arXiv:2603.10551v1 [cs.CV] 11 Mar 2026

P-GSVC：用于可扩展图像与视频的分层渐进式二维高斯泼溅

Longan Wang 新加坡国立大学 新加坡
wanglongan@u.nus.edu

Yuang Shi 新加坡国立大学 新加坡
yuangshi@u.nus.edu

Wei Tsang Ooi 新加坡国立大学 新加坡
dcsooiwt@nus.edu.sg

摘要

高斯泼溅已成为一种具有竞争力的显式表示，可用于图像与视频重建。本文提出 P-GSVC，这是首个分层渐进式二维高斯泼溅框架，为图像和视频中的可扩展高斯表示提供统一解决方案。P-GSVC 将二维高斯泼溅组织为一个基础层和后续增强层，从而实现由粗到细的重建。为有效优化这种分层表示，我们提出一种联合训练策略，同时更新各层中的高斯，使其优化轨迹保持一致，以确保层间兼容性和渐进式重建的稳定性。P-GSVC 支持在质量和分辨率方面进行扩展。实验表明，与采用逐层顺序训练的方法相比，联合训练策略可使视频的峰值信噪比最高提升 1.9 dB，图像的峰值信噪比最高提升 2.6 dB。项目页面：https://longanwang-cs.github.io/PGSVC-webpage/。

CCS 概念

• **信息系统** →**多媒体流传输**。

关键词

可伸缩图像、可伸缩视频、渐进编码、二维高斯泼溅



(a) 剪枝高斯泼溅



(b) P-GSVC

图 1：(a) 剪枝高斯泼溅和 (b) 分层高斯泼溅（我们的 P-GSVC）的渐进式重建。两列使用相同的 splat 数量，随着泼溅数量从左向右增加，质量也随之提升。请注意，(a) 中出现了明显的空洞，而 (b) 保持了场景的完整性。

1 引言

可伸缩图像与视频编解码器支持以多种比特率进行渐进编码和解码，从而能够在网络带宽波动的情况下传输视频，并在异构设备上显示。三十年前，人们就开始开发渐进式编解码器，以实现这种可伸缩性 [1, 28]。通过将编码后的比特流组织为基础层和一个或多个增强层，可伸缩编码能够从基础层生成粗略重建结果，并随着解码层数的增加而逐步获得更高质量的重建结果。

传统的可伸缩标准（如用于图像的 JPEG2000 [28] 和用于视频的 HEVC-SHVC [6] ）支持分辨率和质量方面的可伸缩性。近年来，基于学习的可伸缩编解码器应运而生，它们以端到端优化的神经网络取代手工设计的模块，从而简化了编解码器设计。这些方法通常采用两种可伸缩性范式：渐进式解码分层潜在表示 [16, 19, 30], ，或将神经网络参数划分为基层和增强子网络，以实现质量可伸缩性 [7, 34]。然而，这类方法通常需要大量计算资源和训练数据。此外，隐式表示会使内容编辑和后期处理更加复杂，且难以对比特率和质量进行细粒度控制，这对于在异构网络和设备上进行自适应传输尤为不利 [21, 24]。

三维高斯泼溅（3DGS） [13]已成为一种强大的基于学习的三维场景表示范式。与 NeRF 等基于神经网络的隐式方法 [18], 3不同，3DGS 使用显式高斯泼溅作为图元，通过可微光栅化实现高保真重建和实时渲染。近期，研究者探索了 3DGS 的渐进式扩展，实现了由粗到细重建 [35], 渐进式渲染 [40], 自适应流媒体 [22], 以及可扩展压缩 [8, 23]。

高斯泼溅在三维场景重建中的成功，促使人们采用高斯泼溅作为表示二维内容（包括图像和视频）的新图元。对于图像，2DGaussian Splatting（2DGS）已被用于高效的图像表示 [39] 和压缩 [37]。对于视频，时间冗余带来了额外的压缩机会，而动态内容和场景转换等新挑战则使视频重建更加复杂。此前的尝试采用 3DGS 进行可编辑视频重建 [5, 25, 26], ，但并未聚焦于压缩。研究者还提出将 2DGS 用于视频表示和压缩 [14, 29] ，作为比 3DGS 更轻量的替代方案。值得注意的是，GSVC [29] 通过多种技术实现了最先进的质量与文件大小权衡：(i) 连续学习，即以每个 P 帧的前一帧初始化其高斯泼溅，以利用时间冗余；(ii) 高斯泼溅剪枝

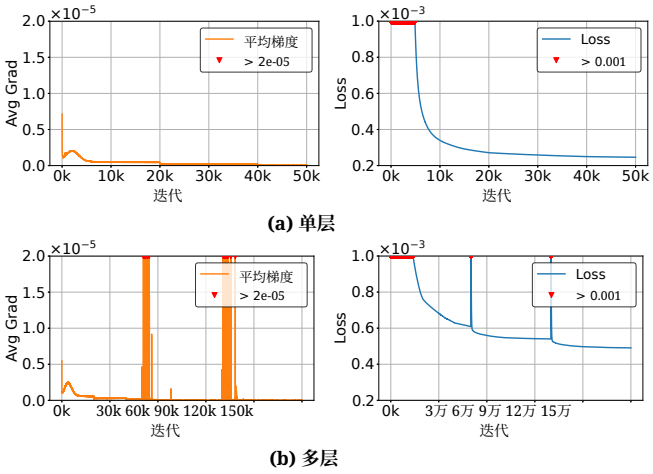


图 2：单层和多层优化策略中，梯度与损失随迭代次数的变化对比：(a) GSVC 中的单层优化；(b) GSVC 中按顺序训练的多层（在第 50k 次和第 100k 次迭代时切换目标层，同时冻结先前训练的层）。可以观察到，多层训练存在不稳定收敛和陷入次优局部极小值的问题。

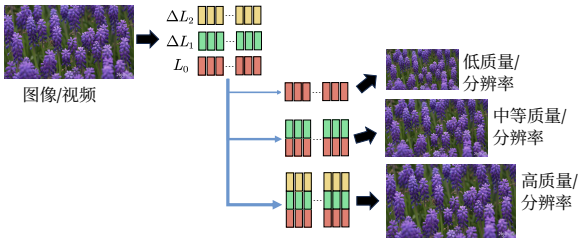


图 3：P-GSVC 的架构，以三层为例展示。基础层 L_0 提供粗略表示，而增强层 ΔL_1 和 ΔL_2 则逐步添加更多二维高斯泼溅。通过逐步加入这些增强层，视频质量得以持续提升，从而生成质量更高、分辨率更高的重建结果。

（GSP），即移除对视觉质量贡献较小的泼溅，在保持保真度的同时降低比特率；(iii) 高斯泼溅增强（GSA），即注入新的泼溅，以捕捉快速运动或新物体等动态内容；以及 (iv) 动态关键帧选择（DKS），即检测场景变化，并在需要时插入新的 I 帧。这些机制使 GSVC 能够在压缩效率和对动态场景的适应性之间取得平衡；与 AV1 和 AVC 等编解码器的现代开源实现相比，它也展现出有竞争力的率失真表现。

