

EntropyGS: 一种高效的三维高斯散点熵编码

Yuning Huang, Jiahao Pang, Fengqing Zhu, Dong Tian

Abstract

作为一种新兴的新颖视图合成方法，三维高斯点 (3DGS) 展现出快速的训练/渲染速度和卓越的视觉质量。3DGS 的两个任务，高斯创建和视图渲染，通常在时间或设备上分开的，因此存储/传输以及最终对 3DGS 高斯的压缩变得必要。我们首先对 3DGS 高斯属性进行了相关性和统计分析。此项工作中的一个启发性发现表明，球谐波 AC 属性严格遵循拉普拉斯分布，而高斯分布的混合可以近似旋转、缩放和不透明度。此外，谐波 AC 属性与除来自颜色空间的继承相关性之外的其他属性表现出弱相关性。随后提出了一种分解和参数化的熵编码方法，EntropyGS。在编码过程中，估计每个高斯属性的分布参数以辅助其熵编码。根据高斯属性类型，自适应地执行熵编码的量化。EntropyGS 在基准数据集上表现出约 30× 的码率减少，同时保持与输入 3DGS 数据相似的渲染质量，并且编码和解码速度很快。

1. 介绍

新视点合成 (NVS) 是一个计算机图形学任务，旨在生成用于虚拟现实、3D 重建和游戏等应用中的新视点图像。通过引入高斯作为基础元素，3D 高斯投影 (3DGS) [12] 有效地表示了 3D 场景的几何和纹理信息。一个高斯基础元素由位置、旋转、不透明度、缩放和球面谐波组成，以捕捉与视点相关的颜色。由于其高度并行化的设计和 GPU 技术的进步，3DGS 比神经辐射场 (NeRF) [17] 框架实现了显著更快的训练和渲染时间，同时提供了具有竞争力的视觉质量。然而，当训练和渲染在时间或设备上分离时，高斯基础元素的高数据速率对 3DGS 模型的传输/存储构成了显著的挑战。

对于高效地将 3DGS 编码成比特流的尝试，可以根据是否允许对 3DGS 的高斯进行离线更新来进行区分。对于那些可以离线更新高斯的应用，应用于高斯创建/更新和压缩的全局优化被采用 [5, 18, 21]。这提供了更多的压缩潜力，但需要额外的离线处理来更新高斯。对于那些低延迟是优先考虑或无法负担离线处理的应用，高斯原语作为预生成的 3DGS 模型，按照它们原有的形式被编码成比特流。因此，范围有限的专门用于压缩的研究有其自身的价值 [8]。这就是我们工作的定位方式。

许多神经网络 (NN) 模型压缩技术，例如模型参数



Figure 1. 我们提出的 EntropyGS 相较于 3DGS [12]，在保持高渲染质量的同时，实现了大约 30× 的显著内存减少。此外，对于 Mip-NeRF 360 数据集中诸如 “Room” 这类场景，其编码和解码时间维持在 5 秒以内 [4]。

量化、剪枝 [2, 9] 和知识蒸馏 [3, 11]，被用于 3DGS 压缩。然而，与我们的提议不同，在这些工作中，高斯属性（位置、旋转、球谐函数等）的物理意义以及显著的统计特性都没有得到很好的考虑。这些 3DGS 压缩工作与一般的 NN 模型压缩方法一致。

我们首先分析 3DGS 属性的统计特性。基于理论归纳和实证验证，我们揭示了球谐 AC 属性之间的弱相关性，但在色彩空间中存在继承的相关性。该发现为我们提出的分解熵编码设计提供了坚实的动机。

此外，对于每个属性，我们揭示它们的统计数据与一种常见的分布类型（例如，高斯分布或拉普拉斯分布）具有很强的 consistency。这导致了提出的熵编码依赖于统计估计，而不依赖于基于学习的方法。因此，我们提出了一种针对 3DGS 的熵编码方法，称为 EntropyGS，这是一个分解和参数化的设计。在文献中首次充分探索其统计一致性，这种方法是新颖且高效的。

我们工作的主要贡献总结如下：

1. A factorized design is proposed to entropy encode 3DGS based on a correlation analysis of Gaussian attributes that indicates a weak correlation in particular between spherical harmonic AC components.
2. A parameterized design is further proposed using common distributions to govern Gaussian attributes, including rotation, scaling, opacity, spherical harmonic AC.
3. An adaptive quantization finally completes the design of entropy encoder EntropyGS for pre-generated 3DGS models with competitive rate-

distortion performance.

