶段策略促进了这种相互适应，从而增强了时间一致性并减少了边界伪影。尽管这种迭代优化会略微增大模型大小，但如图 7所示，其带来的显著视觉改进证明了这一代价是合理的。

5.3.1 径向基函数阶数

为说明我们选择三阶径向基函数的合理性，我们针对不同的多项式阶数进行了实验。结果汇总于表 4。结果表明，选择阶数为 3 的径向基函数能够实现最佳的整体平衡。与阶数 2 相比，它在不牺牲每秒帧数的前提下获得了更高的峰值信噪比，而存储空间仅略有



图5：静态和动态区域训练策略的定性结果。



图7：动态区域比例分析的定性结果。7.8%的设置对应通过标准静态-动态分解得到的动态区域，而95%的设置则通过扩张动态掩码生成，以人为扩大动态区域。



图7：消融研究的定性结果。

增加。与阶数 5 相比，它的峰值信噪比几乎相同，但存储空间需求显著更低，每秒帧数也更高。因此，阶数 3 在重建质量、效率和资源消耗之间实现了最实用的权衡，这也证明了将其作为我们方法默认选择的合理性。

5.3.2 静态和动态区域的训练策略变体

对于静态和动态区域的训练，有三种直观的设计：分别训练、共同训练（即更新静态和动态基元的参数.....

表4：在神经三维视频数据集上评估的不同径向基函数阶数的结果。

阶数	峰值信噪比↑	存储空间(MB)↓	训练时间↓	FPS↑
2	32.16	6.74	18m	300+
3（我们的方法）	32.36	6.87	18m	300+
5	32.44	7.61	18m	290

表5：我们在神经三维视频数据集上测试了三种训练方法的性能。

方法	峰值信噪比↑	存储空间（MB）↓	训练时间↓	FPS↑
分开训练	25.76	5.88	18m	300+
一起训练	32.33	12.01	18m	300+
两阶段（我们的方法）	32.36	6.87	18m	300+

.....），以及我们提出的交替两阶段策略。我们在这三种设置下进行了实验，以验证设计的有效性。结果如表 5 所示。首先，分别训练静态区域和动态区域会导致严重的伪影，质量也显著下降。结果还清楚地表明，同时训练静态和动态斑块效率低下：共同训练策略使存储成本几乎翻倍，却未能提升峰值信噪比，表现甚至略逊于两阶段方法。相比之下，两阶段策略在保持实时帧率的同时，以更少的存储空间实现了更高的峰值信噪比和更快的收敛。这表明，同时训练静态和动态斑块时，两者之间的梯度干扰会阻碍收敛，导致效率低下并可能造成过拟合。因此，采用两阶段训练方案是更有效且实用的解决方案。定性结果如图 5 所示。

表 6：通过逐步扩展动态掩码，评估动态区域占比增加对“煎牛排”场景的影响。

占比	峰值信噪比↑	LPIPS↓	存储空间（MB）↓	训练时间↓	FPS↑
7.8%（我们的方法）	33.34	0.032	4.77	18分31秒	149
20%	33.32	0.032	5.35	18分30秒	136
40%	33.44	0.032	6.18	19分06秒	130
60%	33.95	0.034	6.88	18分11秒	131
80%	33.61	0.033	7.73	19分07秒	131
95%	33.85	0.032	8.30	19分21秒	132