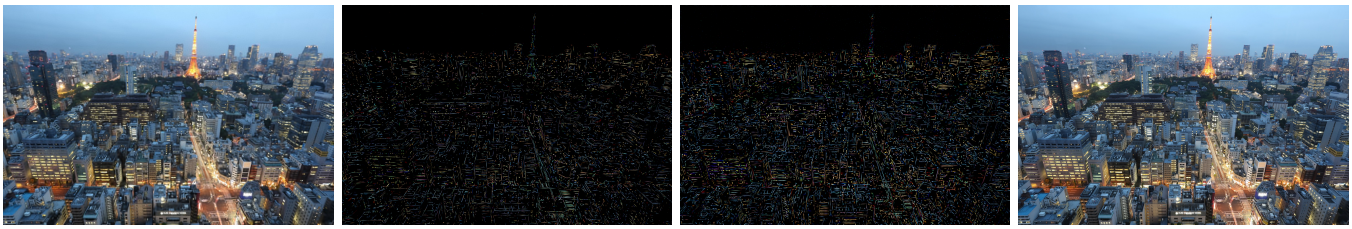
虽然 2DGS 是一种适用于图像与视频的有效表示，但将其扩展到可伸缩编码并非易事。由于图像可以视为单帧视频，为了保持一般性，我们以下将挑战以视频为例进行描述。

为了说明渐进式 2DGS 表示为何并非显而易见，我们来考虑一种简单方案：根据泼溅对渲染输出的贡献 [10, 29] 进行排序，并据此构建各层：贡献较大的泼溅构成粗糙、保真度较低的层；贡献较小的泼溅则用于细化细节，构成更高层。然而，这种方法会在较低层产生明显伪影，例如重建帧中出现显著的空隙，如图 1a 所示。根本原因在于，泼溅经过联合训练，设计上会对最高保真度的输入过拟合，因此彼此高度依赖 [21]。因此，移除任何子集，即使其中的泼溅贡献较小，也会大幅降低整体质量。

接下来，考虑 3DGS（例如 LapisGS [22]）为实现分辨率可扩展性所采用的方法，即逐层递增训练各层。当训练新层时，其所依赖的较低层会被固定（不透明度除外）。尽管这种分离层的方式在静态 3D 场景中可以得到有效实施，但将这一策略直接扩展到 GSVC 会导致训练效果不佳。具体而言，与静态表示不同，视频在各帧之间包含显著的时间动态，包括运动、光照变化和遮挡。由于 GSVC 对 P 帧使用高斯预测，这些动态增加了高斯泼溅之间的相互依赖性，不仅存在于不同层之间，也存在于时间维度上。因此，将不同层限制在彼此独立的目标上，会在优化过程中导致梯度冲突。图 2b 展示了这种跨层优化冲突：在 GSVC 下进行多层训练时，模型会出现不稳定收敛，并陷入次优的局部极小值；与图 2a 中的单层基线相比，其最终损失更高。这表明，相较于静态 3D 设置，按层优化 GSVC 更加困难。

图 2b 还突显了逐层训练中的另一项挑战。当我们在第 50k 次和第100k次迭代开始训练新的增强层时，优化目标会发生变化，训练过程中梯度会急剧增大。最终，损失趋于稳定，但增强层的损失仅略有下降。这种 渐进式不稳定性表明，较低层学习到的泼溅在作为较高层训练的起点时条件不佳，而且优化过程对层切换十分敏感。因此，它难以提供高质量的中间重建结果，削弱了渐进式可扩展性的一项关键要求。

本文解决了训练渐进式、可伸缩 GSCV 版本所面临的挑战。我们提出渐进式高斯泼溅视频编码（P-GSVC），这是一种以二维高斯泼溅作为图元的渐进式可伸缩视频编解码器。P-GSVC 同样适用于可伸缩图像编码，将图像视为单帧视频。图3展示了 P-GSVC 的概览，其中输入视频由多个高斯泼溅层表示。与典型的分层渐进式编解码器一样，仅解码基础层可重建出输入的低保真度版本，而解码额外的增强层则可提高保真度。图4分别展示了各层泼溅的渲染效果。为避免跨层优化冲突并确保渐进过程平滑，P-GSVC 采用联合训练策略。在每次迭代中，它同时优化两个保真度级别：一个使用最高层及以下的所有泼溅，



(a) $\mathcal{G}^0$ (b) $\Delta\mathcal{G}^1$ (c) $\Delta\mathcal{G}^2$ (d) $\mathcal{G}^2$

图4：分层高斯的渐进过程。 $\Delta\mathbf{G}^1$ 和 $\Delta\mathbf{G}^2$ 捕获更高频的细节，并可迭代地添加到$\mathbf{G}^0$, 中，从而逐步细化$\mathbf{G}^2$。

另一个仅使用某一层 i及以下的泼溅。在训练过程中，目标层 i 从基础层轮换至倒数第二层。通过这种联合优化，各层泼溅的优化轨迹得以对齐，同时每层的泼溅仍会针对该层的目标进行训练。此外，通过轮换 i，在不同优化目标之间切换时，梯度场能够保持相对稳定，从而确保最终重建和中间重建都能收敛至可接受的质量。

本文的主要贡献总结如下：

- 我们提出了一种基于 2DGS 的可伸缩渐进表示框架，适用于图像与视频。
- 我们证明了联合训练策略在实现 2DGS 渐进性方面的有效性。
- 我们将 P-GSVC 与现有的 3DGS 和 2DGS 训练方法进行了评估比较。结果表明，与采用逐层顺序训练的方法相比，P-GSVC 在 UVG 视频数据集上的峰值信噪比最高可提升 1.9 dB，在 DIV-HR 图像数据集上的峰值信噪比最高可提升 2.6 dB。

2 相关工作

2.1 图像与视频的可伸缩编码

可伸缩编码将视觉内容（如图像或视频）编码为单一比特流，支持解码出多个版本，从而能够根据不同设备和变化的网络条件进行自适应传输。在可伸缩图像编码中，JPEG 等传统方法采用基于小波的层级结构，在单一比特流中支持分辨率和质量可伸缩性。基于类似原理，可伸缩视频编码引入时间依赖关系以及跨帧的层间预测。MPEG- 等早期标准使用基本层和增强层引入分层的空间和质量可伸缩性。MPEG-4 Visual 通过细粒度可伸缩性（FGS）扩展了质量可伸缩性，使增强层比特流能够渐进式截断。随后，H.264/AVC 的可伸缩扩展正式确立了一种结构化且高效的多层框架，支持空间、时间和质量可伸缩性；HEVC-SHVC 则进一步简化并扩展了该框架，使其支持位深和色域可伸缩性。尽管有这些改进，这些技术在实际中的应用仍然有限，部分原因在于编解码器复杂度较高。