对于不同的应用，根据是否允许离线更新高斯原语，现有的 3DGS 编码可以分为两类：联合优化与压缩，以及专门用于压缩的方法。

前一类的方法表现出更强的压缩能力，因为它们允许进一步的端到端优化。然而，后一类方法实现了低延迟，并且不改变 3DGS 表示。

1.1. 联合优化与压缩

对于这一类别的方法，3DGS 高斯生成过程通常通过将 3DGS 压缩所需的数据速率作为额外损失项进行更新。此外，他们可能通过与其他格式如 3D 网格 [6] 以及基于 MLP 的插值器 [21] 的结合来改变原始高斯表示。此类修改在场景编辑和交互式渲染等应用中应谨慎处理。

在 ScaffoldGS [21] 之上，HAC [5] 采用了一种基于锚点的表示方法，代替了原始的 3DGS 表示，以实现联合优化和压缩。它引入了一个压缩框架，通过引入二进制哈希网格和相关的基于上下文的建模来减少 Scaffold-GS 的内存占用。

SOG [18] 提出另一种改进的高斯表示。3D 高斯喷洒 (3DGS) 的参数被安排在一个二维网格上，然后设计了一种联合优化和压缩方法，以促进局部同质性。然而，3D 到 2D 映射所需的排序导致编码时间较长。

剪枝和后续优化 [1, 8, 22] 也是现有工作中常用的减少 3DGS 速度的技术。可以采用各种剪枝策略来决定哪些高斯基元需被剪除，例如，基于它们对渲染质量的影响。此类剪枝通常与 3DGS 高斯的生成保持一致，因此它们属于联合优化和压缩的研究范围。

对于低延迟是优先事项的情境，3DGS 模型的生成与其压缩必须分开进行。压缩特定研究旨在编码预先生成的 3DGS 模型。3DGS 模型的创建是压缩特定方法的上游和离线任务。

LightGaussian 中的一个代表性工作 [8] 引入了用于神经网络模型压缩的技术，包括矢量量化和知识蒸馏，以压缩原始的 3DGS 模型。C3DGS [19] 提出了具有量化感知训练的敏感性矢量聚类。MesonGS [22] 用 Euler 角替代旋转四元数，并应用区域自适应层次变换 (RAHT) [14] 来编码关键属性。熵编码是数据压缩的重要步骤，也是 3DGS 压缩的关键。HAC [5] 使用过拟合的 MLP 来估计分布参数，以进行 3DGS 模型的熵编码。

我们的工作属于高效熵编码的研究。与文献相比，我们的方法采用了基于识别出的分布模型的参数化设计。

2. 3DGS 属性的统计分析

本节分析了 3DGS 的属性，这启发我们提出一种有效的 3DGS 压缩方法。高斯属性在下文中被划分为六个不同的组：几何、旋转、缩放、不透明度、用于一阶球面谐波系数的 SHDC，以及用于高阶谐波系数的 SHAC。

我们将相关性分析分为两类：SHAC 系数内部的内部相关性和 SHAC 与其他属性之间的外部相关性，包括旋转、缩放、不透明度、SHDC。

对于 SHAC 通道内的相关性，球谐函数的正交性保证了每个分量（或系数）在构建时是独立的。对于通道间的相关性，SHAC 与其他属性之间的关系，由于其不同的物理意义，不应期望有强相关性。总之，数学上的独立性理论上导致通道内相关性为零，概念上的独立性导致通道间相关性可忽略不计。

在高斯创建过程中可能会引入较小的相关性。然而，在这项工作中，假设无论是内部相关性还是外部相关性都是较小的。

注意，颜色空间中的相关性将在相应的球谐系数之间传递。如何利用它们进行熵编码是未来的工作。

衡量相关性的指标：我们使用规范化互信息 (NMI) 作为评估相关性的指标。NMI 量化两个变量之间共享信息的量，因此能够捕捉一般相关性（包括线性和非线性）。

给定两个随机变量 X 和 Y 的概率质量函数（直方图），它们之间的归一化互信息 (NMI) 可以用以下公式计算：

$$\text{NMI}(X, Y) = \frac{2 \cdot I(X; Y)}{H(X) + H(Y)}$$

其中：

- $I(X; Y)$: X 和 Y 之间的互信息：

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}.$$

这里：

- $p(x, y)$: X 和 Y 的联合概率质量函数。
- $p(x)$: 边际概率质量函数 X , $p(x) = \sum_y p(x, y)$ 。
- $p(y)$: Y 和 $p(y) = \sum_x p(x, y)$ 的边际概率质量函数。
- $H(X)$: X 的熵定义为：

$$H(X) = - \sum_{x \in X} p(x) \log p(x)$$

- $H(Y)$: Y 的熵，类似地定义为：

$$H(Y) = - \sum_{y \in Y} p(y) \log p(y)$$

结果是一个在 0 到 1 之间的值：NMI(X, Y) = 1 表示完全相关，NMI(X, Y) = 0 表示没有相互相关性。

用于一个典型场景的标准化互信息热图如图 2 所示。从图的第一行可以观察到一个较弱的自相关性。图的第二行验证了 SHAC 与其他属性之间不存在跨相关性。请注意，所有测试场景都显示了相同的趋势，其他场景的完整结果请参见补充材料。

观察到的弱相关性为分解熵模型设计提供了坚实的支持，我们可以将每个高斯特征的通道独立处理。这样的设计表现出高效率、灵活性，而且可以轻松优化以适应并行计算平台，从而实现快速编码和解码时间。