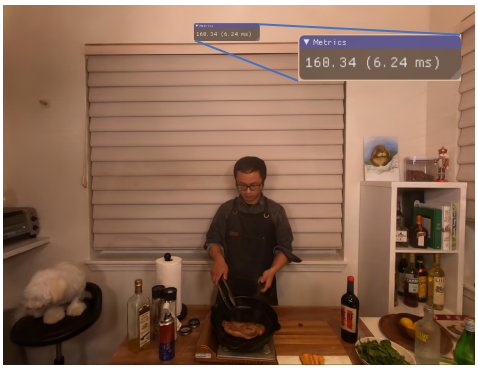

图8：在资源受限设备上的实时渲染性能，已在配备 RTX 3050 GPU 的笔记本电脑上进行验证。

5.4 动态区域比例对性能的影响

我们评估了动态区域占比对本方法性能的影响。为了进一步凸显传统显示器刷新率限制下的性能差异，我们提高了渲染分辨率，从而使帧率出现明显变化。从基线设置开始，此时本方法仅将场景中 7.8% 的部分识别为动态区域，我们逐步将这一比例提高至 95%，以对性能进行压力测试。如表 6 所示，在动态区域占比变化时，本方法仍能保持稳定的峰值信噪比和较低的 LPIPS，体现出可靠的定量表现。然而，将动态建模应用于本质上静态的区域时，细微的数值提升却以牺牲感知质量为代价：动态模糊和时间闪烁等伪影随之出现，在沉浸式观看过程中明显降低视觉稳定性。这些伪影不仅会增加存储需求，还会在实时渲染时分散用户注意力；在这种情况下，稳定且无伪影的感知远比客观指标的微小提升重要。图 6 中给出了代表性示例。更多演示内容请参见补充视频材料。

5.5 资源受限设备上的实时渲染

以往工作通常在高端 GPU 上评估效率，却忽视了其方法在资源受限环境中的广泛适用性。然而，实际上，大多数用户依赖中低端设备，而在这类设备上实现实时渲染要困难得多。为此，我们在配备 RTX 3050 GPU 的笔记本电脑上验证了我们的框架，实现了实时 1352×1014 渲染，如图 8 所示。这一结果凸显了我们方法的实际优势：即使在严格的硬件限制下，也能保持高效。遗憾的是，现有基线方法未提供实时渲染播放器，因此无法直接比较运行时间。尽管如此，在相同分辨率下，我们在 RTX 3050 上的渲染帧率已超过所有在 RTX 3090 上运行的现有方法，

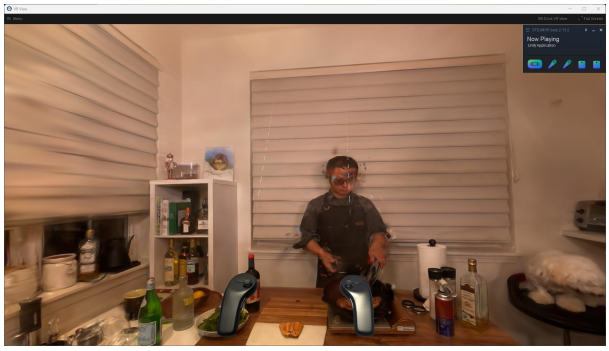

图9：VR 设备上的实时渲染性能。

这清楚地表明，我们的方法在资源受限设备上表现更优。

5.6 虚拟现实系统中的实时渲染

与传统二维渲染相比，虚拟现实渲染对计算能力提出了严格得多的要求，因为它需要更高的分辨率，并且要为双眼进行立体渲染。这些要求给渲染管线带来了沉重负担。现有的动态高斯方法由于模型大小较大、复杂度较高，难以满足此类约束；据我们所知，它们尚未展示出虚拟现实应用。

相比之下，我们改进后的表征显著降低了模型复杂度和计算成本，使虚拟现实条件下的实时渲染成为可能。如图 9所示，我们开发了专用的虚拟现实播放系统，可将动态三维高斯场景直接串流至头戴式显示器（HMD）。借助该系统，用户能够在沉浸式虚拟现实环境中交互式探索动态场景，突破二维显示器演示的局限。

Unity 与 OpenGL 坐标系之间的差异会导致渲染场景出现镜像伪影。此外，尽管我们对 3DGS 格式进行了必要的优化，以适配 Unity 渲染，但这些调整不可避免地引入了混叠，使得视觉保真度相比传统二维渲染略有降低。

6 局限性与未来方向

尽管 HGS 取得了显著改进，但仍存在若干尚待解决的挑战，可作为我们工作的自然延伸。

分割依赖。 我们的方法在一定程度上依赖静态—动态分离的准确性。现有方法的精度足以满足我们的需求；不过，我们还进行了额外测试，以验证框架的鲁棒性。我们明确评估了三种可能的错误情况：轻微的边界不准确（通过两阶段训练缓解，表 3，图 7）；静态区域被误标为动态区域（影响可忽略不计，表 6）；以及动态区域被误标为静态区域（峰值信噪比下降 0.8 dB，可通过我们的方法缓解，并可选择进行人工校正）。这些结果表明，HGS 能够抵御常见的分割错误，未来的工作可整合自适应分割优化。