随着深度学习兴起，基于学习的方法应运而生，通过端到端优化简化了处理流程。这些方法通常遵循两种范式。一种是在潜在空间中表示图像或视频，再由分层解码器以不同尺度采样，以重建不同质量的内容并实现渐进式解码。另一种是将图像或视频直接编码为网络参数，其中基本子网络和增强子网络对应不同的质量级别，因此使用更多参数即可提升质量。然而，这两种范式都依赖隐式表示，妨碍了内容处理以及传统可伸缩编解码器功能的实现，同时还会带来较高的解码成本，限制了实际部署。相比之下，我们的 P-GSVC 利用一种更显式的基本元素——二维高斯泼溅，通过添加或移除泼溅实现渐进式可扩展性。

2.2 图像与视频的高斯泼溅

高斯泼溅使用显式的二维或三维高斯基元表示视觉内容，其保真度可通过高斯泼溅的数量和参数轻松控制。对于图像，Gaussian- Image [37] 使用二维高斯泼溅进行图像表示与压缩，而 LIG [39]则将其扩展到高质量大图像表示。对于视频，现有方法可分为三维和二维两类。在三维高斯视频领域，VGR [26] 将视频视为一系列静态三维场景，并使用 3DGS 独立重建每一帧。相比之下，VeGaS [25] 和 GaussianVideo [5]将整个序列视为连续的时空体，其中泼溅随时间平滑演变。这些三维方法擅长视频表示与编辑，但并非专为压缩或可伸缩编码设计。相比之下，二维高斯视频使用轻量级的 2DGS [37] 基元对每一帧进行编码，从而实现视频表示与压缩 [14, 29]。然而，正如第1节所述，将二维高斯表示扩展至可伸缩编码并非易事。在这项工作中，我们通过将高斯泼溅重新组织为多层结构来扩展 2DGS，从而实现渐进式、可伸缩的图像与视频编码。

2.3 渐进式高斯泼溅

渐进式高斯泼溅以由粗到细的方式重建内容：一部分泼溅即可生成低保真预览，增加泼溅则可逐步优化细节。渐进式高斯泼溅的主要优势在于支持部分

解码，从而快速生成低保真预览，并逐步优化至完整质量，因此能够天然适应不同的带宽和计算资源，具有可扩展性和适应性。渐进式高斯泼溅最早由 FreGS [35] 探索，随后在 3DGS 的背景下扩展至渲染（PRoGS [40]）、流式传输（LapisGS [22] 及其动态扩展 [27]）和压缩（PCGS [8]）。相比之下，更适合图像与视频传输的轻量级 2DGS 渐进编码尚未得到充分探索。首次尝试是 LIG [39], ，它通过分别训练骨架层和残差层，利用双层高斯结构重建图像。尽管 LIG 对大图像效果良好，但它优先考虑最终重建质量，而非可扩展编码：要达到良好的视觉质量，需要大量泼溅来呈现高频细节，因此在带宽有限、只能传输少量内容时，性能会显著下降。此外，如第1节所述，其逐层顺序训练策略会导致各层之间的优化目标相互冲突。

3 P-GSVC

3.1 问题表述

我们首先介绍适用于图像与视频的渐进式、可扩展高斯表示的问题表述。我们区分两个术语："层"指分层表示中的结构组件，而"级"指通过累积各层形成的重建图像或视频。令 $\mathcal{F} = \{f_1, f_2, \ldots, f_T\}$ 表示输入序列，其中 T 为帧数（对于图像，$T = 1$）。具体而言，给定 L 层，我们的目标是将 $\mathcal{F}$ 编码为分层表示：$\widehat{\mathcal{F}} = \widehat{\mathcal{F}}^0 \cup \left(\bigcup_{\ell=1}^{L} \Delta\widehat{\mathcal{F}}^\ell\right)$，其中 $\widehat{\mathcal{F}}^0$ 既作为基础层，也作为基础级，而 $\Delta\widehat{\mathcal{F}}^\ell (\ell = 1, \ldots, L)$ 则为增强层。对于任意 $\ell \in \{1, 2, \ldots, L\}$，级别 ℓ 上的重建通过将前 ℓ 个增强层逐步添加到基础级而得到：

$$\widehat{\mathcal{F}}^\ell = \widehat{\mathcal{F}}^0 + \sum_{i=1}^{\ell} \Delta\widehat{\mathcal{F}}^i. \tag{1}$$

在 P-GSVC 中，$\mathcal{F}$ 由一组二维高斯基元 $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T\}$ 表示，以便在保留可伸缩性的同时重建 $\mathcal{F}$。请注意，P-GSVC 同时支持可伸缩视频（$T > 1$）和图像（$T = 1$）。

3.2 二维高斯泼溅

我们采用二维高斯泼溅 [37] 作为框架中的基本图元。每个高斯泼溅 G 由一个元组 $G = (\boldsymbol{\mu}, \Sigma, \mathbf{c}')$ 参数化，其中 $\boldsymbol{\mu} \in \mathbb{R}^2$ 表示其在图像平面上的中心，$\Sigma \in \mathbb{R}^{2\times2}$ 是控制尺度和方向的协方差矩阵，而 $\mathbf{c}' \in \mathbb{R}^3$ 是与之关联的 RGB颜色和不透明度的学习表示。为确保 Σ 为半正定矩阵，我们采用乔列斯基分解 [12]，将其表示为 $\Sigma = \boldsymbol{L}\boldsymbol{L}^\top$，其中 $\boldsymbol{L}$ 是由乔列斯基向量 $\boldsymbol{\ell} = (\ell_1, \ell_2, \ell_3)$ 参数化的下三角矩阵。总之，每个二维高斯泼溅仅需八个参数，因此表示紧凑且高效。对于渲染，第 i 个像素的颜色 C_i 通过以下方式计算：

覆盖该像素的所有泼溅的加权累积：

$$C_i = \sum_{n\in N} \boldsymbol{c}'_n \exp\left(-\tfrac{1}{2}\boldsymbol{d}_n^\top \Sigma_n^{-1} \boldsymbol{d}_n\right), \tag{2}$$

其中，N 是与像素 i 重叠的泼溅集合，$\boldsymbol{d}_n$ 表示像素中心与第 n 个高斯的中心之间的距离。

在 P-GSVC 中，我们进一步遵循 GSVC [29],中的渲染方案，其中引入了额外的权重向量 $\boldsymbol{w}$ 来跟踪高斯泼溅的贡献。这样便可根据泼溅的贡献动态添加或移除泼溅，从而在质量与大小之间进行权衡，并增强对高动态内容的处理能力。因此，公式（2）中的渲染公式可写为：

$$C_i = \sum_{n\in N} w_n \boldsymbol{c}'_n \exp\left(-\tfrac{1}{2}\boldsymbol{d}_n^\top \Sigma_n^{-1} \boldsymbol{d}_n\right). \tag{3}$$

此外，2DGS 具有适用于可伸缩编码的优势。首先，其表示本身与分辨率无关：固定的一组泼溅无需重新训练即可在任意分辨率下渲染（不过，如果渲染分辨率与训练分辨率相差很大，质量可能会下降）。其次，其显式且模块化的结构支持通过独立解码泼溅子集来实现可伸缩性。

