HPC：面向流式动态高斯泼溅压缩的分层基于点的潜在表示

Yangzhi Ma、Bojun Liu、Wenting Liao、Dong Liu，*IEEE* 高级会员
Zhu Li，*IEEE* 高级会员，以及 Li Li†，*IEEE* 高级会员，

arXiv:2602.00671v1 [cs.CV] 31 Jan 2026

摘要——**动态高斯泼溅推动自由视点视频取得了显著进展，但如何在保持渲染质量的同时降低内存占用，以实现高效的流式传输，仍是一项持续面临的挑战。现有的流式动态高斯泼溅压缩方法通常利用潜在表示驱动神经网络，预测帧间的高斯残差。其核心潜在表示可分为基于结构化网格和基于非结构化点的两种范式。然而，前者不可避免地对未占用空间进行建模，造成显著的参数冗余；后者则因未能利用局部相关性而缺乏紧凑性。为缓解这些局限，我们提出了 HPC，这是一种新颖的流式动态高斯泼溅压缩框架。该框架采用逐高斯运作的分层基于点的潜在表示，以避免未占用空间中的参数冗余。在专门设计的聚合方案的引导下，这些潜在点实现了高紧凑性和低空间冗余。为提高压缩效率，我们进一步首次探索通过挖掘并利用参数的帧间相关性，对用于流式动态高斯泼溅的神经网络进行压缩。结合潜在表示压缩，这构成了一个完全端到端的压缩框架。全面的实验评估表明，HPC 的性能显著优于最先进方法。与基线相比，它将存储量减少了 67%，同时保持了较高的重建保真度。**

索引词—**动态高斯泼溅压缩、自由视点视频、潜在表示**

I. 引言

随着沉浸式媒体的出现，能够从任意视角实时观看的自由视点视频（FVV）技术，在推动下一代媒体系统演进方面发挥了关键作用。高效、高质量的自由视点视频框架有望催生多种应用，例如虚拟现实（VR）和增强现实（AR）。为此，已有工作研究了多种场景表示，包括网格、点云和辐射场。

在这些候选方案中，近来兴起的三维高斯泼溅（3DGS）[1] 正开辟出大量新方向可用于自由视点视频应用。这种表示擅长处理复杂视觉内容，同时支持实时渲染。为满足流媒体和沉浸式直播等实时视频应用对低延迟的需求，开创性研究将 3DGS 扩展为针对在线场景优化的动态框架 [2]–[5]。这些框架逐帧增量优化帧间形变，从而实现即时重建。然而，尽管动态高斯泼溅表示具有显著优势，其庞大的数据规模仍给流式传输带来了巨大挑战。因此，将压缩技术融入动态高斯泼溅已成为一项重要研究方向。

在压缩领域，一种常见的动态高斯处理策略是采用潜变量表示，并结合若干神经网络进行预测 [6]–[9]。潜变量表示与神经网络经过联合优化、压缩和传输。在这一框架下，潜变量表示的设计成为核心挑战之一。受早期辐射场研究中成功经验的启发，结构化潜在网格 [10], [11] 被引入高斯泼溅 [2], [6]–[8], [12]。在这种表示中，可学习潜在嵌入被分配到结构化网格的顶点上，以生成覆盖整个空间的潜在场。在推理过程中，通过查询各个高斯的位置，获取其对应的潜在嵌入。然而，在整个空间分配潜在嵌入会导致参数数量激增，使其难以直接优化和存储。因此，这类基于网格的方法通常采用低分辨率网格 [6] 或冲突哈希 [2], [7], [8] 来减少参数，但这不可避免地会损害性能。

或者，鉴于高斯的位置构成离散集合，仅占据三维空间中极小的一部分 [13]，直接将潜在嵌入附加到每个点的位置上、忽略未占据区域，是一种自然的做法。然而，现有的基于点的潜在表示 [9]主要关注孤立的点，未能对局部邻域内的相关性建模，因此造成了空间冗余。此外，鉴于高斯呈非均匀分布，其单尺度设计无法捕捉多样的空间特征，导致性能欠佳。

除了潜在表示方面的挑战之外，另一个关键问题在于压缩策略的不足。现有

Y. Ma、B. Liu、W. Liao、D. Liu和L. Li就职于脑启发智能感知与认知教育部重点实验室，中国科学技术大学，合肥230027，中国（电子邮箱：mayz@mail.ustc.edu.cn;liubj@mail.ustc.edu.cn;liaowenting@mail.ustc.edu.cn; dongeliu@ustc.edu.cn; lil1@ustc.edu.cn）。Z. Li就职于密苏里大学堪萨斯城分校。（电子邮箱：lizhu@umkc.edu）

† 表示通讯作者。

研究 [6]–[8]致力于通过率感知优化和熵编码压缩潜在表示，但忽视了对神经网络参数的压缩。这一局限不仅阻碍了进一步降低比特率，还导致潜在表示与网络参数之间的码率分配失衡。因此，亟需一种集成框架，对潜在表示和神经网络参数进行联合优化与压缩。

针对这些局限性，我们提出 HPC，这是一种新颖的流式动态高斯泼溅框架，将**分层点**式潜在表示与完全端到端的**压缩**策略相结合。**一方面**，我们的潜在表示基于离散点构建，并以单个高斯为单位进行处理，从而避免未占用空间中的参数冗余。为适应高斯的非均匀分布，我们通过渐进式下采样为该表示配备多尺度感受野，从而形成潜在层级。我们进一步设计了一种兼顾尺度内处理与跨尺度处理的聚合方案，以减少空间冗余。该方案促进了多个分辨率下的局部信息共享，从而实现紧凑表示。**另一方面**，我们首次探索了流式动态高斯泼溅中的神经网络压缩。通过分析参数分布，我们发现相邻帧之间可能存在时间冗余。基于这一观察，我们提出一种时间参考策略，利用最新解码的参数来指导当前帧参数的优化与压缩。该策略使我们能够对能量紧凑的帧间残差进行编码，而非编码完整参数，从而显著降低比特率消耗。结合潜在表示压缩后，我们的框架执行完全端到端的优化，在潜在参数与网络参数之间平衡比特率，从而取得优异的率失真性能。

综上所述，所提出的HPC具有以下主要贡献：

- HPC框架引入了一种新颖的分层潜在点表示，以匹配高斯的非均匀分布，同时保持参数效率。通过在尺度内和跨尺度运行的专用聚合机制，有效降低了空间冗余，促进了局部信息共享，从而得到紧凑表示。

- HPC率先利用帧间时间冗余，对流式动态高斯泼溅中的网络参数进行压缩。由此得到的时间参考策略对紧凑残差进行编码，并在我们的全端到端框架中与潜在表示压缩相结合，从而实现最优比特率分配。

- 大量实验结果表明，HPC相对于最先进方法取得了显著的率失真性能。全面的消融实验与分析证实了每个所提组件的有效性。

本文其余部分的结构如下。第二节回顾相关领域的相关工作。第三节介绍我们框架的预备知识。第四节和第五节详细阐述所提出的 HPC 方法。第六节介绍并分析实验设置与结果。最后，第七节总结全文。

II. 相关工作

A. 静态高斯泼溅压缩

为解决 3DGS 数据占用量庞大的问题，高斯泼溅压缩致力于消除高斯表示中的冗余，同时保持渲染保真度。本节概述静态高斯泼溅压缩，涵盖两大主要分支：训练后压缩和生成式压缩。

训练后压缩将压缩过程与 3DGS 训练分离 [14]–[19]。它直接对预训练的 3DGS 模型进行操作，无需额外的优化开销，即可实现快速压缩流程。现有训练后方法通常利用成熟的变换 [14] 或编码工具 [15], [20]。

为实现最优的率失真权衡，生成式压缩方法会联合训练压缩模型和 3DGS 表示 [17], [21]–[32]。一些方法采用剪枝 [21],[22] 或掩码 [23], [24] 技术，剔除不重要的元素，直接减少高斯的数量。另一些方法则应用矢量量化 [23], [24], [26],[27]，以减少属性级冗余。通过利用 3DGS 的局部空间冗余，Scaffold-GS [28]构建了更高效的表示：将一片区域内的高斯聚类到少数锚点上，并采用隐式表示，从而大幅减少存储量。在 Scaffold-GS 的基础上，若干研究 [29]–[32] 更进一步，引入专用熵模型和端到端率失真优化，以实现最佳性能。

B. 动态高斯泼溅表示

动态高斯泼溅将 3DGS 扩展到时间域，以满足自由视点视频应用的需求。为适应不同的应用需求，主流表示大致可分为两种范式：离线优化和在线优化。前者以整个视频序列或多帧作为训练和传输的基本单位，因此能够利用更丰富的时间上下文作为参考。后者则以单帧为基本单位进行处理。该方法逐帧增量式地建模帧间变化，因此能够自然地支持流媒体应用。

对于离线方法，一种范式是将高斯扩展到四维时空域，其中每个高斯都能捕获空间和时间上的局部区域 [33]–[38]。另一类方法则基本上将变形应用于规范空间中的高斯 [39]–[43]。然而，在需要流式传输或涉及长视频序列的场景中，这些方法存在局限性。

相比之下，在线方法以迭代方式处理四维场景，逐帧适应变化，以

支持流式传输。在线动态高斯的表示主要分为两种范式：结构化网格和非结构化点表示。沿用前一种范式，3DGStream [2] 利用哈希网格 [10]隐式建模帧间运动。类似地，4DGC [6]采用潜在网格进行运动预测，并辅以补偿策略以减少信息损失。iFVC [7] 基于 Scaffold-GS [28]提出了一种二值三平面，用于高效地进行残差预测。然而，由于在整个空间中分配特征会造成难以承受的内存占用，这类表示天然受限于哈希冲突 [2], [7] 或较低的分辨率 [6],，因而表达能力欠佳。