时序运动建模。 作为一种局部的基于核的方法，显式的时序径向基函数建模在捕捉高度复杂的非刚性运动时可能面临挑战。为此，我们采用两阶段训练策略，使动态高斯能够随时间出现或消失；实践证明，这种方法行之有效。未来的扩展可将径向基函数与表达能力更强的形式相结合，以更好地捕捉细粒度形变。

训练策略。 我们的两阶段训练设计能够抑制边界伪影并提高收敛稳定性。尽管这会带来一定的额外计算开销，但这种权衡

有利于提升质量，而探索更精简的训练计划将进一步增强其在大规模或超高分辨率场景中的可扩展性。

7 结论

我们提出了混合高斯泼溅（HGS），一种用于动态场景重建的新型框架。通过在统一表示中将场景分解为静态和动态区域，HGS 减少了参数冗余并提高了计算效率。我们的静态与动态分解（SDD）策略为静态基元共享紧凑的时间不变参数，同时，两阶段训练策略提升了静态和动态区域边界处的一致性。实验表明，与最先进的方法相比，HGS 能够实现相当或更优的渲染质量，同时将模型大小最多减少 98%，并在 RTX 3090 上以每秒帧数 125 FPS 实现实时4K渲染。此外，它还能够在 RTX 3050 上以 1352×1014 保持 160 FPS，并已成功集成到 VR系统中，证明了其在沉浸式应用中的实用性。定性结果进一步表明，HGS 能够保留精细纹理并减轻突发场景变化造成的伪影，为高效、高保真的动态视图合成树立了新的基准。

致谢

作者谨向 A、B 和 C 致谢。本研究部分由 XYZ 提供的资助支持。

参考文献

[1] B. Attal、J.-B. Huang、C. Richardt、M. Zollhoefer、J. Kopf、M. O'Toole 和 C. Kim。Hyperreel：通过射线条件采样实现高保真六自由度视频。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 16610–16620 页，2023年 1、2、5、6、7–。[2] J. Bae、S. Kim、Y. Yun、H. Lee、G. Bang 和 Y. Uh。基于逐高斯嵌入的可变形三维高斯泼溅形变。载于《欧洲计算机视觉会议》，第 321–335 页。施普林格，2024 1, 2, 4, 5, 6。J. T. Barron、B. Mildenhall、M. Tancik、P. Hedman、R. Martin-Brualla 和 P. P. Srinivasan。Mip-nerf：用于抗混叠神经辐射场的多尺度表示。载于《IEEE/CVF国际计算机视觉会议论文集》，第 5855 5864, 2021 1 – 页。M. Broxton、J. Flynn、R. Overbeck、D. Erickson、P. Hedman、M. Duvall、J. Dourgarian、J. Busch、M. Whalen 和 P. Debevec。采用分层网格表示的沉浸式光场视频。《ACM Transactions on Graphics》(TOG), 39 4 86 1, 2020 5 ( ): –。A. Cao 和 J. Johnson。Hexplane：动态场景的快速表征。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 130 141, 2023 2 – 页。A. Chen、Z. Xu、A. Geiger、J. Yu 和 H. Su。Tensorf：张量辐射场。载于《欧洲计算机视觉会议》，第 333– 350 2022 1 页。施普林格，。Z. Chen、Z. Li、L. Song、L. Chen、J. Yu、J. Yuan 和 Y. Xu。Neurbf：采用自适应径向基函数的神经场表示。载于《IEEE/CVF国际计算机视觉会议论文集》，第 4182 4194, 2023 1 – 页。Y. Duan、F. Wei、Q. Dai、Y. He、W. Chen 和 B. Chen。4d-rotor高斯泼溅：面向动态场景高效新视角合成。载于《ACM SIGGRAPH 2024会议论文集》，第 1–11 页，2024年。[9] J. Fang、T. Yi、X. Wang、L. Xie、X. Zhang、W. Liu、M. Nießner 和 Q. Tian。采用时间感知神经体素的快速动态辐射场。载于《SIGGRAPH Asia 2022 1 9, 2022 2 会议论文集》，第 – 页。S. Fridovich-Keil、G. Meanti、F. R. Warburg、B. Recht 和 A. Kanazawa。K-planes：空间、时间与外观中的显式辐射场。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 12479–12488 页，2023年。1、2、5