3.3 渐进式高斯泼溅

为实现渐进式可扩展性，我们将每个输入（视频或图像）表示为由多个高斯集合构成的分层集合。对于基础级表示，$\widehat{\mathcal{F}}^0$，每一帧 $\widehat{f}_t^0$ 都与一个高斯集合相关联：$\mathcal{G}_t^0 = \{G_{t,1}, G_{t,2}, \ldots, G_{t,N^0}\}$，其中 N^0 表示

每帧基础层中的泼溅数量 . 增强层 $\Delta\widehat{\mathcal{F}}^\ell$（$\ell = 1, \ldots, L$）的定义方式类似，每帧由 $\Delta\mathcal{G}_t^\ell = \{G_{t,1}, G_{t,2}, \ldots, G_{t,N^\ell}\}$ 表示，其中 N^ℓ 表示每帧第 ℓ 层中的泼溅数量。随后，通过合并基础层和前 ℓ 个增强层高斯集合，即可得到第 ℓ 级的帧分层高斯表示：

$$\mathcal{G}_t^\ell = \mathcal{G}_t^0 \cup \left(\bigcup_{i=1}^{\ell} \Delta\mathcal{G}_t^i\right). \tag{4}$$

相应地，$\widehat{\mathcal{F}}^\ell$ 表示为 $\mathcal{G}^\ell = \{\mathcal{G}_1^\ell, \ldots, \mathcal{G}_T^\ell\}$。其重建通过可微光栅化器 R(·) 渲染这些高斯集合得到：

$$\widehat{\mathcal{F}}^\ell = \mathcal{R}(\mathcal{G}^\ell), \qquad \widehat{f}_t^\ell = \mathcal{R}(\mathcal{G}_t^\ell). \tag{5}$$

算子 R(·) 遵循公式（3）中的逐像素累积规则，并将其应用于合并后的高斯集合 $\mathcal{G}_t^\ell$。

这种分层构造支持渐进式解码：基础集 $\mathcal{G}_t^0$ 提供粗略但完整的表示，而每个 $\Delta\mathcal{G}_t^i$ 则增添更精细的细节。因此，增加 ℓ 可单调提升质量，而无需修改先前传输的泼溅。如图4所示，$\mathcal{G}^0$ 生成粗略重建，$\Delta\mathcal{G}^1$ 和 $\Delta\mathcal{G}^2$ 依次细化细节，而它们的并集 $\mathcal{G}^2 = \mathcal{G}^0 \cup \Delta\mathcal{G}^1 \cup \Delta\mathcal{G}^2$ 则生成更高质量的结果。

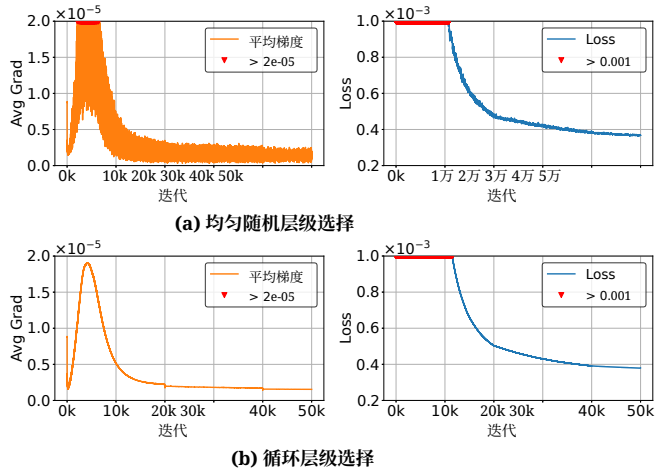


图 5：随机与循环联合训练策略下梯度和损失随迭代的变化对比：以随机顺序 (a) 和循环顺序 (b) 联合优化各层。

3.4 联合训练

优化分层泼溅的一种基线方法是逐层顺序训练。具体而言，对于层级 ℓ，重建可写为

$$\widehat{\mathcal{F}}^{\ell} = \widehat{\mathcal{F}}^{\ell-1} + \mathcal{R}(\Delta\mathcal{G}^{\ell}), \tag{6}$$

其中，$\widehat{\mathcal{F}}^{\ell-1}$ 表示前面各层的重建结果，$\mathrm{R}(\Delta\mathcal{G}\ell)$表示第 ℓ 层中新引入泼溅的渲染结果。训练过程首先对基础集 $\mathcal{G}^0$, 进行优化，然后冻结其参数，随后依次训练增强集 $\Delta\mathcal{G}^1, \Delta\mathcal{G}^2, \ldots, \Delta\mathcal{G}^L$。这种策略类似于 LIG [39], , 后者使用两个顺序式训练的高斯层表示大图像，并冻结第一层，将其作为第二层的基础。

然而，这种逐层训练方法会导致若干问题。首先，由于不同层追求不同的优化目标，其优化过程遵循不同的轨迹。当在已经收敛的低层之上训练更高层时，优化必须适应不同的目标景观，从而阻碍收敛。如图 2b 所示，在 50k 和 100k 迭代时切换目标层会导致梯度剧烈波动，表明模型难以适应目标的变化。与图 2a 中的单层 GSVC 优化相比，逐层训练最终收敛到更高的损失值，说明模型陷入了次优的局部极小值。

为了有效优化分层泼溅，需要考虑两个关键挑战：跨层优化冲突和渐进式稳定性。

跨层优化冲突。P-GSVC 需要一个协调一致的多层解决方案，但这并不容易，因为不同层追求不同的目标：基础层捕获粗略结构，而增强层则细化细节。为提高各层优化目标之间的兼容性，我们采用联合损失函数，对多个层级的层进行联合训练，而非分别训练，从而使不同层中的泼溅能够沿着一致的轨迹进行优化。

这一过程从一开始就促使每一层中的泼溅彼此兼容。以第 t 帧 f_t 为例，在每次迭代中，我们都会监督完整重建 $\widehat{f}_t^{L}$ 和中间重建 $\widehat{f}_t^{\ell}$。形式上，损失函数定义为所选层级上的 L_2 损失之和：

$$\mathcal{L}_t = \sum_{\ell\in\mathcal{S}} \mathcal{L}_2(\widehat{f}_t^{\ell}, f_t^{\ell}) = \sum_{\ell\in\mathcal{S}} \frac{1}{HW} \sum_{i=1}^{HW} \left\|\widehat{f}_{t,i}^{\ell} - f_{t,i}^{\ell}\right\|^2, \tag{7}$$

其中，$\widehat{f}_{t,i}^{\ell}$ 表示 f_t 在第 ℓ层重建的第 i个像素，H和W分别表示空间维度，S 是当前迭代中联合优化的层的集合。实际应用中，S 包含最终层和一个中间层 $\ell \in \{1, 2, \cdots L-1\}$。每一层都由对应分辨率下的真实值进行监督。这种设置确保较低层专注于粗略结构，而较高层则细化精细细节，并防止较低层出现混叠 [39]。损失可表示为：

$$\mathcal{L}_t(\ell) = \mathcal{L}_2(\widehat{f}_t^{L}, f_t^{L}) + \mathcal{L}_2(\widehat{f}_t^{\ell}, f_t^{\ell}), \tag{8}$$