在分析过程中，我们发现对于接近的高斯基元，高斯属性之间的相关性更强，这是合理的，因为可以预期

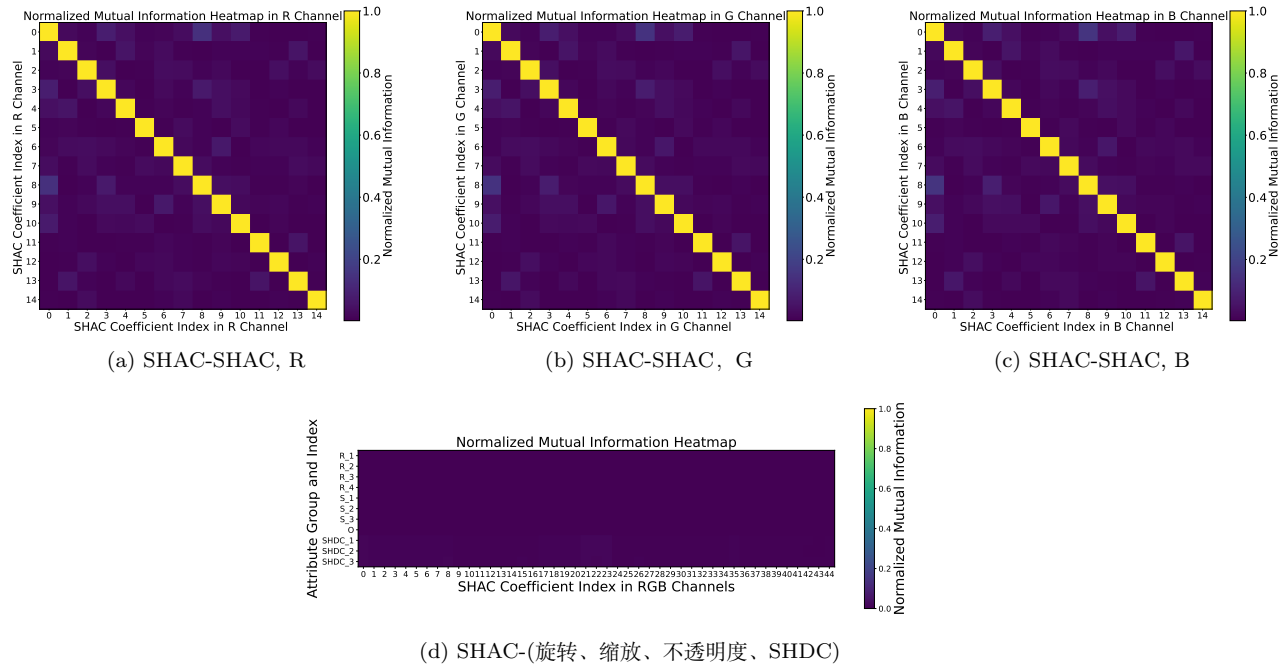


Figure 2. 来自 Mip-NeRF 360 数据集中“自行车”的互信息热图。第一行显示 SHAC 内部的相关性，第二行显示 SHAC 与其他高斯属性之间的相关性

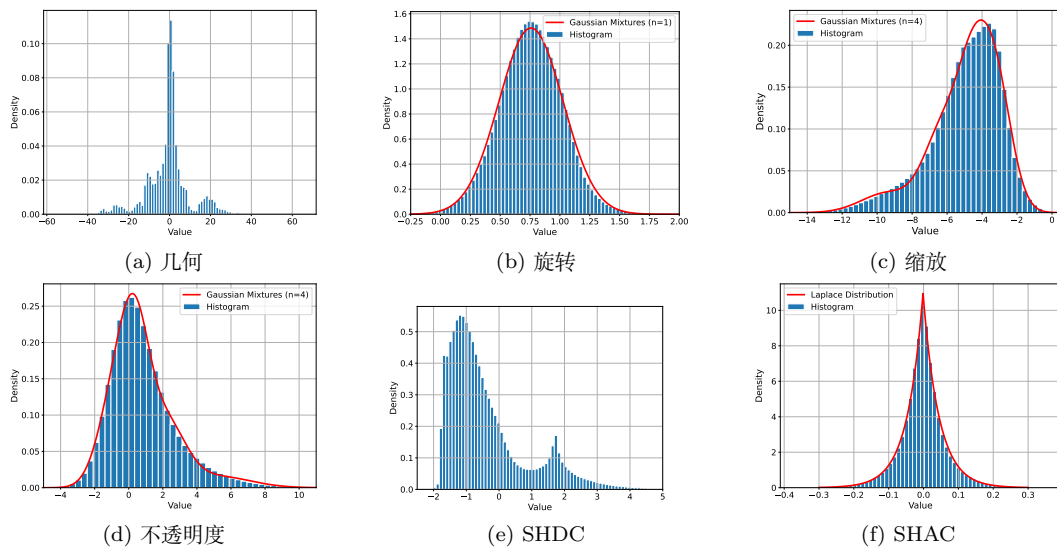


Figure 3. “Bicycle” 场景中 Gaussian 属性的示例直方图和估计分布（红色曲线），来自 Mip-NeRF 数据集。每个图都与相应属性组的一个通道相关。我们不估计 Geometry 和 SHDC 的分布。请注意，不同数据集中的其他场景也观察到了类似的统计行为。