在点表示范式中，参数直接附加到每个高斯的位置上，从而消除了未占用区域所造成的参数浪费。遵循这一原则，HiCoM [3]优化了层级一致运动，以实现高效预测。作为 HiCoM 的扩展，ReCon-GS [4]引入了动态层级重配置策略以提升性能。ComGS [5] 结合关键点运动模型与关键帧策略，实现了高效的动态重建。尽管这些方法训练速度快，但其表示紧凑性有限，因为它们直接在原始参数空间中优化和处理运动。另一种途径则是优化潜在表示，依靠神经网络学习对运动或残差的预测。在这一范式下，QUEEN [9] 为每个高斯分配一个潜变量，并采用联合优化的潜在解码器进行残差推断。然而，其逐点建模方案无法捕捉局部邻域内的相关性，导致空间冗余。此外，其单尺度设计不适用于非均匀分布的高斯，无法自适应地捕捉空间特征。

C. 动态高斯泼溅与隐式神经视频表示的压缩

与其表示方式的分类类似，动态高斯泼溅的压缩也可分为两大类：离线和在线。在离线方法中，MEGA [37] 通过解耦颜色属性并采用熵约束形变来实现压缩。GIFStream [42] 和 4DGV [43]则利用成熟的视频编解码器实现高效压缩。

关于在线压缩方法，QUEEN [9] 量化潜在表示，随后通过学习式门控模块稀疏化位置残差，以实现存储节省。iFVC [7] 将其三平面潜在嵌入量化为二进制，并利用经验频率分布进行熵估计，从而促进端到端率失真优化。4DGC [6]对其运动网格和补偿高斯均采用因子化熵模型 [44]。在此基础上，4DGCPro [8] 专门实现了渐进式流式传输和实时编码能力。现有方法主要侧重于压缩潜在表示。然而，一个关键疏忽在于，它们均未联合考虑神经网络参数。这一疏忽不仅阻碍了进一步的码率降低，还导致了低效的比特分配。

为弥合这一差距，我们可以借鉴隐式神经视频表示领域的相关见解 [45]–[50],。在该领域，压缩神经网络本身一直是研究重点，并已形成成熟的策略。作为该领域的先驱，NeRV [45] 直接对神经网络参数进行剪枝、量化和熵编码。后续方法 [46], [47] 引入码率估计网络进行码率感知训练，并在推理过程中采用统计频率或人工设计的熵模型。Zhang 等人 [48]利用全局统计信息引入一种无需网络的熵模型，以确保训练与推理的一致性。为捕获神经网络参数之间的依赖关系，后续研究 [49], [50] 引入了上下文感知自回归建模，显著提升了压缩效率。基于这些见解，我们旨在设计一种适用于动态高斯泼溅特性和流式传输需求的神经网络压缩策略。结合潜变量表示压缩，我们希望构建一个完全端到端的率失真优化框架。

III. 预备知识

A. 三维高斯泼溅

3DGS [1]通过结合大量高斯、可微泼溅和基于图块的光栅化，实现实时自由视点渲染。每个高斯由一组属性参数化，其空间范围和形状由均值 $\mu \in \mathbb{R}^3$ 和协方差矩阵 $\Sigma \in \mathbb{R}^{3\times3}$定义，

$$G(x) = \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right), \tag{1}$$

其中 $x \in \mathbb{R}^3$ 是三维空间中的任意位置， Σ由缩放矩阵 S 和旋转矩阵R 构成，即 $\Sigma = RSS^\top R^\top$。随后，渲染像素颜色 C 通过 α混合合成，将这些重叠泼溅合成为：

$$C = \sum_{i\in\mathcal{N}} c^i \alpha^i \prod_{j=1}^{i-1}(1-\alpha^j), \tag{2}$$

其中 $\mathcal{N}$ 是对该像素有贡献的已排序高斯集合， α^i 是投影后第 i 个高斯的不透明度， c^i 是其由球谐函数获得的视点相关颜色。

B. 神经高斯表示

Scaffold-GS [28] 通过隐式表示消除 3DGS 的局部空间冗余，从而提升其紧凑性和保真度。它引入神经高斯 $\mathcal{NG}$作为基本表示单元，利用基于体素的聚类机制来管理 M 局部高斯。每个神经高斯由一组属性定义：

$$\mathcal{NG} = \{X \in \mathbb{R}^3, o \in \mathbb{R}^{3\times M}, F \in \mathbb{R}^D, l \in \mathbb{R}^3\}, \tag{3}$$

其中 X 表示通过体素化获得的锚点位置， o 表示由 M 管理的局部的偏移量

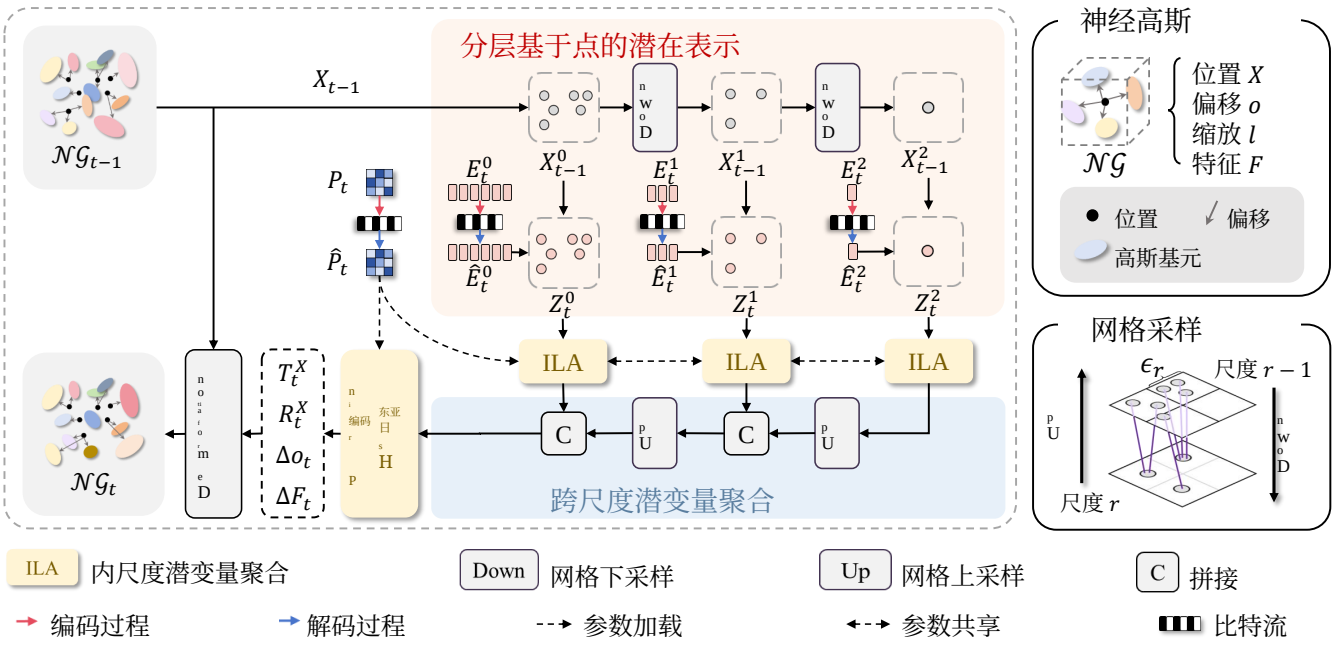

图1：所提出的HPC框架流程。该框架以最新解码的神经高斯 $\mathcal{NG}_{t-1}$作为参考。随后，它通过逐步对其位置进行下采样，构建分层潜在表示 $\{Z_t^r\}_{r=0}^{L-1}$（此处为 $L=3$），并将其与解码后的潜在嵌入 $\{\hat{}E_t^r\}_{r=0}^{L-1}$配对。经过内尺度潜在聚合（ILA）和跨尺度潜在聚合（CLA）后，这些潜在点被送入预测头，以获得用于变形的帧间残差。在HPC中，潜在嵌入 $\{E_t^r\}_{r=0}^{L-1}$和网络参数 P_t 都会经过压缩以便传输。我们用帽形标记表示从解码器重建的元素。

高斯，而 l 是用于正则化其空间分布的缩放因子。利用这些属性，M 局部高斯的位置可计算为：

$$\{\mu^i\}_{i=0}^{M-1} = X + \{o^i\}_{i=0}^{M-1} \cdot l. \tag{4}$$

局部高斯的其余属性通过若干专用多层感知机Φ_S与隐式特征 F 解耦，并以相对视角信息 δ为条件：

$$\{c^i, R^i, S^i, \alpha^i\}_{i=0}^{M-1} = \Phi_S(F, \delta). \tag{5}$$

IV. 动态高斯表示

基于神经高斯在压缩静态3DGS场景方面的成功经验 [29]–[31]，我们通过采用一系列神经高斯 $\{\mathcal{NG}_t\}_{t=0}^{T-1}$作为每帧的场景表示，将该表示扩展至动态场景，其中 t 表示时间步，T 表示总帧数。遵循前一帧的工作 [7]，，我们利用 HAC [29] 为初始帧生成紧凑的 $\mathcal{NG}_0$ 。

后续帧的重建流程如图1所示。给定时间步 t，我们将最新解码的神经高斯 $\mathcal{NG}_{t-1}$ 作为起点。为高效建模时间变化，我们通过逐步对 $\mathcal{NG}_{t-1}$ 的位置进行下采样，并与可学习潜在嵌入配对，构建分层点式潜在表示（第IV-A节）。为实现紧凑表示，潜在表示经过两阶段聚合：先进行内尺度聚合，再进行跨尺度聚合。随后，聚合后的潜在表示引导 $\mathcal{NG}_{t-1}$到 $\mathcal{NG}_t$的变形（第IV-B节），并通过运动仿射变换和特征补偿重建当前时间步的场景。