其中，$f_t^{\ell} \in \mathbb{R}^{H_\ell \times W_\ell}$是将相应的真实帧下采样至分辨率（$H_\ell$，$W_\ell$）后的结果。

渐进式稳定性。为了实现渐进式可扩展性，中间层的重建质量必须保持较高水平，理想情况下应与具有相同泼溅数量的非分层版本相当。为此，我们采用循环联合训练，在迭代训练过程中，按照递增的循环顺序选择中间层，而不是均匀随机地选择。

朴素的均匀随机层级选择可能导致连续选择同一层级，从而造成过拟合；也可能有若干次迭代都未选择某一层级，导致其与最高层级重建的优化过程不对齐。如图 5a 所示，随机联合训练会导致梯度不稳定、损失更新不规则。相比之下，循环选择会强制以固定周期切换中间层级，避免出现空闲间隔和连续重复，因此能带来更平滑的优化轨迹，以及更稳定的梯度和损失，如图 5b 所示。

最后，第 t 帧 $f_t \in \mathcal{F}$在第 k 次迭代时的损失定义如下：

$$\mathcal{L}_t(\ell_k) = \mathcal{L}_2(\widehat{f}_t^{L}, f_t^{L}) + \mathcal{L}_2(\widehat{f}_t^{\ell_k}, f_t^{\ell_k}), \tag{9}$$

其中，$\ell_k \equiv k (\mathrm{mod}(L-1))$ 表示按循环顺序选取的中间层级。

3.5 整合所有内容

在 P-GSVC 中，为了实现高效的 2DGS 表示，我们借鉴了 GSVC 的核心范式 [29]。对于视频，帧预测利用时间冗余，根据前一帧预测每个 P 帧；高斯泼溅剪枝（GSP）移除贡献较低的泼溅，以平衡质量与比特率；高斯泼溅增强（GSA）引入新的泼溅，以捕捉动态程度较高的区域；动态关键帧选择（DKS）则检测场景转换，自适应地插入 I 帧。对于图像，我们将其视为一个没有时间冗余的单帧 I 帧视频。

P-GSVC 分辨率可扩展性的完整训练流程总结于算法 1 中。

算法 1 P-GSVC训练流程

要求： 输入序列 $\mathcal{F}=\{f_1、f_2、\ldots\ldots、f_T\}$，层数 L，泼溅数量 (N, N_{aug}, N_{prune})，GSP配置 $(Z_{GSP}、K_{GSP})$，目标分辨率 $\{(H_\ell、W_\ell)\}_{\ell=0}^{L}$。

1：通过对给定的 $\mathcal{F}\{(H_\ell、W_\ell)\}_{\ell=0}^{L}$ 进行下采样，获得 $\{\mathcal{F}^\ell\}_{\ell=1}^{L}$。

2: **如果** $T>1$ **则**

3: 使用 DKS 选择 I帧。

4: **结束 if**

5: **对于** $t=1$ 到 T **执行**

6: **如果** $T>1$ (视频) 且 t 是 I 帧 **或** $T=1$ (图像) **则**

7: 使用 $N+N_{aug}$ 个随机高斯斑点初始化 $\mathcal{G}_t=\{\mathcal{G}_t^0, \Delta\mathcal{G}_t^1、\ldots\ldots、\Delta\mathcal{G}_t^L\}$（均匀分布在所有层上）。

8: **else**

9: 初始化 $\mathcal{G}_t=\{\mathcal{G}_t^0, \Delta\mathcal{G}_t^1, \ldots, \Delta\mathcal{G}_t^L\}$ 其中 $\mathcal{G}_t^0=\mathcal{G}_{t-1}^0$ 和 $\Delta\mathcal{G}_t^i=\Delta\mathcal{G}_{t-1}^i$ 对于 $i=1\ldots L$。

10: 注入 N_{aug} 个随机高斯斑点（均匀分布在所有层上）。

11: **结束条件判断**

12: **当** 未收敛时，迭代 $k=0$，1、2、.....**执行**

13: 循环选择 ℓ_k：$\ell_k\equiv k(\mathrm{mod}(L-1))$。

14: 根据公式（5）逐步渲染 $\widetilde{f}_t^{\ell_k}$ 和 $\widetilde{f}_t^{L}$。

15: 根据公式（9）计算联合损失，并更新 $\mathcal{G}_t$。

16: 在前 K_{GSP} 次迭代中，每隔 Z_{GSP} 次迭代，根据公式（3）中的 w，移除贡献最小的 $N_{\mathrm{prune}}\cdot {}^{Z}\!/_{\mathrm{GSP}K_{\mathrm{GSP}}}$ 泼溅。

17: **结束 while 循环**

18: **结束 for 循环**

确保： 优化后的分层泼溅 $\mathcal{G}=\{\mathcal{G}^0, \Delta\mathcal{G}^1, \ldots\ldots, \Delta\mathcal{G}^L\}$。

首先，根据预定义的目标分辨率 $\{(H_\ell、W_\ell)\}_{\ell=0}^{L}$ 对输入序列 F 进行下采样，构建多分辨率图像（$T=1$）或视频（$T>1$）序列（第 1 行）。如果输入是视频序列（$T>1$），则使用动态关键帧选择（第 2–4 行）来选取 I帧。

对于每一帧 f_t（第 5–11 行），初始化过程取决于输入是否为视频，以及当前帧是否为 I 帧；如果输入是图像，初始化过程也有所不同。如果 f_t 是 I 帧，或输入是图像，我们会用 $N+N_{\mathrm{aug}}$ 随机高斯斑点初始化分层高斯泼溅 $\mathrm{G}_t=\{\mathrm{G}_t^0, \Delta\mathrm{G}_t^1, \ldots\ldots, \Delta\mathrm{G}_t^L\}$（第 6–7 行）。否则，我们使用上一帧的泼溅 G_{t-1} 初始化 G_t，并注入 N_{aug} 随机高斯斑点（第 8–10 行），使其均匀分布于所有层中。

在每次迭代 k（第 12 行）中，根据 $\ell_k\equiv k(\mathrm{mod}(L-1))$ 循环选择目标分辨率级别 ℓ_k（第 13 行）。根据公式（5），通过渲染前 ℓ_k 个高斯层，得到分辨率级别为 ℓ_k 的渲染输出（第 14 行）。根据公式（9）计算联合损失，并相应地更新分层高斯参数（第 15 行）。为减少冗余，会定期移除贡献极小的高斯泼溅（通过公式（3）中的重要性权重 w 进行度量）。具体而言，在前 K_{GSP} 次迭代中，每经过 Z_{GSP} 次迭代，就会剪除 $N_{\mathrm{prune}}\cdot {}^{Z}\!/_{\mathrm{GSP}K_{\mathrm{GSP}}}$ 泼溅（第 16 行）。针对当前第 t帧，优化会持续进行直至收敛（第 12–17 行）。最后，得到优化后的分层高斯表示 $\mathrm{G}=\{\mathrm{G}^0, \Delta\mathrm{G}^1, \ldots\ldots, \Delta\mathrm{G}^L\}$。