局部相似性。然而，有效地结合这种相关性是具有挑战性的（需要 KNN）并且会破坏我们熵模型的因式设计；因此，我们认为这超出了讨论范围。

2.1. 3DGS 属性的控制分布

在此，我们分析了 3DGS 属性的统计行为，即几何、旋转、缩放、不透明度，以及球谐函数（SH）的直流和交流分量。在图 3 中展示了样本直方图和估计分布。除

了这个具体的例子，我们在基准数据集的所有场景中观察到了一致的统计行为。

对不同数据集的实证结果显示，每个属性组具有不同的统计特征：

- 几何：表现出高度动态的范围和稀疏性，与常用分布没有明显的对齐；
- 旋转/缩放/不透明度：通常遵循简单的高斯混合

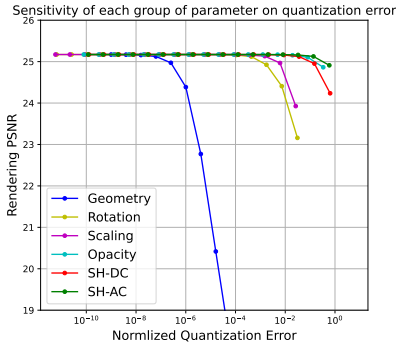


Figure 4. 灵敏度比较

模型，组成部分少于四个，且在许多情况下可以用单一高斯分布来近似；

- SHDC：不符合任何常见分布，并在两个极端表现出较小的峰值，对应于曝光过度或曝光不足的像素；
- SHAC：与拉普拉斯分布高度一致。

我们还进行了一项实证研究，以验证上述高斯属性与相关概率分布之间的一致性。具体来说，通过直方图计算量化高斯属性的香农熵 [20]，并将其与通过我们的估计实现的每样本实际比特数 (bps) 进行比较，如表格 1 所示。由于香农熵表示无损编码的理论最小码长，从表格 1 中可以看出，通过用估计分布对高斯属性进行参数化，可以实现非常低的编码开销。这个观察表明了常用分布用于描述高斯属性统计的近乎最佳的精准度。

Attribute	Q	Entropy	Actual	Overhead (%)
Rotation	8	6.243	6.261	0.29
Scaling	8	6.769	6.775	0.09
Opacity	8	6.337	6.342	0.08
SHAC	4	2.436	2.446	0.41

Table 1. 验证真实数据分布和估计分布之间的一致性。结果在 Mip-Nerf 360 上取平均，Q 表示量化深度（在第 4.3 节讨论）。

这些统计洞察是我们压缩方法的基础。策略很简单：对于旋转、缩放、不透明度和 SHAC，我们使用最大似然估计 (MLE) 或期望最大化 (E-M) 来估计它们的分布参数。对于几何和 SHDC，我们将它们与颜色属性结合为一个点云，然后应用现成的点云压缩方法。

2.2. 量化敏感性

我们的熵编码方法依赖于量化以获得离散表示。因此，研究每个属性组对量化误差的敏感性是至关重要的。请注意，本工作中使用的量化仅用于熵编码，这在信息论中有明确的定义。它不同于之前工作中提到的“量化”，后者表示神经网络参数和计算精度的降低。

我们通过分析相对于误差大小的渲染降级（见图 4），来研究量化对各种高斯属性的影响。归一化量化误差是通过将均方误差 (MSE) 按每个属性组的平均值进行缩放来计算的。我们的研究表明量化敏感性存在

显著的变化：几何属性最敏感，而 SHAC 属性表现出最小的敏感性。这一见解启发了一种自适应量化策略来优化速率失真权衡。该方法非常直接：对更敏感的属性组应用更精细的量化步骤，而对较不敏感的属性组则应用更粗糙的步骤。

此外，鉴于几何对敏感性的反应非常强烈，而 SHAC 则是最不敏感的，但却消耗最多的内存 (76.3 %)，这两个方面都需要额外关注。将在下一节中看到，对于几何，我们重点在于降低敏感性和最小化 PSNR 的降级。对于 SHAC，我们的目标是实现更小的量化深度以最大化内存节省。

3. 用于高斯散点编码的熵 GS

3.1. 编码流程概述

整体编码流程包括两个独立的阶段，如图 5 所示。

该流水线从准备阶段开始，该阶段与早期的方法共用。在这里，预生成的高斯模型通过统计指导的修剪和后续优化进行改进，以获得更易于压缩的高斯模型。

第二阶段，即我们主要贡献的 EntropyGS，是压缩过程的核心。EntropyGS 执行自适应量化和熵编码，利用统计分布估计。

需要强调的是，EntropyGS 被设计为一个独立的插件模块。它的操作不受制备阶段设计的限制，可以灵活地应用于预生成的 3DGS 模型，无论是否经过提前修剪和优化。