A. 分层基于点的潜在表示

为与神经高斯固有的离散且非均匀空间分布相匹配，我们提出使用分层基于点的潜在表示进行帧间预测。分层潜在点的构建如图1所示。我们首先根据主位置将最细尺度初始化为 $X_{t-1}^0 = X_{t-1}$，然后通过网格采样 [51]递归地获得更粗的尺度：

$$X_{t-1}^r = \left\lfloor X_{t-1}^{r-1}/\epsilon^r + 0.5 \right\rfloor \cdot \epsilon^r, \quad r = 1, 2, \ldots, L-1, \tag{6}$$

其中，r、ϵ和 L 分别表示尺度索引、网格尺寸和总尺度数。对于每个由此得到的位置X_{t-1}^r，我们初始化一个相应的潜在嵌入 E_t^r。经过优化、压缩和传输后，解-ˆ码器接收重建后的版本 E_t^r，随后将其与相应的 X_{t-1}^r 相结合，构建完整的分层潜在点 $\{Z_t^r = (X_{t-1}^r, \hat{}E_t^r)\}_{r=0}^{L-1}$。通过这种设计，潜在点能够捕获不同尺度下的局部空间特性，从而更好地捕获非均匀分布的神经高斯的时序动态，同时保持紧凑性，以便高效存储和传输。

为此，我们利用已建立的潜在表示，从最粗尺度开始逐步聚合潜在点

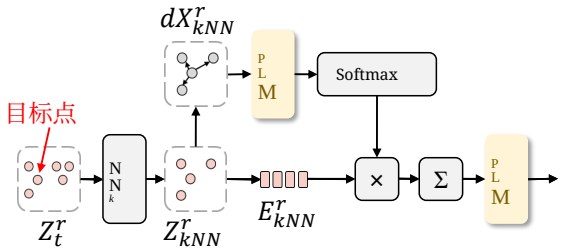

图2：内尺度潜在聚合（ILA）。ILA 以一个目标点为输入，并定位其k个最近邻。随后，根据相对位置预测聚合权重，对邻居嵌入进行加权求和，最后将结果输入多层感知机以生成输出。

直至最精细尺度。我们将这一聚合过程分解为两个互补的组成部分：内部尺度潜变量聚合（ILA）和跨尺度潜在聚合（CLA）。

内尺度潜在聚合。 给定潜在点 $Z^r_t = (X^r_{t-1},\ \hat{}E^r_t)$，ILA 模块旨在聚合某个点的邻居嵌入，以建模局部特征。如图2所示，我们首先介绍 kNN 算法，用于为每个潜在点搜索 k 个最近邻。随后，将目标潜在点与其邻居之间的相对位置偏移 dX^r_{kNN} 作为辅助信息，用于引导聚合。具体而言，将这些相对位置输入多层感知机，以预测每个邻居的聚合权重，并通过 Softmax 对其进行归一化。然后，根据这些权重自适应地融合邻居嵌入 E^r_{kNN} 。最后，将融合结果输入最终的多层感知机，以整合所有通道的信息并生成聚合结果。为减少参数数量，不同尺度的 ILA 模块共享同一组参数。这种聚合方案有助于相邻潜在点之间进行交互，从而促进冗余削减并实现空间紧凑性。

跨尺度潜在聚合。在 ILA 的基础上，我们进一步引入 CLA 过程，以融合不同分辨率的信息。如图1所示，CLA 对经过尺度内聚合的潜在点进行操作，并逐步从粗尺度处理到细尺度。在每一步中，粗尺度嵌入通过复制到对应的细尺度位置进行上采样，从而恢复分辨率。随后，将这些上采样后的点与原生细尺度潜在点拼接，以实现跨尺度融合。该过程递归重复，直至达到原始分辨率，最终生成用于后续预测的跨尺度聚合潜在点。通过 CLA 过程，聚合后的潜在表示整合了从多个感受野中捕获的局部空间上下文。这种强大的上下文使其能够有效建模非均匀分布的神经高斯的内在特征，从而为后续预测提供更优的基础。

B. 动态神经高斯形变

在聚合后的潜在点中，我们继续预测当前时间步的神经高斯 $\mathcal{NG}_{t-1}$ =，并以 $\mathcal{NG}_{t-1}$ = $\{X_{t-1}, o_{t-1}, F_{t-1}, l_{t-1}\}$前一帧状态为条件。在此变形过程中，缩放因子 l 保持不变，因为已有研究证明，改变该因子会降低性能 [7],；而 X、 o和 F 则会更新。在此设置下，变形通过两种互补机制实现：对隐式特征进行残差补偿，以及对几何形变进行运动仿射变换。

具体而言，聚合后的潜在点首先输入基于多层感知机的预测头，得到一组形变参数：特征残差 ΔF_t、锚点平移 T^X_t、锚点旋转 R^X_t 和偏移残差 Δo_t。对于隐式特征，通过添加预测得到的残差 ΔF_t 进行补偿：

$$F_t = F_{t-1} + \Delta F_t. \tag{7}$$

几何形变通过运动仿射变换建模，该变换结合了全局调整与局部调整。该变换通过平移T^X_t更新锚点位置。此外，每个局部偏移量都会获得额外的独立调整 Δo_t。为了实现表达能力更强的几何变换，我们采用了 [42] 中的策略，将全局旋转 R^X_t 应用于所有对应的偏移量。综合而言，完整的更新可简洁地表示为：

$$X_t = X_{t-1} + T^X_t, \tag{8}$$

$$o_t = R^X_t(o_{t-1} + \Delta o_t). \tag{9}$$

通过这种机制，神经高斯获得了足够的表达能力来建模时间变化，从而能够通过神经高斯泼溅实现高保真渲染 [28]。

V. 压缩方案

我们在 HPC 中集成了一套定制的压缩方案，旨在满足流式自由视点视频的传输要求。在我们的框架中，需要压缩和传输的内容包括潜在嵌入 E_t = $\{E^r_t\}^{L-1}_{r=0}$（第 V-A 节）和神经网络参数 P_t （第 V-B 节）。尤其是，在流式动态高斯泼溅领域，神经网络参数的压缩在很大程度上被忽视；现有方法 [2], [6]–[9]只是直接传输全精度的 32 位浮点数。为弥补这一不足并提升率失真效率，我们提出了一种用于压缩流式神经网络参数的新方案。最后，我们结合对这两部分的压缩，推导出整体率失真目标，以实现完全端到端的优化（第 V-C 节）。

A. 潜在嵌入压缩

得益于ILA与CLA的设计，优化后的潜在嵌入 E_t 更为紧凑，空间冗余得以降低，因而非常适合压缩。如图3（a）所示，压缩流程包含一个量化操作，将潜在嵌入离散化为整数，随后是熵编码模块。

为避免训练过程中量化的不可微性，我们采用先前工作 [46]–[48] 中的常见做法，使用不同的量化代理

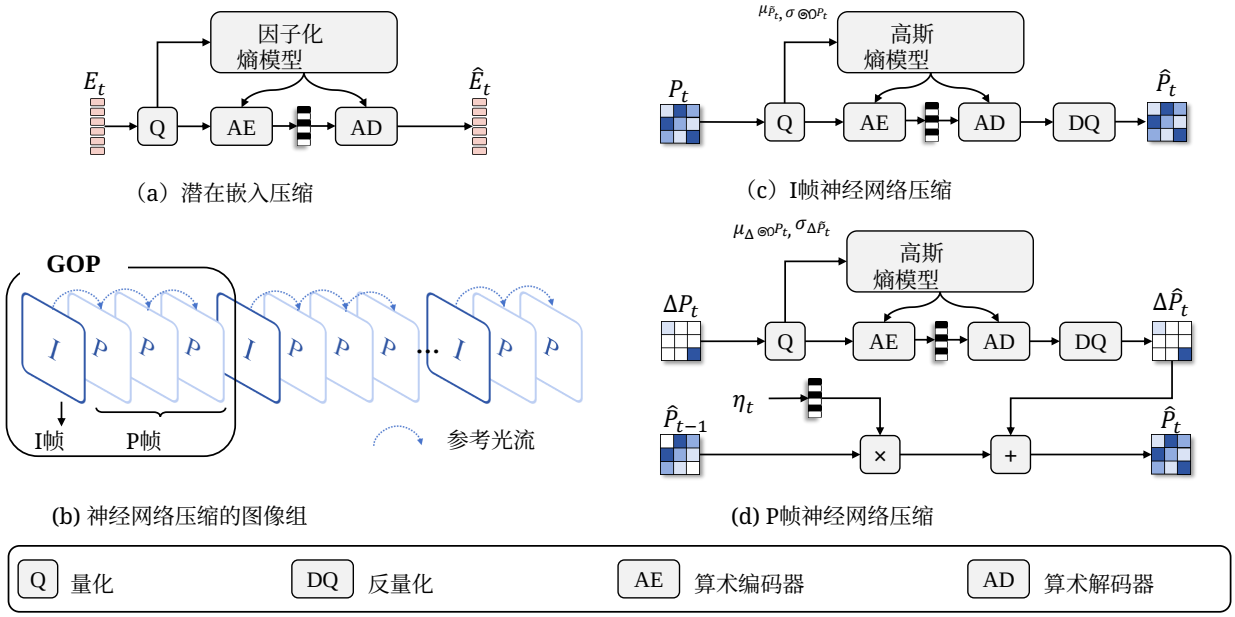

图3：HPC 的压缩方案 。我们对潜在嵌入和神经网络都进行压缩 s.