3.6 量化

我们采用 GSVC [29], 的时间感知微调和量化策略，同时保留渐进式泼溅的分层结构，并采用联合优化。对于 I 帧，所有基础层和增强层中的高斯参数均直接进行量化。对于 P 帧，我们对相对于参考帧的参数差值进行量化。

对每个参数进行量化：位置 μ 以低精度浮点数 [39]存储。具体而言，I 帧使用 16 位浮点数，P 帧使用 12 位浮点数。对于乔列斯基向量 ℓ，我们应用 b位非对称量化 [4]:

$$\begin{aligned} Q_\ell(\ell_i)&=\hat{\ell}_i\,\gamma_i+\beta_i \\ \hat{\ell}_i&=\left\lfloor \mathrm{clamp}\left(\frac{\ell_i-\beta_i}{\gamma_i}, L_{\min}, L_{\max}\right)\right\rfloor, \\ L_{\max}-L_{\min}&=2^b-1. \end{aligned} \tag{10}$$

其中 $i\in\{0, 1, 2\}$，γ_i和 β_i分别表示以 32 位浮点格式存储的可学习缩放和偏移量。对于 I 帧，基础层量化为无符号整数，采用 $[L_{\min}, L_{\max}]=[0, 2^b-1]$,；增强层则量化为有符号整数，采用 $[L_{\min}, L_{\max}]=[-2^{b-1}, 2^{b-1}-1]$，其中 b设为 6。对于 P 帧，基础层和增强层均量化为有符号整数，采用 $[L_{\min}, L_{\max}]=[-2^{b-1}, 2^{b-1}-1]$，其中 b设为 5。对于加权颜色 c，采用残差矢量量化（VQ） [33]，堆叠 M 个矢量量化阶段 [11]，码本大小为 B:

$$\begin{aligned} Q_c(c')&=\textstyle\sum_{k=1}^{M} C^k[i^k], \\ i^k&=\arg\min_k \left\| C^k[k]-\left(c'-\sum_{j=1}^{k-1} C^j[i^j]\right)\right\|_2^2, \end{aligned} \tag{11}$$

其中，$C^k\in\mathbb{R}^{B\times 3}$ 是第 k 个码本，i^k 是其索引。对于 I 帧和 P 帧，B均设为 6 位。码本通过承诺损失 [37]进行优化。

4 实验

4.1 实验设置

数据集。对于图像实验，我们采用 Kodak 数据集 [9]，其中包含 24 张分辨率为 768×512 的自然图像；以及 DIV2K 验证集 [3]，其中包含 100 张 2K 级别的高质量图像。这两个数据集涵盖了图像重建场景中不同的分辨率。对于视频实验，我们使用 UVG 数据集 [17] 中所有 1080p 视频。每段视频均统一为 1920×1080 分辨率、4:2:0 色度子采样、8 位深度和 YUV 色彩空间。在实验中，我们使用每段视频的前 50 帧。

我们进行两组实验，一组针对分辨率可扩展性，另一组针对质量可伸缩性。对于分辨率可扩展性，我们根据 splat数量 N^i 对数据集中的图像和视频进行下采样，生成具有多个分辨率级别的图像和视频，其中级别 l 的比率定义为 $(\sum_{i=0}^{l} N^i/N)^{1/2}$。每个级别用于评估的真实值与渲染输出降低后的分辨率相匹配。对于质量可伸缩性，每个级别用于评估的真实值均为全分辨率版本。

实现。我们的 P-GSVC 框架基于 gsplat 库 [32] 实现，采用基于 CUDA 的光栅化与累积混合 [37]。高斯参数使用 Adan 优化器 [31] 进行优化，初始学习率为 1×10^{-3}，每 20,000 步减半。我们采用三层（$L = 3$），泼溅在各层间均匀分布（$\boldsymbol{N} = [\frac{N}{3}, \frac{N}{3}, \frac{N}{3}]$），并通过将泼溅总数 N从 12k 到 27k、步长为 3k 进行变化来应用恒定速率编码。移除比率在基础层设为 0.2，在增强层设为 $\boldsymbol{N} \odot [0.2, 0.4, 0.4]$ 0 4 Z_{GSP} $\boldsymbol{N}_{aug} = \boldsymbol{N}_{prune} =$ 。4,000,

（）。对于 I 帧，设为，对于 P 帧，K_{GSP} 设为 1,000。Z_{GSP} 设为 100。DKS、GSP 和 GSA 的其他设置遵循 GSVC [29]。训练持续至收敛，收敛定义为联合损失变化小于 1×10^{-7} 的 100 次迭代，所有实验均在 NVIDIA H100 GPU 上进行。

对比方法。我们分别将 P-GSVC 与三种替代方法进行了比较：

- 剪枝。训练一个完整规模的高斯模型，并根据公式（3）中基于 $\boldsymbol{w}$ 的贡献，通过剪枝泼溅获得较低层级。这种朴素方法作为可伸缩编码的基线方法。
- 单体式。针对每个层级分别独立训练高斯模型，并逐步增加高斯数量。此方法不可伸缩，因为每个层级都需要一组独立的泼溅，而非通过渐进式构建得到。在我们的实验中，该方法作为质量上界。
- 顺序式。按顺序训练分层高斯模型：冻结较低层的参数并传递给下一个较高层，在该层添加并优化增强高斯基元。此策略采用与 P-GSVC 相同的分层结构，但各层独立训练，不进行联合优化。该方法常用于分层 3DGS [22] 和 2DGS [39]，并作为与我们的联合训练策略进行比较的最先进基线方法。

为全面展示 P-GSVC 的有效性，我们针对图像和视频案例开展了三组实验。首先，我们在图像数据集上评估联合训练策略，由于不存在时间依赖性，因此可以将其与逐层顺序训练进行独立比较。具体而言，对于顺序式方法，我们采用 LIG [39]，这是目前最先进的图像分层二维高斯训练框架，同样遵循顺序优化方案。为确保公平，所有方法均采用相同的三级配置，每层的高斯预算相等，同时其他参数保持一致。其次，我们在视频数据集上评估可扩展性。由于图像可视为视频的一种特殊情况（单个 I 帧序列），我们将框架扩展至完整视频序列，以考察质量和分辨率可扩展性。我们在相同设置下比较剪枝、单体式、顺序式方法与 P-GSVC。最后，我们在视频数据集上评估量化条件下的压缩性能，考察率失真权衡，其中单体式模型作为上界，并将 P-GSVC 与顺序式基线进行比较。