在准备阶段，通过修剪去除冗余的高斯，然后更新量化的高斯以补偿修剪导致的渲染质量损失。我们采用了 LightGaussian [8] 提出的修剪策略，并做了小的修改，具体内容将在下文中介紹。特别地，修剪分为两个步骤。

基于重要性的剪枝步骤直接应用了 LightGaussian [8] 中的方法，因为它简单且有效。一旦确定了每个高斯基元的重要性水平，我们将应用一个阈值 θ_1 （在 % 中）并根据该值移除不太重要的高斯。

基于第 2.2 节中的敏感性分析观察结果，我们提出了一种基于几何的剪枝策略，来消除难以编码且对渲染贡献很小的高斯分布。具体来说，基于以下策略剪去那些几何值远离场景中心的高斯分布：对于每个方向 (x, y, z) ，剪去高斯分布绝对值最大的前 θ_2 （在 % 中）。注意， θ_2 被故意设置得较小以避免移除潜在有意义的高斯分布。

修剪后，剩余的高斯数量为：

$$N_p = N_o \cdot (1 - \theta_1\%) \cdot (1 - \theta_2\%)^3 \quad (1)$$

，其中 N_o 表示初始高斯数量。

对于后修剪优化，我们首先利用 SHAC 的拉普拉斯分布特性来调整它们的值。这旨在缩小 SHAC 值的范围，从而减少后续最小-最大量化引入的误差。详情请参见补充材料。

接下来，为了补偿剪枝引起的质量损失，需进行一个额外的优化步骤，与原始 3DGS 训练相比，这大约增加了 20 % 的训练开销。此开销在离线处理时被认为

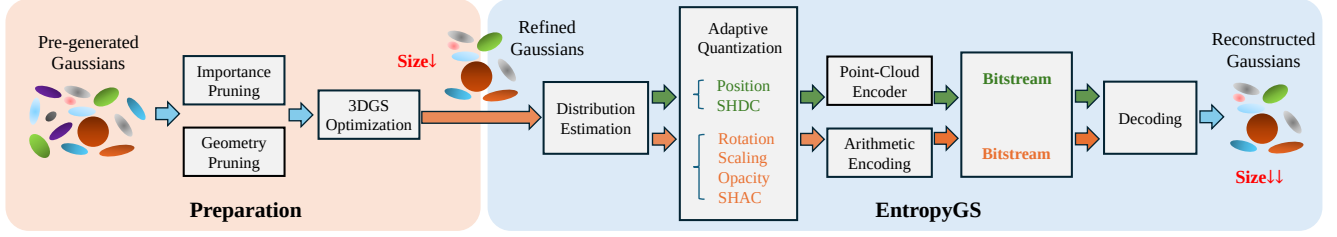


Figure 5. 我们压缩流程的概述。请注意，由于空间限制，完整的解码过程在图中用一个模块表示，解码过程与编码过程是对称的。第一阶段“准备”在流程中是可选的。如果实际应用中不允许额外优化，EntropyGS 阶段可以直接应用于预生成的高斯分布。

是可以接受的，特别是与像 HAC 这样的联合方法引入的 170 % 开销相比。

准备阶段主要用作测试条件，以确保在实验中被编码的高斯函数是有意义的，这是文献中的常见做法。如果不允许额外优化，EntropyGS 仍然可以应用于预生成的 3DGS。

在准备阶段之后，我们使用 EntropyGS 压缩优化的高斯分布，它能够高效地利用高斯属性的统计特性。

我们首先根据每组高斯属性的敏感性来量化它们的量化深度。对于每一组内的每个通道（例如，旋转组中的 4 个通道），量化是单独应用的。

应用于除 SHAC 之外的所有属性的量化方法是一个简单的最小-最大均匀量化：

$$q(x) = \text{round} \left(\frac{(x - v_{\min}) \cdot (L - 1)}{v_{\max} - v_{\min}} \right) \quad (2)$$

，其中 x 是待量化通道的一项， $v_{\min}$ 和 $v_{\max}$ 是该通道的最小值和最大值，而 L 是量化层级的数量。注意， $L = 2^Q$ 适用于特定的量化深度 Q 。

我们使用最多由四个成分组成的高斯混合模型来建模旋转、缩放和不透明度的分布。注意，这也对组内的每个信道单独进行。期望最大化（EM）算法 [7] 被用于以迭代方式估计每个成分的概率、均值和方差。当成分的数量 N 被限制为 4 时，EM 算法可以高效地执行。

Attributes	Quantization Depth		
	Ours-L	Ours-M	Ours-S
Geometry	17	16	15-16
Rotation	8	8	7
Scaling	8	8	7
Opacity	8	8	7
SHDC	8	8	8
SHAC	4-5	3-4	2-4