来进行失真计算和熵估计，从而获得量化嵌入 E_t。具体而言，失真计算采用直通估计器（STE）：

$$\hat{E}_t = \mathrm{SG}(\lfloor E_t \rceil - E_t) + E_t, \tag{10}$$

其中，SG(·) 表示梯度停止操作。进行熵估计时，使用添加均匀噪声 u替代舍入量化：

$$\hat{E}_t = E_t + u, \quad u \sim \mathcal{U}(-\frac{1}{2}, \frac{1}{2}). \tag{11}$$

量化后，采用广泛使用的因子化模型 [44]进行熵估计。熵模型包含若干可学习层，逐步细化输入嵌入，以计算其概率和比特率：

$$p_{\mathrm{PMF}}(\hat{E}_t) = p_{\mathrm{CDF}}(\hat{E}_t + \frac{1}{2}) - p_{\mathrm{CDF}}(\hat{E}_t - \frac{1}{2}), \tag{12}$$

$$\mathcal{R}(\hat{E}_t) = \mathbb{E}_{\hat{E}_t}[-\log_2 p_{\mathrm{PMF}}(\hat{E}_t)], \tag{13}$$

其中， $p_{\mathrm{PMF}}(\cdot)$ 是概率质量函数， $p_{\mathrm{CDF}}(\cdot)$ 是由因子化模型近似得到的累积分布函数，而 $\mathcal{R}(E_t)$ 是 E_t 的比特率估计值。利用近似得到的概率，我们可以对潜在嵌入进行熵编码，以实现进一步压缩。

B. 神经网络压缩

为了探索 HPC 中神经网络的最优压缩方式，我们首先分析其参数的特征。从图4所示的参数分布可以观察到，相应层中的参数在相邻帧之间呈现相似的分布，而不同层之间则存在显著差异。这促使我们利用时间上下文来削减冗余。

然而，实施这种朴素策略可能会引入误差传播，其中优化不佳的参数在某一帧中会对后续帧产生不利影响，从而导致性能级联下降。受传统视频编码的启发，我们借鉴图像组（GOP）[52] 的概念，以获得最优参考结构。具体而言，如图3（b）所示，一个GOP通常由一个帧内编码帧（I帧）和若干预测编码帧（P帧）组成。位于GOP起始处的I帧被独立优化并压缩，而后续P帧则通过参考前一帧进行帧间预测。通过引入周期性插入独立I帧的GOP结构，我们防止了误差传播，并随时间保持优化稳定性。

***I*帧神经网络压缩。**如图3（c）所示，I帧的压缩流程包括先量化、再进行熵编码。由于网络参数对量化误差特别敏感，我们在量化之前引入缩放操作，以更好地保留其精度：

$$\tilde{P}_t = \lfloor \frac{P_t - \min(P_t)}{\max(P_t) - \min(P_t)} \cdot (2^B - 1) \rceil, \tag{14}$$

其中， P_t 表示经过量化并传输的参数， B 表示量化的位深，用于控制量化精度。将 P_t 的最小值和最大值作为辅助信息发送给解码器，以获得ˆ 重建参数 P_t:

$$\hat{P}_t = \tilde{P}_t \cdot [\max(P_t) - \min(P_t)]/(2^B - 1) + \min(P_t). \tag{15}$$

在训练过程中，我们采用与潜在嵌入相同的代理策略：在失真项中使用 STE 进行梯度传播，并添加均匀噪声以实现可微熵估计。

量化后，采用熵编码对 $\tilde{P}_t$ 用于最终压缩。接下来， [48], 我们引入无需网络的高斯分布进行熵建模，该分布

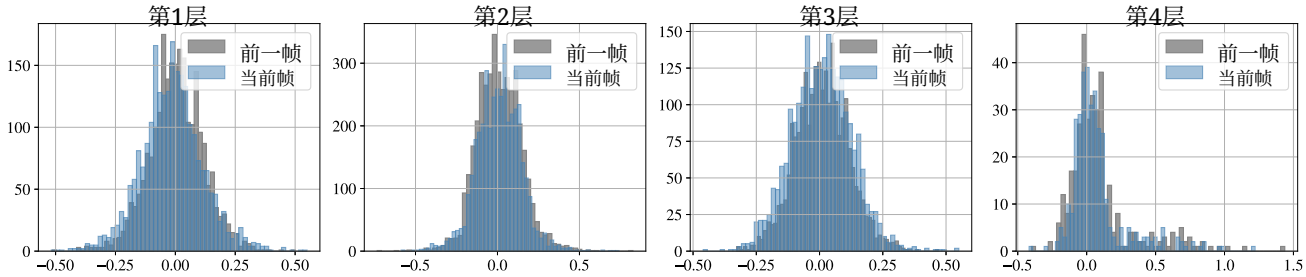

图4：不同层中相邻帧之间的参数分布。

仅由统计均值 $\mu_{\tilde{P}_t}$ 和方差 $\sigma^2_{\tilde{P}_t}$ 参数化。因此，近似概率 $p_{\text{PMF}}(\tilde{P}_t)$ 和比特率 $\mathcal{R}(P_t)$ 可表示为：

$$p_{\text{PMF}}(\tilde{P}_t) = \prod_i (\mathcal{N}(\mu_{\tilde{P}_t}, \sigma^2_{\tilde{P}_t}) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\tilde{P}_t^i), \tag{16}$$

$$\mathcal{R}(\tilde{P}_t) = \mathbb{E}_{\tilde{P}_t}[-\log_2 p_{\text{PMF}}(\tilde{P}_t)], \tag{17}$$

其中 $*$ 表示卷积，P_t^i 表示 P_t中的每个元素。该设计能够实现可微熵估计，而无需引入额外的网络参数。

***P*帧神经网络压缩。** 对于P帧，我们旨在利用前一帧的解码参数$\hat{P}_{t-1}$ 作为上下文先验，以指导当前参数的优化与压缩。为此，我们借鉴经典差分编码来优化并编码连续帧之间的残差，该残差具有更强的能量聚集特性。

P帧神经网络压缩如图3（d）所示。具体而言，受图4所示的逐层时间相关性启发，我们首先引入一个可学习的逐层缩放因子 η_t ，以调整 P_{t-1} ，从而对当前参数进行粗略预测。为弥合剩余差距，我们进一步优化逐参数残差 ΔP_t ，以实现细粒度补偿。η_t 和 ΔP_t 均被传输至解码器。由于其数量极少，逐层 η_t 以原始精度保留，不进行压缩。相比之下，逐参数 ΔP_t 会进一步压缩以实现码率降低。基于接收到的 η_t 与解码残差$\hat{\Delta P_t}$，最终参数 P_t 可按如下方式重建：

$$\hat{P}_t = \eta_t \cdot \hat{P}_{t-1} + \hat{\Delta P_t}. \tag{18}$$

为压缩残差 ΔP_t，我们采用与 I帧参数类似的量化策略。在此，我们通过引入由 P_{t-1}推导出的完整参数动态范围，对能量紧凑残差 ΔP_t 进行归一化处理，确保重构参数 P_t 以一致的目标精度表示，同时实现码率节省：

$$v_{\min} = \min(\hat{P}_{t-1}, \Delta P_t), \quad v_{\max} = \max(\hat{P}_{t-1}, \Delta P_t). \tag{19}$$

$$\Delta\tilde{P}_t = \lfloor \frac{\Delta P_t - v_{\min}}{v_{\max} - v_{\min}} \cdot (2^B - 1) \rceil, \tag{20}$$

$$\hat{\Delta P_t} = \Delta\tilde{P}_t \cdot (v_{\max} - v_{\min})/(2^B - 1) + v_{\min}. \tag{21}$$

对于进一步的熵计算，我们采用与 I帧参数压缩相同的策略：

$$p_{\text{PMF}}(\Delta\tilde{P}_t) = \prod_i (\mathcal{N}(\mu_{\Delta\tilde{P}_t}, \sigma^2_{\Delta\tilde{P}_t}) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\Delta\tilde{P}_t^{\,i}), \tag{22}$$

$$\mathcal{R}(\Delta\tilde{P}_t) = \mathbb{E}_{\Delta\tilde{P}_t}[-\log_2 p_{\text{PMF}}(\Delta\tilde{P}_t)]. \tag{23}$$

在所提出的时间参考策略中，优化后的残差具有更高的能量紧凑性。与编码完整参数相比，这降低了传输成本，同时仍保留了神经网络的表征能力。

C. 率失真优化

最后，我们通过最小化总估计比特率与渲染失真的组合损失，对整个框架进行联合优化，以端到端的方式追求最佳率失真性能。给定时间步 t，我们计算失真损失 $\mathcal{L}_{\mathcal{D}}^{(t)}$ 与码率损失 $\mathcal{L}_{\mathcal{R}}^{(t)}$ 的加权和，作为整体监督目标 $\mathcal{L}^{(t)}$：

$$\mathcal{L}^{(t)} = \mathcal{L}_{\mathcal{R}}^{(t)} + \lambda\mathcal{L}_{\mathcal{D}}^{(t)}, \tag{24}$$

其中，λ 是用于控制率失真平衡的拉格朗日乘子。具体而言，我们采用 3DGS渲染损失 [1] 作为失真项：

$$\mathcal{L}_{\mathcal{D}}^{(t)} = \mathcal{L}_1^{(t)} + \lambda_{SSIM}\mathcal{L}_{SSIM}^{(t)}. \tag{25}$$

对于码率项，我们将潜在嵌入和神经网络参数的估计比特率结合起来：

$$\mathcal{L}_{\mathcal{R}}^{(t)} = \begin{cases} \mathcal{R}(\hat{E}_t) + \mathcal{R}(\tilde{P}_t), & t \bmod T_{GOP} = 0, \\ \mathcal{R}(\hat{E}_t) + \mathcal{R}(\Delta\tilde{P}_t), & t \bmod T_{GOP} \neq 0, \end{cases} \tag{26}$$

其中，T_{GOP} 是预定义的图像组大小。

VI. Experiments

A. 实验设置

数据集。 我们在两个广泛采用的自由视点视频数据集上评估了我们的方法：**(1) N3DV 数据集** [54],由一个包含 21 台相机的相机阵列采集，其中包含 2704×2028、帧率为 30 FPS 的动态场景。**(2) Meet Room 数据集** [55],由一个包含 13 台相机的相机系统采集，其中包含 1280×720 、帧率为 30 FPS 的动态场景。对于每个数据集，我们留出一个相机视角用于测试，其余视角用于训练。