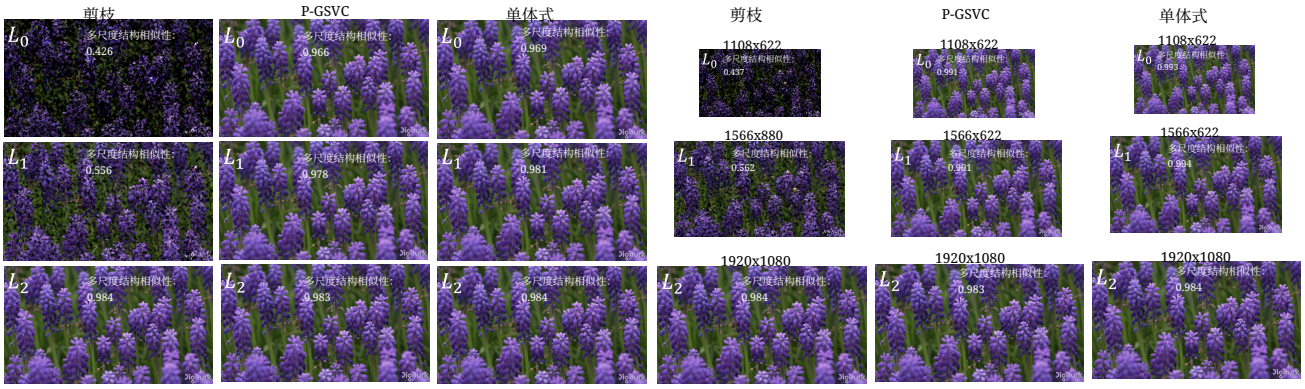
表 1：在不同泼溅数量下，P-GSVC 与序列式方法（LIG）在 Kodak 和 DIV-HR 数据集上的视觉质量比较。

数据集		Kodak			DIV-HR		
泼溅数量		5×10^3	7×10^3	9×10^3	5×10^4	7×10^4	9×10^4
峰值信噪比 ↑	LIG	26.6	27.3	28.1	27.8	28.7	29.5
	P-GSVC	**28.5**	**29.4**	**30.2**	**30.2**	**31.3**	**32.1**
多尺度结构相似性 ↑	LIG	0.918	0.937	0.948	0.944	0.958	0.966
	P-GSVC	**0.943**	**0.957**	**0.965**	**0.967**	**0.975**	**0.981**
学习感知图像块相似度 ↓	LIG	0.436	0.380	0.339	0.383	0.325	0.284
	P-GSVC	**0.271**	**0.223**	**0.189**	**0.217**	**0.178**	**0.150**

指标。我们使用峰值信噪比（PSNR）和多尺度结构相似性（MS-SSIM）评估重建质量。PSNR 以分贝（dB）为单位，通过比较信号功率与重建误差来量化像素级保真度，数值越高表示质量越好。MS-SSIM 衡量多个尺度上的感知相似度，取值范围为 0 到 1，越接近 1 表示结构相似度越高。对于图像，我们还使用学习感知图像块相似度（LPIPS）指标 [36]，通过比较从预训练神经网络中提取的深层特征表示来评估感知相似度。LPIPS 的取值通常为 0 到 1，数值越低表示感知相似度越高、视觉质量越好。对于视频，我们还使用视频多方法评估融合（VMAF）指标 [15]，这是一种基于学习的感知质量模型，通过结合多项质量指标来预测人类主观评价。VMAF 分数通常为 0 到 100，数值越高表示感知视觉质量越好。

4.2 结果与分析

图像。我们在 Kodak 和 DIV-HR 数据集上，将 P-GSVC 的联合训练策略用于图像表示任务，并与顺序式逐层训练方法进行比较。按照前文所述的设置，我们将 P-GSVC 与 LIG [39] 实现的顺序方法进行比较；后者是当前最先进的框架



（a）质量可伸缩性　　（b）分辨率可扩展性

图 6：从 *HoneyBee* 中采样的一帧在三个层级 L_0、L_1 和 L_2 下的渲染结果。(a) 展示质量可伸缩性，(b) 展示分辨率可扩展性。P-GSVC 消除了剪枝方法产生的伪影，取得了与单体方法相当的结果。

用于图像的分层 2D 高斯训练。LIG [39] 采用分层高斯结构，依次训练骨架层和残差层。结果汇总于表 1。

我们首先报告在不同总高斯预算下，Kodak 和 DIV-HR 数据集上的最高层级图像重建性能。后续实验将进一步考察单个层的性能及其渐进式贡献。

结果表明，在 Kodak 和 DIV-HR 两个数据集上，P-GSVC 在所有高斯预算下均取得了更高的峰值信噪比和多尺度结构相似性，证实了联合训练在优化分层高斯表示方面的优势。具体而言，P-GSVC 在 Kodak 上将峰值信噪比提升约 1.9–2.1 dB，在 DIV-HR 上提升约 2.4–2.6 dB，同时在多尺度结构相似性和学习感知图像块相似度方面也取得了稳定提升。总体而言，P-GSVC 始终优于 LIG，证明在图像表示任务中，我们的联合训练策略比顺序逐层方法更有效。

视频。由于图像可以视为仅包含单个 I 帧的序列，我们进一步将实验扩展到视频数据集，以评估 P-GSVC 的质量可伸缩性和分辨率可扩展性。

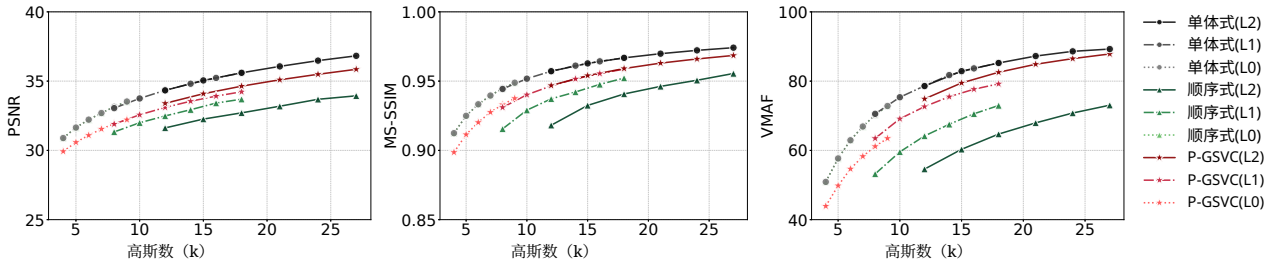
我们首先对三个层级的输出进行定性比较。如图 6 所示，在质量和分辨率可扩展性任务中，当仅解码 L_0 and L_1 中的部分泼溅时，剪枝方法会产生严重伪影，例如空洞和破碎结构。这些缺失区域直接表明，仅按贡献分数排序会丢弃负责粗略结构的泼溅。贡献分数只能反映每个高斯的整体重要性，却无法区分其承载的是粗略结构还是精细细节。因此，仅选择贡献最高的泼溅无法在较低层级保留完整的重建结果，导致视觉质量严重下降。相比之下，P-GSVC 将泼溅组织为分层结构并进行联合优化，从而在 L_0 and L_1 中实现无伪影的中间重建结果。此外，与单体方法相比，P-GSVC 在所有层中都能实现相当的视觉质量，表明它在仅牺牲少量质量的情况下保留了可扩展性。