Table 2. 不同编码配置的量化深度。

对于 AC 系数，使用拉普拉斯分布来建模它们的值。我们采用最大似然估计（MLE）来确定拉普拉斯分布的参数：

$$\mu = \text{median}(x), \quad b = \frac{1}{N} \sum_{i=1}^N |x_i - \mu| \quad (3)$$

，其中 μ 是位置参数， b 是缩放参数。AC 系数的拉普拉斯性质是一个重要且有趣的发现。尽管超出了这项

工作的范围，我们假设这可能源于 3DGS 中使用的基于 L1 损失的优化 [12]。

在量化和分布估计之后，我们计算每个量化区间的概率以形成概率质量函数（PMF）。然后，这个 PMF 被诸如 Arithmetic Coding 算法 [16] 之类的算法用来实现无损压缩。

对于几何和 SHDC 组，我们避免拟合常用的分布。相反，我们使用点云编解码器对这些属性进行编码。具体来说，我们采用 MPEG PCC 标准中的高效 G-PCC 编解码器¹。它将具有几何和 SHDC 属性的高斯处理为一个彩色点云。请注意，基于学习的点云压缩器也可以直接应用于此处。

在我们的方法中，实现不同的速率（模型大小）是简单的，不需要额外的训练。只需调整应用于高斯属性的不同组的量化深度即可实现。

4. 实验

4.1. 实验设置

所提出的方法在 Mip-NeRF 360 [4]、Tank & Temples [13] 和 Deep Blending [10] 上进行了评估。

我们的方法与最近发表在顶级会议上的四种 3DGS 压缩方法进行了对比：LightGaussian [8]、C3DGS [19]、CompactGS [15]、MesonGS [22]。回顾第 ?? 节中对 3DGS 编码的分类，所选择的方法是专注于预生成的 3DGS 场景压缩的特定方法，而没有进行联合优化，因为这种情况下无法进行大量额外的训练。另一方面，HAC [5] 和 SOG [18] 作为联合方法，已经展示了出色的率失真性能。然而，这些联合方法不在我们的研究范围内，因此未被纳入直接比较（更多信息请参见补充材料）。

需要注意的是，上述不同的基线方法采用了不同的预生成 3DGS 模型，这导致预压缩指标的变化。例如，LightGaussian 报告在 Mip-NeRF 数据集上使用了 PSNR 为 27.53dB 的 3DGS，与原始 3DGS [12] 的 PSNR 为 27.28dB 相比。因此，为了确保公平比较，我们关注每个基线方法报告的压缩模型结果与未压缩模型结果之间的指标差异。例如，PSNR 差异定义为：

$$\Delta \text{PSNR} = \text{PSNR}_c - \text{PSNR}_o \quad (4)$$

¹<https://github.com/MPEGGroup/mpeg-pcc-tmc13>

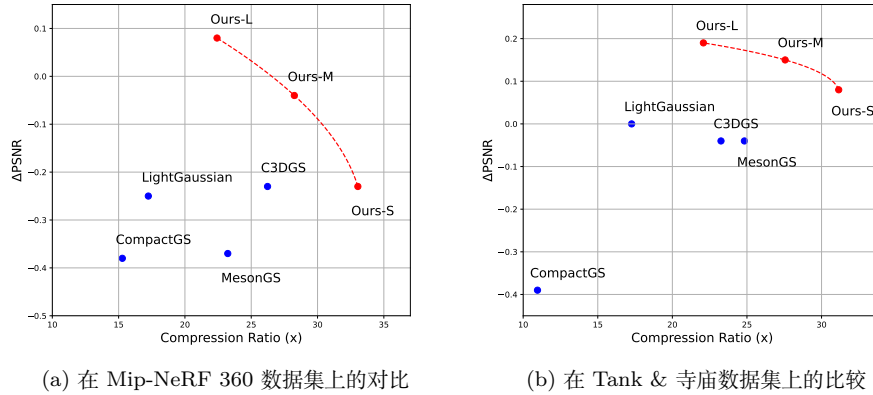


Figure 6. 在不同方法的数据集上的性能比较，右上方更好

其中 $PSNR_c$ 是压缩后的 PSNR, $PSNR_o$ 是压缩前的 PSNR。

PSNR (或 SSIM/LPIPS) 的差异表明由于压缩导致的渲染质量损失。它们的解释遵循原始指标的趋势：对于 $\Delta PSNR$ 和 $\Delta SSIM$ ，较高的值意味着更好的质量，而对于 $\Delta LPIPS$ ，较低的值意味着更好的质量。通常，这种差异（对于 PSNR/SSIM）是负的，代表压缩后的质量下降。然而，由于应用了额外的优化，这种差异可以是正的，意味着质量有所提升。

4.2. 实现细节

我们所有的实验均在 NVIDIA RTX 3090 和 Intel 11700K 上进行，使用的是来自 3DGS 论文 [12] 官方库的预生成模型，而不是在我们的平台上重新训练。就超参数设置而言，我们将场景分为室内和室外两组。对每组内的所有场景应用相同的处理策略，展示了我们方法的适应性，即不需要特定场景的配置。有关剪枝和优化阶段的设置，请参见补充材料。