[11] S. Fridovich-Keil、A. Yu、M. Tancik、Q. Chen、B. Recht 和 A. Kanazawa。Plenoxels：无需神经网络的辐射场。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 5501 5510, 2022 1 – 页。X. Guo、J. Sun、Y. Dai、G. Chen、X. Ye、X. Tan、E. Ding、Y. Zhang 和 J. Wang。用于动态场景新视角合成的前向光流。载于 IEEE/CVF国际计算机视觉会议论文集，第 16022 16033, 2023 2 – 页。Y.-H. Huang、Y.-T. Sun、Z. Yang、X. Lyu、Y.-P. Cao 和 X. Qi。Sc-gs：用于可编辑动态场景的稀疏控制高斯泼溅。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 4220 4230, 2024 1, 2 – 页。Y. Jeong、J. Lee、H. Choi 和 M. Cho。Rodygs：适用于随手拍摄视频的鲁棒动态高斯泼溅。arXiv预印本 arXiv:2412 03077，2024 年。[15] K. Katsumata、D. M. Vo 和 H. Nakayama。适用于单目/多视角动态场景的高效3D高斯表示。（无标题），2023年。1[16] B. Kerbl、G. Kopanas、T. Leimk¨uhler 和 G. Drettakis。用于实时辐射场渲染的3D高斯泼溅。ACM图形学汇刊，42 4 139 1, 2023 1, 2, 5 （）：–。D. P. Kingma 和 J. Ba。Adam：一种随机优化方法。arXiv预印本 arXiv: *1412 6980*, 2014 5 。A. Kratimenos、J. Lei 和 K. Daniilidis。Dynmf：结合3D高斯泼溅、用于实时动态视图合成的神经运动分解。载于欧洲计算机视觉会议论文集，第252–269页。施普林格，2024 1, 2 。J. C. Lee、D. Rho、X. Sun、J. H. Ko 和 E. Park。用于辐射场的紧凑型3D高斯表示。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第21719–21728, 2024 1, 4 页。T. Li、M. Slavcheva、M. Zollhoefer、S. Green、C. Lassner、C. Kim、T. Schmidt、S. Lovegrove、M. Goesele 等。基于多视角视频的神经3D视频合成。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 5521 5531, 2022 5 –页。Z. Li、Z. Chen、Z. Li 和 Y. Xu。用于实时动态视图合成的时空高斯特征泼溅。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 8508 8520, 2024 1, 2, 4, 5, 6, 7 –页。Z. Li、S. Niklaus、N. Snavely 和 O. Wang。用于动态场景时空视角合成的神经场景流场。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 6498 6508, 2021 2 –页。Y. Liang、N. Khan、Z. Li、T. Nguyen-Phuoc、D. Lanman、J. Tompkin 和 L. Xiao。Gaufre：用于实时动态新视角合成的高斯形变场。arXiv预印本 arXiv:2312.11458，2023年。1、2[24] Y. Lin、Z. Dai、S. Zhu 和 Y. Yao。Gaussian-flow：利用动态3D高斯粒子进行4D重建。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第21136– 21145, 2024 1, 2 页。L. Liu、J. Gu、K. Zaw Lin、T.-S. Chua 和 C. Theobalt。神经稀疏体素场。神经信息处理系统进展，33：15651–15663，2020年。1[26] T. Lu、M. Yu、L. Xu、Y. Xiangli、L. Wang、D. Lin 和 B. Dai。Scaffold-gs：用于视角自适应渲染的结构化3D高斯。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 20654 20664, 2024 1 –页。J. Luiten、G. Kopanas、B. Leibe 和 D. Ramanan。动态3D高斯：通过持久动态视角合成实现跟踪。载于2024年3D视觉国际会议（3DV），第800–809页。IEEE，2024年。2[28] B. Mildenhall、P. P. Srinivasan、M. Tancik、J. T. Barron、R. Ramamoorthi 和 R. Ng。NeRF：将场景表示为用于视角合成的神经辐射场。ACM通讯，65(1)：99–106，2021年。1、2[29] W. Morgenstern、F. Barthel、A. Hilsmann 和 P. Eisert。通过自组织高斯网格实现紧凑型3D场景表示。载于欧洲计算机视觉会议论文集，第18–34页。施普林格，2024年。1