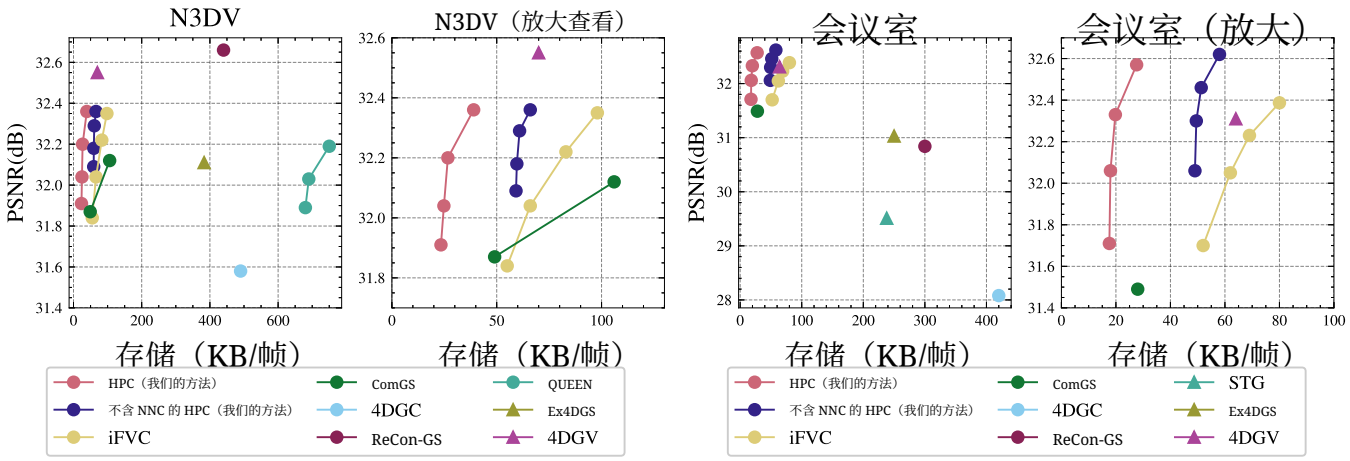

图 5：N3DV 和 Meet Room 数据集上的率失真曲线。标记为 ◯ 和 △ 的方法分别表示在线优化方法和离线优化方法。越接近左上角的点和曲线，性能越好。



图 6：*flame steak* 和 *vrheadset* 上的定性比较。为确保公平比较，我们使用相同的初始稀疏点重新训练所有方法，而 iFVC [7] 和 HPC 使用相同的初始帧模型。

实现细节。对于潜在点层次结构，我们采用 ContextGS [30] 的方法构建 $L = 3$个尺度，相邻尺度间的点数比例设为 0.2。每个尺度的潜在嵌入包含 16 个通道。在 ILA 模块中，我们将 $k = 4$ 设为 k-近邻搜索的参数。对于神经网络压缩，参数量化为 B =8 位深。图像组大小 T_{GOP} 设为 5。训练时，将 λ从 0.048 调整至 0.002，以实现可变比特率。我们按照 iFVC [7]的方法训练 HAC [29] 作为初始帧，迭代 15000 次。对于后续帧，我们训练模型 500 次迭代。我们使用 torchac 库 [56]进行算术编码。所有实验均在搭载 NVIDIA GeForce RTX 3090 图形处理器的设备上进行。

基线。我们将提出的 HPC 与当前最先进的在线框架进行比较，以评估其性能，包括 3DGStream [2], HiCoM [3], iFVC [7], 4DGC [6], QUEEN [9],ComGS [5], 4DGCPro [8], 和 ReCon-GS [4]。我们还将几种在多帧范围内进行优化的离线方法 [34], [37], [42], [43], [53], 纳入基准测试，作为参考，以便更全面地评估性能。由于现有方法缺少神经网络压缩，我们通过在完整模型中禁用该组件，引入相应的基线 HPCw/o NNC。这样既能与以往方法进行公平比较，也能单独评估其他组件的贡献。

指标。我们使用峰值信噪比（峰值信噪比）和结构相似性指数（结构相似性指数） [57] 作为客观失真指标来评估率失真性能，比特率以每帧千字节（KB/帧）衡量。为精确比较率失真效率，我们计算

表I：定量比较。在线方法按顺序分为三类：基于网格的方法、现有的基于点的方法和我们的方法，并用水平线分隔各类别。对于具有多个率失真点的方法，我们报告质量最低（*-l*）和最高（*-h*）的结果。最佳和次佳结果分别以红色和橙色单元格突出显示。

N3DV 数据集				
类别	方法	PSNR（dB）↑	结构相似性指数↑	存储（KB/帧）↓
离线	STG [34]	32.05	-	666
	GIFStream [42]	31.75	0.938	33
	Ex4DGS [53]	32.11	0.94	383
	MEGA [37]	31.49	-	83
	4DGV [43]	32.55	-	70
在线	3DGStream [2]	31.69	0.948	7780
	4DGC [6]	31.58	0.943	500
	4DGCPro [8]-*l*	30.68	0.926	210
	4DGCPro [8]-*h*	31.64	0.944	640
	iFVC [7]-*l*	31.84	0.950	63
	iFVC [7]-小时	32.35	0.953	99
	HiCoM [3]	31.17	-	900
	QUEEN [9]-*l*	31.89	0.946	680
	QUEEN [9]-*h*	32.19	0.946	750
	ComGS [5]-*l*	31.87	-	49
	ComGS [5]-高	32.12	-	106
	ReCon-GS [4]	32.66	0.957	440
	HPC（无NNC）-低（我们的方法）	32.22	0.953	59
	HPC（无NNC）-高（我们的方法）	32.36	0.955	66
	HPC-低（我们的方法）	31.91	0.951	23
	HPC-*h*（我们的方法）	32.36	0.955	39

Meet Room 数据集				
类别	方法	PSNR（dB）↑	结构相似性指数↑	存储（KB/帧）↓
离线	STG [34]	29.51	0.932	238
	Ex4DGS [53]	31.03	0.946	250
	4DGV [43]	32.31	0.957	64
在线	3DGStream [2]	30.79	-	4100
	4DGC [6]	28.08	0.922	420
	iFVC [7]-*l*	32.05	0.956	63
	iFVC [7]-小时	32.39	0.959	80
	HiCoM [3]	26.73	-	600
	ComGS [5]	31.49	-	28
	ReCon-GS [4]	30.84	0.954	300
	HPC（无NNC）-*l*（我们的）	32.26	0.959	52
	HPC（无NNC）-*h*（我们的）	32.62	0.962	61
	HPC-*l*（我们的）	31.71	0.957	18
	HPC-*h*（我们的）	32.57	0.961	28

Bjontegaard Delta比特率（BD-BR）[58]，，用于量化在等效峰值信噪比质量下的比特率节省。请注意，负的BD-BR值表示与锚定方法相比，在相同保真度下存储有所降低，这是理想的。

B. 比较结果

定量比较。定性比较详见表I。HPC 在两个数据集上均实现了有竞争力的重建质量，同时存储开销最低约为 20 KB/帧，显著低于其他在线优化方法。通过在训练期间调整 λ，HPC 能够在 Meet Room 数据集上实现最佳保真度，在 N3DV 数据集上实现次佳保真度，同时存储开销低于 40 KB/帧。值得注意的是，重建保真度会受到初始帧的显著影响，相关讨论见第 VI-D 节。综合而言，图 5 展示了总体率失真性能，可以看出，我们的 HPC 显著优于以往最先进的方法。为进行公平比较，我们使用与我们采用相同初始帧的 iFVC [7]，作为锚点来计算 BD-BR 指标。如表 II 所示，HPC 在两个数据集上的码率节省达到 67.3%。即使禁用神经网络压缩，其码率仍降低了 28%，这凸显了其表示设计的重要贡献。

表II：与 iFVC [7] 相比的 BD-BR（%）结果。测试采用 N3DV 和 Meet Room 数据集。

方法	N3DV	Meet Room	平均值
iFVC [7]	0.0	0.0	0.0
HPC	**-64.9**	**-70.7**	**-67.3**
HPC（无NNC）	-27.1	-28.9	-28.0
HPC（无ILA）	-60.0	-66.7	-63.4
HPC（无CLA）	-45.9	-51.0	-48.5

我们对显著的码率降低进行了全面分析。与 HiCoM [3], ComGS [5],和 ReCon-GS [4],等直接优化高斯属性残差的方法不同，HPC采用了不同策略：它优化潜在表示，并配合轻量级网络来预测残差。这种隐式建模方式使熵约束能够得到恰当应用，从而实现更高的紧凑性和压缩效率。与 iFVC [7], 4DGC [6], 4DGCPro [8], 和 QUEEN [9]，等采用隐式动态模型的方法相比，HPC构建了更紧凑、更高效的潜在表示。结合神经网络压缩后，整体性能提升更为显著。

定性比较。我们在 *flame steak* 序列和 *vrheadset* 序列上与 4DGC [6] 和 iFVC [7] 进行了定性比较。为确保比较公平，所有方法均使用相同的稀疏点云进行初始化。此外，iFVC [7] 和我们的 HPC 使用完全相同的首帧模型。如图 6 所示，HPC 在实现最小模型大小的同时，获得了最佳重建质量。值得注意的是，HPC 能更好地保留面部特征、动物毛发和火焰纹理等精细细节。此外，在对快速运动的手臂和手部等大幅运动部件进行建模时，它也表现出更强的能力。这些结果表明，HPC 能够准确捕捉动态场景元素，保留复杂物体的高保真细节，并实现高度紧凑的表示。