基于所描述的码率控制方法，我们引入了 EntropyGS 的三种编码配置：EntropyGS-L、EntropyGS-M 和 EntropyGS-S，分别对应于大、中、小位流大小。详见 Tab. 2。所有配置均应用于准备阶段输出的相同 3DGS。

4.3. 主要结果

在表 3 和图 6 中，我们评估了我们的方法以及其他 3DGS 压缩方法，注意在表中，我们仅显示 EntropyGS-M 的结果以方便比较。实验是在 Mip-NeRF 和 Tank & Temples 数据集上进行的，因为某些基线方法没有报告 Deep Blending 数据集的结果。

我们的方法在所有数据集上实现了显著更小的模型尺寸，有效减少了存储需求。尽管进行了这样的压缩，EntropyGS 仍然保持了相当或更高的 PSNR 值，尤其是在大规模配置中，视觉质量的下降极小。速率控制方法的灵活性使得能够在压缩和渲染质量之间进行权衡，使得我们的方法对于需要灵活性的应用来说是理想的

选择。

与这些基线的比较表明，EntropyGS 在这两个基准测试中始终表现优于它们。例如，它在所有基线方法中均获得更高的 $\Delta PSNR$ 和压缩率。EntropyGS 的竞争性表现证明了我们提案的有效性，该提案结合了分布估计、自适应量化和熵编码，并由 3DGS 属性的分析指导。

此外，EntropyGS 以精简、低复杂度的设计实现了具有竞争力的性能，不像基线方法需要可训练模块，比如用于矢量量化的代码本或用于概率估计的神经网络。这个精简的设计为渲染质量和压缩比之间的取舍提供了灵活且透明的控制，并且可以很容易地针对特定的硬件配置进行优化。

在编码和解码复杂度方面，值得注意的是，仅有两种方法，C3DGS (5 分钟) 和 MesonGS (1 分钟)，在他们各自的 GPU 平台论文中报告了编码时间。我们在表格 4 中报告了编码和解码时间。如预期，我们的方法表现出显著更快的编码和解码时间，这归功于分解的熵模型设计和无学习的统计分布估计。

4.4. 消融研究

在本节中，我们将研究该方法中提出的每个步骤的有效性以及它们如何对最终性能做出贡献（见表 5），在此实验中，使用了 EntropyGS-M 设置。

3DGS 原语的修剪最初会减少模型大小，但会导致一些渲染质量损失。然而，随后的后期优化能够有效地恢复质量，甚至略微改善 PSNR。我们提出的几何修剪对即时影响最小，因为只有一小部分高斯被修剪，但这一步有助于在后续阶段降低量化误差。

将自适应量化应用于 3DGS 内的不同属性组进一步减小了模型大小，但由于量化误差引入了一些渲染质量的下降。接下来的熵编码以无损的方式进一步压缩量化后的数值。熵编码未应用于几何或 SHDC 属性。相反，我们使用标准的 G-PCC 点云编解码器以无损模式编码这些组件，从而实现最小的模型大小。值得注意的是，单凭量化、熵编码和点云编码的组合就能实现

Method	Mip-NeRF 360				Tank & Temples			
	Δ PSNR $\uparrow$	Δ SSIM $\uparrow$	Δ LPIPS $\downarrow$	Ratio* $\uparrow$	Δ PSNR $\uparrow$	Δ SSIM $\uparrow$	Δ LPIPS $\downarrow$	Ratio* $\uparrow$
3DGS [12]	-	-	-	-	-	-	-	-
LightGaussian [8]	-0.25	-0.005	0.016	17.24 $\times$	0	-0.005	0.012	17.27 $\times$
C3DGS [19]	<u>-0.23</u>	-0.014	0.024	<u>26.23 $\times$</u>	-0.04	-0.009	<u>0.011</u>	23.26 $\times$
MesonGS [22]	-0.37	-0.009	<u>0.013</u>	23.23 $\times$	-0.04	-0.001	0.006	<u>24.83 $\times$</u>
CompactGS [15]	-0.38	-0.014	0.025	15.28 $\times$	-0.39	-0.014	0.023	10.96 $\times$
EntropyGS (Ours)-M	-0.04	<u>-0.007</u>	0.012	28.25 $\times$	0.15	<u>-0.005</u>	0.013	27.56 $\times$

Table 3. 在 Mip-NeRF 360 和 Tank & Temples 数据集上对比方法。在每个指标中，最佳结果用粗体显示，第二好的结果是 underlined。我们的方法在提供最高压缩比的同时，实现了显著更高的 PSNR。* 比率表示压缩比（原始模型大小除以压缩后模型大小）。

Method	CPU coding time		GPU coding time	
	Encoding	Decoding	Encoding	Decoding
EntropyGS (Ours)-M	16.4s	13.8s	3.6s	0.2s