[30] T. M¨uller、A. Evans、C. Schied 和 A. Keller。结合多分辨率哈希编码的即时神经图形基元。ACM图形学汇刊（TOG），41 4 1 15, 2022 1 （）：–。K. Navaneet、K. Pourahmadi Meibodi、S. Abbasi Koohpayegani 和 H. Pirsiavash。Compgs：通过矢量量化实现更小、更快的高斯泼溅。载于欧洲计算机视觉会议论文集，第 330 349 2024 5 –页。施普林格，。K. Park、U. Sinha、J. T. Barron、S. Bouaziz、D. B. Goldman、S. M. Seitz 和 R. Martin-Brualla。Nerfies：可变形神经辐射场。载于 IEEE/CVF国际计算机视觉会议论文集，第 5865 5874, 2021 2 –页。K. Park、U. Sinha、P. Hedman、J. T. Barron、S. Bouaziz、D. B. Goldman、R. Martin-Brualla 和 S. M. Seitz。Hypernerf：用于拓扑变化神经辐射场的高维表示。arXiv预印本 arXiv: *2106 13228*, 2021 2 。A. Pumarola、E. Corona、G. Pons-Moll 和 F. Moreno-Noguer。D-nerf：用于动态场景的神经辐射场。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 10318 10327, 2021 2 –页。R. Shao、Z. Zheng、H. Tu、B. Liu、H. Zhang 和 Y. Liu。Tensor4d：用于高保真动态重建与渲染的高效神经 4D分解。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 16632 16642, 2023 2 –页。L. Song、A. Chen、Z. Li、Z. Chen、L. Chen、J. Yuan、Y. Xu 和 A. Geiger。Nerfplayer：一种基于分解式神经辐射场、可流式传输的动态场景表示。IEEE计算机图形学与可视化汇刊，29 5 2732 2742, 2023 2, 5, 6 （）：–。C. Sun、M. Sun 和 H.-T. Chen。直接体素网格优化：辐射场重建的超快速收敛。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 5459 5469, 2022 1 –页。E. Tretschk、A. Tewari、V. Golyanik、M. Zollh¨ofer、C. Lassner 和 C. Theobalt。非刚性神经辐射场：从单目视频重建动态场景并合成新视角。载于 IEEE/CVF国际计算机视觉会议论文集，第 12959 12970, 2021 2 –页。F. Wang、S. Tan、X. Li、Z. Tian、Y. Song 和 H. Liu。用于快速多视角视频合成的混合神经体素。载于 IEEE/CVF国际计算机视觉会议论文集，第19706–19716页，2023年。1、2、5[40] L. Wang、Q. Hu、Q. He、Z. Wang、J. Yu、T. Tuytelaars、L. Xu 和 M. Wu。用于可流式自由视点视频的神经残差辐射场。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 76 87, 2023 2 –页。G. Wu、T. Yi、J. Fang、L. Xie、X. Zhang、W. Wei、W. Liu、Q. Tian 和 X. Wang。用于实时动态场景渲染的4D高斯泼溅。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 20310 20320, 2024 1, 2, 5, 6 –页。W. Xian、J.-B. Huang、J. Kopf 和 C. Kim。用于自由视点视频的时空神经辐照场。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第9421–9431, 2021 2 页。J. Yan、R. Peng、L. Tang 和 R. Wang。结合尺度感知残差场和自适应优化的4D高斯泼溅，用于时间复杂动态场景的实时渲染。载于第32届ACM多媒体国际会议论文集，第7880–7871页，2024年。1, 3, 6 [44] J. Yang、M. Gao、Z. Li、S. Gao、F. Wang 和 F. Zheng。Track anything：Segment anything与视频相结合。arXiv预印本 arXiv:2304 11968，2023年5月。[45] Z. Yang、X. Gao、W. Zhou、S. Jiao、Y. Zhang 和 X. Jin。用于高保真单目动态场景重建的可变形3D高斯。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 20331 20341, 2024 1, 2 –页。Z. Yang、H. Yang、Z. Pan 和 L. Zhang。利用4D高斯泼溅实现实时逼真动态场景表示与渲染。arXiv预印本 arXiv: *2310 10642*, 2023 1, 2 。R. Zhang、P. Isola、A. A. Efros、E. Shechtman 和 O. Wang。深层特征作为感知度量的惊人效用。载于 IEEE计算机视觉与模式识别会议论文集 识别，第 586–595 页，2018 年。2