C. 消融研究

为评估各个组件的影响，我们进行了全面的消融研究。具体而言，我们围绕三项主要贡献展开分析：潜在表示、潜在表示聚合方案和神经网络压缩方案。

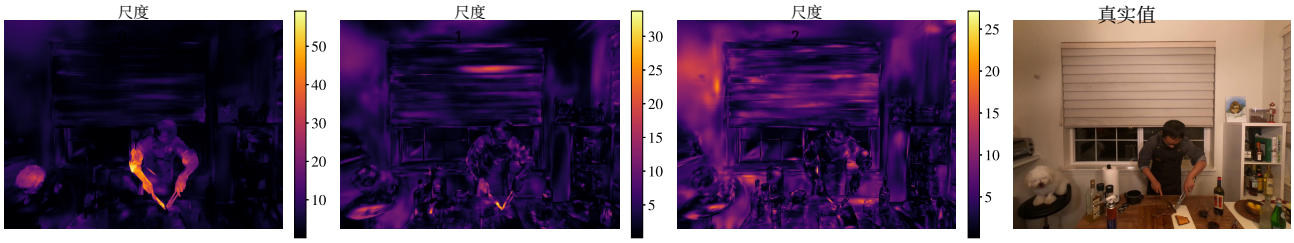

图 7：不同尺度下潜在嵌入的比特分配。此可视化将每个潜在嵌入消耗的比特率映射为颜色值，再渲染到二维图像上，以展示比特分配。

表 III：N3DV 数据集上不同潜在表示的消融研究。三平面是 BD-BR 的锚点。

	三平面	哈希网格	分层点（我们的方法）
BD-BR (%) ↓	0.0	-6.7	**-21.8**

表 IV：神经网络量化采用不同位深的研究。结果基于 Meet Room 数据集的最高比特率。

位深	PSNR (分贝) ↑	潜变量 存储（千字节）↓	网络 存储（千字节）↓	总计 存储（千字节）↓
32	**32.62**	16	42	58
16	32.60	16	21	37
8	32.57	16	10	**26**
4	7.31	60	5	65

不同类型的表示。 我们首先将提出的分层潜在点与两种成熟的结构表示进行比较：哈希网格和三平面。哈希网格采用与 3DGStream [2], 相同的配置，而三平面采用与 iFVC [7] 相同的配置。如表 III 所示，提出的分层点表示在三种候选方案中表现最佳。这一结果表明，与结构表示相比，我们的设计能够为离散高斯实现更高效的参数分配。

潜变量聚合。我们按照表II中的描述，分别消融 ILA 和 CLA 模块，以评估所提出的聚合方案。加入 ILA 模块后，我们的模型可额外节省 4% 的比特率。这一结果证实，该组件能够有效促进局部信息共享，从而减少空间冗余。在评估 CLA 模块时，其额外节省了 20% 的比特率，凸显了聚合多分辨率感受野信息的有效性。为了进一步

表V：Meet Room数据集上的神经网络压缩消融研究。M_0 是用于BD-BR计算的锚点方法。

方法	M_0	M_1	M_2
量化	✓	✓	✓
熵约束		✓	✓
时间参考			✓
BD-BR（%）↓	0.0	-6.6	**-14.6**

证明这一点，我们展示了不同尺度下的比特分配结果。如图 7 所示，不同尺度间的比特分配存在显著差异。这表明，多尺度表示使 HPC 能够捕捉不同粒度层次的特征。粗尺度下的潜在嵌入侧重于捕捉大范围区域中的运动（例如，移动的手臂），而较细尺度则擅长建模细节变化。跨尺度的信息聚合使 HPC 能够更高效、更全面地对场景进行建模。

神经网络压缩。压缩前，必须先对参数进行量化。为探究其影响，我们将精度从原始的32位降低到若干较低的位深，包括16位、8位和4位。如表 IV 所示，网络参数的存储成本随量化位深呈线性下降。与原始精度相比，8位或16位下的重建结果仅有轻微劣化（小于1 dB）。然而，采用4位量化时，训练会变得不稳定并出现发散，导致重建质量显著下降。同时，对因子化熵模型进行过度量化会导致潜在表示的比特率增加。基于上述分析，我们选择8位量化作为工作点，在有效降低存储开销的同时保留神经网络的核心能力。

为进一步开展消融研究，我们逐步将其他组件纳入采用8位量化的基线，记为M_0。如表 V 所示，为了高效压缩参数，我们在参数优化中加入熵约束，记为M_1。该方法通过可微无网络高斯熵模型 [48] 实现，以支持比特率估计和熵编码，从而带来6.6%的性能提升。

最后，我们利用跨帧时间参考来挖掘帧间相关性、削减冗余，记作 M_2。如图8所示，我们的时间参考策略使模型能够捕捉并传输紧凑的残差。这一策略在不损害网络表征能力的前提下，显著节省了码率。然而，如图9所示，直接整合时间参考模型会导致严重的误差传播，表现为随着参考间隔增大，重建质量持续下降。因此，我们引入 GOP 参考结构，并研究其大小的影响。如图9所示，与不采用帧间参考（GOP 大小为）相比，

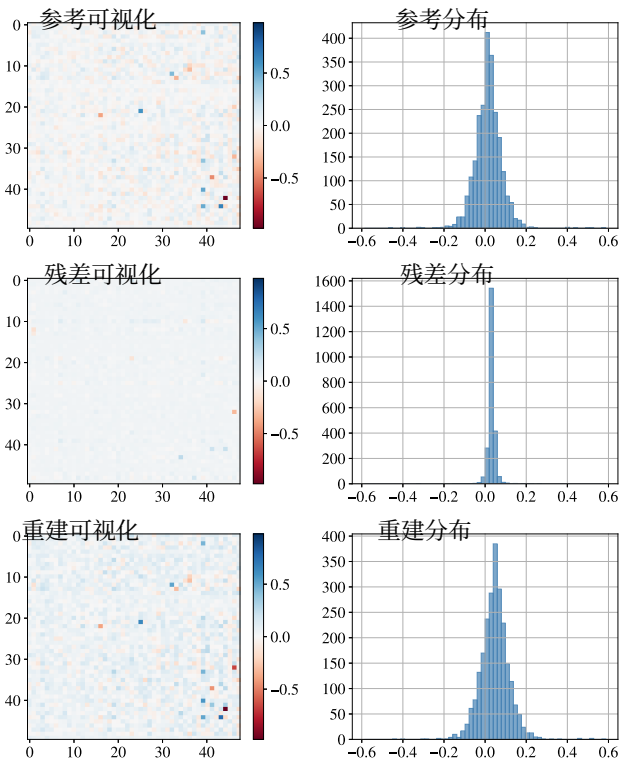

图8：神经网络参数和残差分析。重建参数根据公式（18）利用参考参数和残差计算得出。

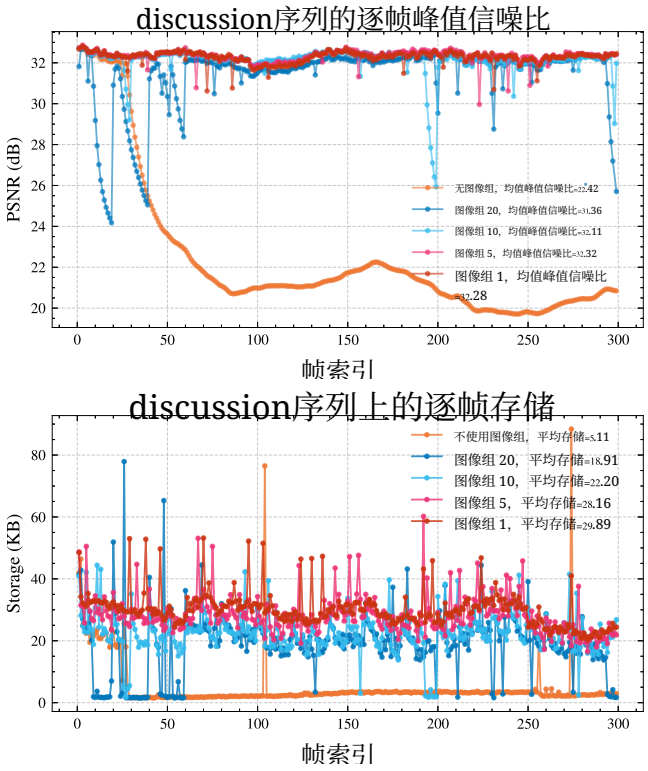

图9：不同 GOP 大小下，discussion 序列的逐帧峰值信噪比和存储。

表V：复杂度对比。结果在 NVIDIA RTX 3090 GPU 上评估，数值为 N3DV 和 Meet Room 数据集上的均值。

方法	训练 (分钟)↓	渲染 (帧率)↑	编码 (s)↓	解码 (s)↓
4DGC [6]	0.83	184	0.70	0.11
iFVC [7]	0.17	140	0.10	0.09
不含 NNC 的 HPC（我们的方法）	0.60	141	0.13	0.10
HPC（我们的方法）	1.01	140	0.33	0.18

较大的 GOP 大小始终能够降低总体存储量。在重建质量方面，GOP 大小为 10 和 20 时仍会出现周期性退化，从而损害整体性能。根据经验，我们选择 GOP 大小为 5，因为它不仅能够降低比特率，还能略微提升重建质量。这一质量提升源于恰当选择的时间参考所带来的有益优化偏置，它引导网络朝着更优的优化方向发展，从而增强其表达能力。采用 GOP 参考结构后，如表 V 所示，我们的时间参考策略使性能较基础模型提升了 14.6%。

D. 分析

复杂度。表VI比较了我们的方法与 4DGC [6] 和 iFVC [7] 在训练时间、渲染速度、编码时间和解码时间方面的复杂度。对于训练时间而言，HPC 较长的耗时源于其针对网络参数的率感知训练方案，该方案引入了更复杂的梯度更新，从而减慢了优化速度。在移除神经网络压缩模块后，HPC 的训练速度超过了 4DGC [6]，但仍慢于 iFVC [7]。这主要是由于引入的聚合模块和熵模型增加了额外参数，进而需要更多迭代次数才能收敛。由于神经高斯泼溅 [28],带来的开销，HPC 和 iFVC [7] 的渲染速度略慢；相比之下，4DGC [6]使用了速度更快的非神经原始泼溅 [1]。HPC 的整体编码时间更长，是因为它需要对潜在嵌入和网络参数都进行熵编码，而另外两种方法仅压缩前者。在仅关注编码潜在嵌入这一共同任务时，HPC 的编码速度与 iFVC [7] 相当，并优于 4DGC [6]。