Table 4. 在 Mip-NeRF 360 上的 CPU 和 GPU 编码时间的平均值

Method	PSNR	Size (MB)
Original 3DGS	27.28	760.1
+ Importance Pruning	25.54	266.0
+ Geometry Pruning	25.45	264.4
+ Optimization	27.42	264.4
+ Quantization	27.24	41.5
+ Entropy Coding	27.24	32.2
+ G-PCC	27.24	26.9

Table 5. 消融研究：在 Mip-NeRF360 上测试，管道中的每一步如何影响 3DGS 的渲染质量和大小

10 倍压缩比，而无需任何额外的训练（例如，之前提到的准备阶段）。这突显了我们提出的 EntropyGS 的实际优势，它在不需要进一步优化的情况下，将 3DGS 模型大小减少了 10 倍，同时实现了快速编码和解码。

5. 结论

以对 3D 高斯点推进属性的统计特征的新颖见解为动机，我们引入了一种简单但有效的压缩方案，专门用于编码 3D 高斯点推进模型。实验结果表明，我们的方法在实现竞争性压缩性能的同时，依然保持简便和计算效率高。值得注意的是，EntropyGS 采用了一个分解熵模型，独立地编码每个属性通道。该设计在简化的概率估计、减少计算开销和增强可扩展性方面具有优势。虽然可以通过考虑除 SHAC 之外的属性之间的潜在关系来进一步提高率失真性能，但有效地整合这些关系是一个复杂的挑战，我们将其留待未来的工作中解决。

References

[1] Muhammad Salman Ali, Sung-Ho Bae Maryam Qamar, and Enzo Tartaglione. Trimming the fat: Efficient compression of 3d gaussian splats through pruning. arXiv preprint, 2024.

[2] Sajid Anwar, , and Wonyong Sung Kyuyeon Hwang. Structured pruning of deep convolutional neural networks. ACM Journal on Emerging Technologies in Computing Systems, 2017.

[3] Jimmy Ba, , and Rich Caruana. Do deep nets really need to be deep. Advances in neural information processing systems, 2014.

[4] Jonathan T. Barron, Dor Verbin Ben Mildenhall, , and Peter Hedman Pratul P. Srinivasan. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022.

[5] Yihang Chen, Weiyao Lin Qianyi Wu, , and Jianfei Cai Mehrtash Harandi. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In European Conference on Computer Vision, 2025.

[6] Yonghan Lee Jaehoon Choi, Heesung Kwon Hyungtae Lee, and Dinesh Manocha. Adaptive mesh-aligned gaussian splatting for high-quality rendering. Asian Conference on Computer Vision, 2024.

[7] Arthur P. Dempster, , and Donald B. Rubin. Nan M. Laird. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society, 1977.

[8] Zhiwen Fan, Kairun Wen Kevin Wang, Dejia Xu Zehao Zhu, , and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. Advances in Neural Information Processing Systems, 2024.

[9] Yang He, Xuanyi Dong Guoliang Kang, , and Yi Yang Yanwei Fu. Soft filter pruning for accelerating deep convolutional neural networks. arXiv preprint, 2018.

[10] Peter Hedman, True Price Julien Philip, George Drettakis Jan-Michael Frahm, , and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics, 2018.

[11] Geoffrey Hinton. Distilling the knowledge in a neural network. arXiv preprint, 2015.

[12] Bernhard Kerbl, Thomas Leimkühler Georgios Kopanas, , and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics, 2023.

[13] Arno Knapitsch, Qian-Yi Zhou Jaesik Park, , and Vladlen Koltun. Tanks and temples: Benchmarking

- large-scale scene reconstruction. *ACM Transactions on Graphics*, 2017.
- [14] Ricardo L, Chou De Queiroz, and Philip A. Compression of 3d point clouds using a region-adaptive hierarchical transform. *IEEE Transactions on Image Processing*, 25(8):3947–3956, 2016.
 - [15] Joo Chan Lee, Xiangyu Sun Daniel Rho, , and Eunbyung Park Jong Hwan Ko. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.
 - [16] Fabian Mentzer, Michael Tschannen Eirikur Agustsson, , and Luc Van Gool. Radu Timofte. Practical full resolution learned lossless image compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.
 - [17] Ben Mildenhall, Matthew Tancik Pratul P. Srinivasan, Ravi Ramamoorthi Jonathan T. Barron, , and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021.
 - [18] Wieland Morgenstern, Florian Barthel, Anna Hilsman, and Peter Eisert. Compact 3d scene representation via self-organizing gaussian grids. In *European Conference on Computer Vision*, 2025.
 - [19] Simon Niedermayr, , and Rüdiger Westermann Josef Stumpfegger. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.
 - [20] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 1948.
 - [21] Tao, Yu Lu, Linning Mulin, Xu, Yuanbo, Wang, Xiangli, Dahua, Dai Bo Limin, and Lin. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.
 - [22] Shuzhao Xie, Chen Tang Weixiang Zhang, Rongwei Lu Yunpeng Bai, , and Zhi Wang Shijia Ge. Mesongs: Post-training compression of 3d gaussians via efficient attribute transformatio. In *European Conference on Computer Vision*, 2025.