初始帧分析。在流式自由视点视频中，初始帧对后续所有重建都至关重要。鉴于其关键作用，我们通过分析不同方法对初始帧的处理方式及结果，全面评估了我们的工作，如表VII所示。

不同方法采用了多种初始表示和训练策略。3DGStream [2] 对初始帧采用原始三维高斯泼溅表示及训练策略。ReCon-GS [4]通过向位置属性注入高斯噪声来正则化3DGS的训练，以缓解过拟合。与原始三维高斯泼溅不同，iFVC和我们的HPC均采用HAC框架 [29]，以获得高度压缩的神经高斯 [28] 表示，用于初始

表七：N3DV 上不同方法的初始帧分析。所有帧的平均结果基于最高比特率。

方法	表示	初始训练策略	初始存储（千字节）↓	平均存储（千字节）↓	初始峰值信噪比（分贝）↑	平均峰值信噪比（dB）↑	峰值信噪比增益相较于初始值（%）↑
3DGStream [2]	原始三维高斯泼溅	原始三维高斯泼溅训练	51660	7780	32.13	31.69	-1.36
ReCon-GS [4]	原始三维高斯泼溅	注入噪声的三维高斯泼溅训练	11120	440	**32.75**	**32.66**	-0.27
iFVC [7]	神经高斯	HAC 训练	**2480**	99	32.21	32.35	+0.43
HPC（我们的方法）	神经高斯	HAC 训练	**2480**	**39**	32.21	32.36	**+0.46**

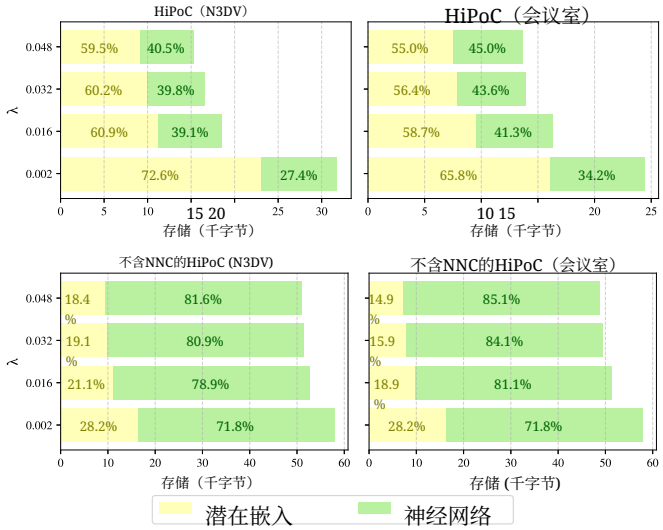

图 10：潜在嵌入与神经网络之间的比特分配。

帧。该方法实现了显著的存储节省（相对于3DGStream [2]和ReCon-GS [4], 分别减少20×和4.5×）。在重建保真度方面，噪声注入策略有效防止ReCon-GS [4]对训练视图过拟合，从而在初始帧和后续帧上均取得最高峰值信噪比。然而，我们观察到，使用原始三维高斯泼溅 [1] 作为表示会导致质量下降，而采用神经高斯 [28] 则能相对于初始重建质量获得稳定增益。在基于神经高斯的方法中，我们的HPC取得了最高的0.46%峰值信噪比增益，进一步证明了其有效性。

比特分配。为了全面分析我们的方法，我们分析了不同比特率下潜在嵌入与神经网络之间的比特分配结果。如图 10 所示，不同比特率点的码率分配各不相同。在较低比特率下，两者分配到的码率相近。随着总比特率预算增加，分配给潜在嵌入的比例持续上升，在最高比特率下达到约 70%。

当神经网络压缩模块被禁用时，我们观察到，在低比特率和中等比特率点上，网络参数的比特率占比激增至 80%以上。这种过高的分配不仅阻碍了进一步的码率降低，还可能制约模型的整体表达能力，从而表明压缩网络参数。

七、结论与未来工作

我们探索了一种专为动态高斯泼溅设计的分层点式潜在表示，并提出了 HPC。该方法能够高效地分配参数，有效捕获非均匀分布的高斯之间的空间相关性，最终实现高效且高质量的重建。我们设计了一种完全端到端的压缩框架，联合优化潜在表示和神经网络，以实现最佳率失真权衡并显著节省码率。大量实验表明，与现有方法相比，HPC 实现了最先进的压缩性能。全面的研究分析并验证了其各项技术组件的有效性。通过显著节省动态高斯泼溅的码率，我们的工作为自由视点视频流传输领域铺平了道路。

HPC的主要局限在于，与竞争方法相比，其训练时长更长、编码延迟更高，这是由于额外的率感知优化以及对网络参数的压缩所致。未来工作应纳入轻量化设计，以加速训练和编码过程，从而促进实际部署。

参考文献

[1] B. Kerbl, G. Kopanas, T. Leimk¨uhler 和 G. Drettakis，“用于实时辐射场渲染的三维高斯泼溅”。*ACM*图形学汇刊，第42卷，第4期，第139–1页，2023年。[2] J. Sun, H. Jiao, G. Li, Z. Zhang, L. Zhao 和 W. Xing，“3DGStream：面向照片级真实感自由视点视频高效流传输的3D高斯即时训练”，载于*IEEE/CVF*计算机视觉与模式识别会议（*CVPR*）论文集，2024年6月，第20 675–20 685页。[3] Q. Gao, J. Meng, C. Wen, J. Chen 和 J. Zhang，“HiCoM：利用3D高斯泼溅实现动态可流式传输场景的分层相干运动”，载于神经信息处理系统进展（*NeurIPS*），2024年。[4] J. Fu, Q. Gao, C. Wen, Y. Wu, S. Ma, J. Zhang 和 J. Zhang，“Recon-GS：保持连续性的高斯流传输，实现动态场景的快速紧凑重建”，载于第三十九届神经信息处理系统年会，2025年。[5] J. Chen, Q. Mao, Y. Bao, X. MENG, F. Meng, R. Wang 和 Y. Liang，“运动至关重要：用于自由视点视频重建的紧凑型高斯流传输”，载于第三十九届神经信息处理系统年会，2025年。[6] Q. Hu, Z. Zheng, H. Zhong, S. Fu, L. Song, X. Zhang, G. Zhai 和 Y. Wang，“4DGC：面向高效可流式传输自由视点视频的率感知4D高斯压缩”，载于计算机视觉与模式识别会议（*CVPR*）论文集，2025年6月，第875–885页。[7] L. Tang, J. Yang, R. Peng, Y. Zhai, S. Shen 和 R. Wang，“将可流式传输自由视点视频压缩至每帧0.1 MB”，载于*AAAI*人工智能会议论文集，第39卷，第7期，2025年，第7257–7265页。

[8] Z. Zheng, Z. Wu, H. Zhong, Y. Tian, N. Cao, L. Xu, J. Yao, X. Zhang, Q. Hu 和 W. Zhang，“4DGCPro：用于渐进式体积视频流传输的高效分层4D高斯压缩”，载于第三十九届神经信息处理系统年会，2025年。[9] S. Girish, T. Li, A. Mazumdar, A. Shrivastava, D. Luebke 和 S. D. Mello，“QUEEN：用于自由视点视频流传输的量化高效编码”，载于第三十八届神经信息处理系统年会，2024年。[10] T. M¨uller, A. Evans, C. Schied 和 A. Keller，“采用多分辨率哈希编码的即时神经图形基元”，ACM图形学汇刊（TOG），第41卷，第4期，第1–15页，2022年。[11] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht 和 A. Kanazawa，“K- 平面：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，2023年，第12 479–12 488页。[12] H. Yu, W. Gong, J. Chen 和 H. Ma，“Get3dgs：基于点变形场生成3D高斯”，IEEE视频技术电路与系统汇刊，第35卷，第5期，第4437–4449页，2025年。[13] Y. Ma, B. Liu, J. Li, L. Li 和 D. Liu，“哈希网格特征剪枝”，arXiv预印本 arXiv:2512.22882，2025年。[14] S. Xie, W. Zhang, C. Tang, Y. Bai, R. Lu, S. Ge 和 Z. Wang，“Mesongs：通过高效属性变换对3D高斯进行训练后压缩”，载于欧洲计算机视觉会议。施普林格，2024年，第434–452页。[15] S. Lee, F. Shu, Y. Sanchez, T. Schierl 和 C. Hellge，“利用优化特征平面和标准视频编解码器压缩三维高斯泼溅”，载于IEEE/CVF 国际计算机视觉会议（ICCV）论文集，2025年10月，第25 496–25 505页。[16] S. Girish, K. Gupta 和 A. Shrivastava，“Eagles：采用轻量级编码的高效加速3D高斯”，载于欧洲计算机视觉会议，2024年，第54–71页。[17] S. Li, C. Wu, H. Li, X. Gao, Y. Liao 和 L. Yu，“Gscodec studio：高斯泼溅压缩的模块化框架”，IEEE视频技术电路与系统汇刊，第1–1页，2026年。[18] Q. Yang, L. Yang, G. V. der Auwera 和 Z. Li，“HybridGS：利用双通道稀疏表示和点云编码器实现高效的高斯泼溅数据压缩”，载于第四十二届国际机器学习会议，2025年。[19] Y. Wang, M. Liu, Q. Yang, H. Huang, L. Yang 和 Y. Xu，“动态网络上的三维高斯泼溅高效自适应流传输”，IEEE视频技术电路与系统汇刊，第1–1页，2025年。[20] C. Wang, S. N. Sridhara, E. Pavez, A. Ortega 和 C. Chang，“面向三维高斯泼溅数据变换编码的自适应体素化”，载于2025年IEEE国际图像处理会议（ICIP），2025年，第2414–2419页。[21] M. S. Ali, M. Qamar, S.-H. Bae 和 E. Tartaglione，“去除冗余：通过剪枝高效压缩3D高斯泼溅”，载于第35届英国机器视觉会议2024（BMVC 2024），英国格拉斯哥，2024年11月25日至28日。BMVA，2024年。[22] Z. Fan, K. Wang, K. Wen, Z. Zhu, D. Xu 和 Z. Wang，“Lightgaussian：压缩率提升15倍的无界3D高斯压缩，以及 200+帧率”，载于第38届神经信息处理系统国际会议论文集，系列：NIPS ’24。美国纽约州雷德胡克：Curran Associates Inc.，2024年。[23] H. Wang, H. Zhu, T. He, R. Feng, J. Deng, J. Bian 和 Z. Chen，“端到端率失真优化的3D高斯表示”，载于欧洲计算机视觉会议。施普林格，2024年，第76–92页。[24] 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2024年IEEE/CVF计算机视觉与模式识别会议（CVPR），2024年，第21 719–21 728页。[25] X. Liu, X. Wu, P. Zhang, S. Wang, Z. Li 和 S. Kwong，“Compgs：通过压缩高斯泼溅实现高效3D场景表示”，载于第32届ACM多媒体国际会议论文集，2024年。[26] K. Navaneet, K. P. Meibodi, S. A. Koohpayegani 和 H. Pirsiavash，“Compgs：通过矢量量化实现更小、更快的高斯泼溅”，ECCV，2024年。[27] S. Niedermayr, J. Stumpfegger 和 R. Westermann，“用于加速新视图合成的压缩三维高斯泼溅”，载于2024年IEEE/CVF计算机视觉与模式识别会议（CVPR），2024年，第10 349–10 358页。[28] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin 和 B. Dai，“Scaffold-GS：用于视图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，2024年，第20 654–20 664页。[29] Y. Chen、Q. Wu、W. Lin、M. Harandi和J. Cai，“HAC：面向3D高斯泼溅压缩的哈希网格辅助上下文”，载于欧洲计算机视觉会议。施普林格，2024年，第422–438页。[30] Y. Wang、Z. Li、L. Guo、W. Yang、A. Kot和B. Wen，“ContextGS：基于锚点级上下文模型的紧凑3D高斯泼溅”，《神经信息处理系统进展》，第37卷，51 551，2024年，第51 532页–。[31] Y.-T. Zhan、C.-Y. Ho、H. Yang、Y.-H. Chen、J. C. Chiang、Y.-L. Liu和W.-H. Peng，“Cat-3dgs：面向率失真优化3DGS压缩的上下文自适应三平面方法”，载于第十三届国际学习表征会议，2025年。[32] Y. Chen、Q. Wu、W. Lin、M. Harandi和J. Cai，“HAC++：迈向3D高斯泼溅的100倍压缩”，《IEEE模式分析与机器智能汇刊》，第47卷，10 226，2025年，第11期，第10 210页–。[33] Y. Duan、F. Wei、Q. Dai、Y. He、W. Chen和B. Chen，“4D-Rotor高斯泼溅：面向动态场景的高效新视图合成”，载于ACM SIGGRAPH 2024年会议论文，丛书SIGGRAPH ’24。美国纽约州纽约：计算机协会，2024年。[34] Z. Li、Z. Chen、Z. Li和Y. Xu，“面向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（CVPR）论文集，2024年6月，第8508–8520页。[35] Z. Yang、H. Yang、Z. Pan和L. Zhang，“基于4D高斯泼溅的实时逼真动态场景表示与渲染”，载于第十二届国际学习表征会议，2024年。[36] Y. Wang、P. Yang、Z. Xu、J. Sun、Z. Zhang、Y. Chen、H. Bao、S. Peng和X. Zhou，“Freetimegs：面向动态场景重建的任意时间、任意位置自由高斯基元”，载于CVPR，2025年。[37] X. Zhang、Z. Liu、Y. Zhang、X. Ge、D. He、T. Xu、Y. Wang、Z. Lin、S. Yan和J. Zhang，“MEGA：面向动态场景的内存高效的4D高斯泼溅”，arXiv预印本 arXiv:2410.13613，2024年。[38] M. Lee、B. Lee、L. Y. Lee、E. Lee、S. Kim、S. Song、J. C. Lee、J. H. Ko、J. Park和E. Park，“优化的最简4D高斯泼溅”，2025年。[39] G. Wu、T. Yi、J. Fang、L. Xie、X. Zhang、W. Wei、W. Liu、Q. Tian和X. Wang，“面向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（CVPR）论文集，2024年6月，第20 310–20 320页。[40] Z. Yang、X. Gao、W. Zhou、S. Jiao、Y. Zhang和X. Jin，“面向高保真单目动态场景重建的可变形3D高斯”，载于2024年IEEE/CVF计算机视觉与模式识别会议（CVPR），2024年，第20 331–20 341页。[41] Z. Guo、W. Zhou、L. Li、M. Wang和H. Li，“面向高效动态场景重建的运动感知3D高斯泼溅”，《IEEE视频技术电路与系统汇刊》，第35卷，第4期，第3119–3133页，2025年。[42] H. Li、S. Li、X. Gao、A. Batuer、L. Yu和Y. Liao，“GIFStream：带特征流的基于4D高斯的沉浸式视频”，载于计算机视觉与模式识别会议论文集，2025年，第21 761–21 770页。[43] P. Dai、P. Zhang、Z. Dong、K. Xu、Y. Peng、D. Ding、Y. Shen、Y. Yang、X. Liu、R. W. H. Lau和W. Xu，“带运动分层的4D高斯视频”，《ACM图形学汇刊》，第44卷，第4期，2025年。[44] J. Ball´e、D. Minnen、S. Singh、S. J. Hwang和N. Johnston，“基于尺度超先验的变分图像压缩”，载于国际学习表征会议，2018年。[45] H. Chen、B. He、H. Wang、Y. Ren、S. N. Lim和A. Shrivastava，“NeRV：视频神经表示”，《神经信息处理系统进展》，第34卷，第21 557–21 568页，2021年。[46] C. Gomes、R. Azevedo和C. Schroers，“基于熵约束神经表示的视频压缩”，载于2023年IEEE/CVF计算机视觉与模式识别会议（CVPR），2023年，第18 497–18 506页。[47] S. R. Maiya、S. Girish、M. Ehrlich、H. Wang、K. S. Lee、P. Poirson、P. Wu、C. Wang和A. Shrivastava，“Nirvana：基于自适应网络与自回归分块建模的视频神经隐式表示”，载于2023年IEEE/CVF计算机视觉与模式识别会议（CVPR），2023年，第14 378–14 387页。[48] X. Zhang、R. Yang、D. He、X. Ge、T. Xu、Y. Wang、H. Qin和J. Zhang，“利用条件解码器增强视频神经表示”，载于IEEE/CVF计算机视觉与模式识别会议

，2024年。

[49] H. M. Kwan、G. Gao、F. Zhang、A. Gower 和 D. Bull，《NVRC：神经视频表示压缩》，神经信息处理系统进展，第 37 卷，第 132 440–132 462 页，2024年。[50] J. Wang、X. Zhang、G. Zhang、J. Zhu、L. Tang 和 L. Zhang，《 UAR-NVC：一种内存高效的神经视频压缩统一自回归框架》，*IEEE*视频技术电路与系统汇刊，第 1–1 页，2025年。[51] H. Thomas、C. R. Qi、J.-E. Deschaud、B. Marcotegui、F. Goulette 和 L. J. Guibas，《 KPConv：面向点云的灵活可变形卷积》，载于*IEEE/CVF* 国际计算机视觉会议论文集，2019年，第 6411–6420 页。[52] G. J. Sullivan、J.-R. Ohm、W.-J. Han 和 T. Wiegand，《高效视频编码（HEVC）标准概述》，*IEEE*视频技术电路与系统汇刊，第 22 卷，第 12 期，第 1649–1668 页，2012年。[53] J. Lee、C. Won、H. Jung、I. Bae 和 H.-G. Jeon，《完全显式的动态高斯泼溅》，载于神经信息处理系统论文集，2024年。[54] T. Li、M. Slavcheva、M. Zollhoefer、S. Green、C. Lassner、C. Kim、T. Schmidt、S. Lovegrove、M. Goesele、R. Newcombe 和 Z. Lv，《基于多视角视频的神经三维视频合成》，载于*2022*年 *IEEE/CVF*计算机视觉与模式识别会议（*CVPR*），2022年，第 5511–5521 页。[55] L. LI、Z. Shen、Z. Wang、L. Shen 和 P. Tan，《用于三维视频合成的流式辐射场》，载于神经信息处理系统进展，S. Koyejo、S. Mohamed、A. Agarwal、D. Belgrave、K. Cho 和 A. Oh 编，第 35 卷。Curran Associates, Inc.，2022年，第 13 485–13 498 页。[56] F. Mentzer、E. Agustsson、M. Tschannen、R. Timofte 和 L. Van Gool，《实用的全分辨率学习式无损图像压缩》，载于*IEEE*计算机视觉与模式识别会议（*CVPR*）论文集，2019年。[57] Z. Wang、A. Bovik、H. Sheikh 和 E. Simoncelli，《图像质量评估：从误差可见性到结构相似性》，*IEEE*图像处理汇刊，第 13 卷，第 4 期，第 600–612 页，2004年。[58] G. Bjontegaard，《计算 RD 曲线之间的平均峰值信噪比差异》，国际电信联盟 *SG16* 文件 *VCEG-M33*，2001年。