由于剪枝会导致中间质量严重下降并产生伪影，因此无法作为有意义的基线，下面的实验结果中不再考虑该方法。

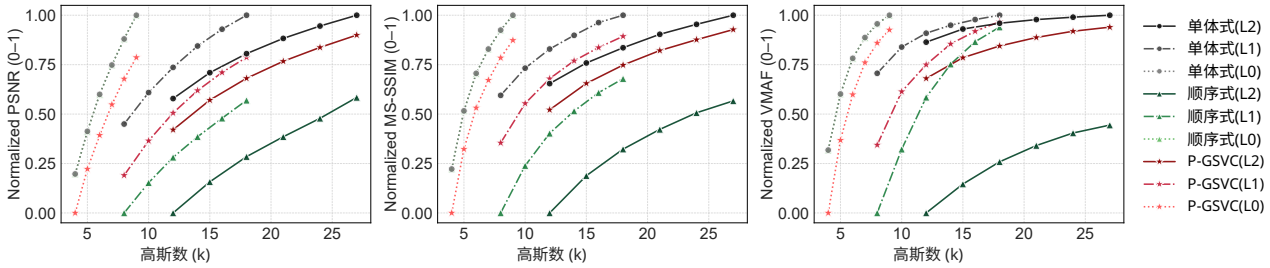
接下来，我们在 UVG 数据集上，从峰值信噪比、多尺度结构相似性和视频多方法评估融合这几个指标出发，展示 P-GSVC 的可伸缩性实验结果，同时考虑图7所示的质量可伸缩性和分辨率可扩展性。

图7a展示了 UVG 数据集上以峰值信噪比和多尺度结构相似性衡量的质量可伸缩性结果。横轴表示泼溅数量，该数量与模型大小正相关；纵轴表示重建质量。结果揭示了三个关键观察：（i）对于所有方法，随着添加更多泼溅，更高层始终能够提升质量。（ii）对于顺序方法而言，基础层达到了与单体方法相近的质量，这是因为其优化不受更高层约束，因此遵循与上界相同的训练过程。然而，尽管基础层相对较强，增强层却无法得到有效优化，这是由于不同层之间存在相互冲突的优化目标。结果是，训练陷入次优的局部极小值，这反映在各层之间较小的质量增益上。例如，如图所示，顺序方法的最高质量水平仍远低于 L_1 单体方法的水平，且 L_2 之间的差距超过 2.2 dB。（iii）相比之下，P-GSVC 实现了比顺序方法更高效的可伸缩性。尽管其基础层较弱（低< 3%），但在更高层级上取得了显著更大的提升，其中 L_1 比顺序方法高出约 0.5 dB，而 L_2 则高出近 2 dB。同时，其与单体方法之间的差距大幅缩小，表明 P-GSVC 在保持可伸缩性的同时，仅付出了较小的质量牺牲。

图7b展示了 UVG 数据集上的分辨率可扩展性结果。在分辨率可扩展性评估中，每个层级都与对应分辨率下的真实值进行比较，且层级越高，分辨率越高。因此，绝对峰值信噪比、多尺度结构相似性或



（a）UVG 上的质量可伸缩性



（b）UVG 上的分辨率可扩展性

图7：UVG 数据集上的可扩展性评估，对比单体式、顺序式和 P-GSVC 方法在（a）质量和（b）分辨率方面的表现。

视频多方法评估融合值无法直接跨层级比较；因此，我们在每个层级内将结果归一化到 $[0,1]$。对于层级 l 上的指标值 m_l，我们采用最小-最大归一化：

$$\tilde{m}_l = \frac{m_l - m_l^{\min}}{m_l^{\max} - m_l^{\min}}, \tag{12}$$

其中，$m_l^{\min}$ 和 $m_l^{\max}$ 分别表示在第 l 层内所有评估方法和高斯配置所得指标值的最小值和最大值。在此图中，x 轴表示泼溅数量，y 轴表示归一化质量分数。与顺序方法相比，我们的方法在增强层中带来了显著更大的增益：在 L_1 时，归一化 VMAF 增益超过 0.25；在 L_2时，增益接近 0.5，同时缩小了与单体式方法之间的差距。这些结果表明，我们的 P-GSVC 比顺序方法实现了更有效的分辨率可扩展性。

率失真权衡。 图8展示了 UVG 数据集上的率失真结果，对比了量化后我们的 P-GSVC 与单体式方法和顺序式方法，并纳入了三层设置下的 SHVC [6] 。在这项对比中，SHVC 使用经过充分优化的 SHM-12.4 参考软件运行，因此具有更强的率失真性能，是标准化可伸缩视频编码的有力基线。单体方法分别训练每个层级，不进行参数共享。由于它不采用逐层训练，因此不存在跨层优化目标冲突，也不会在切换层时出现稳定性问题。在质量可伸缩性设置下，它本质上等同于在不同高斯预算下评估 GSVC。该策略不具备可伸缩性，因为每个层级都是独立训练的，而非由单个比特流派生；在我们的实验中，它作为上界，用于量化实现 P-GSVC 可伸缩性所需付出的质量开销。相反，顺序方法通过跨层级参数共享实现可伸缩性。然而，由于添加新层时优化目标会发生冲突，其逐层训练导致了显著的开销和性能下降。

我们的 P-GSVC 采用分层结构，在各层级之间共享高斯参数，以实现可伸缩视频。因此，率失真曲线呈现分段形式，但相邻层级之间的差距仍然非常小（约为 0.2 dB），证明了联合训练的有效性。不可避免的是，实现可伸缩性会带来一定的质量开销。在默认策略下，即在各层之间均匀分配泼溅，在相同的高斯预算下，所有层的峰值信噪比质量差距均保持在约 1.1 dB 以内。随着泼溅总数量增加，较高质量层级之间的质量差异会减小。这一趋势表明，为增强层分配更多泼溅，可以在保持可伸缩性的同时进一步降低开销，因为增强层主要用于重建高频细节。

尽管我们当前的实现尚未达到高度优化的现代标准（例如 SHVC）的性能，但我们缩小了 SHVC 与顺序方法之间的差距，展现出朝着标准化可伸缩编码质量迈进的实质性进展。此外，需要注意的是，P-GSVC 使用 L2损失进行优化；通过调整收敛条件，我们可以在不增加泼溅数量的情况下进一步提升峰值信噪比。

大小与计算成本。对于一段包含 600 帧的 YUV视频（1920×1080，原始格式约 1.9 GB），在每帧高斯预算为 12k– 27k 的情况下，使用 P-GSVC 对全部三层进行编码后，每段视频生成的平均比特流总大小约为 45– 100MB。我们实现的 P-GSVC 每帧编码耗时约 720 秒。同时，P-GSVC 可实现实时渲染，以约

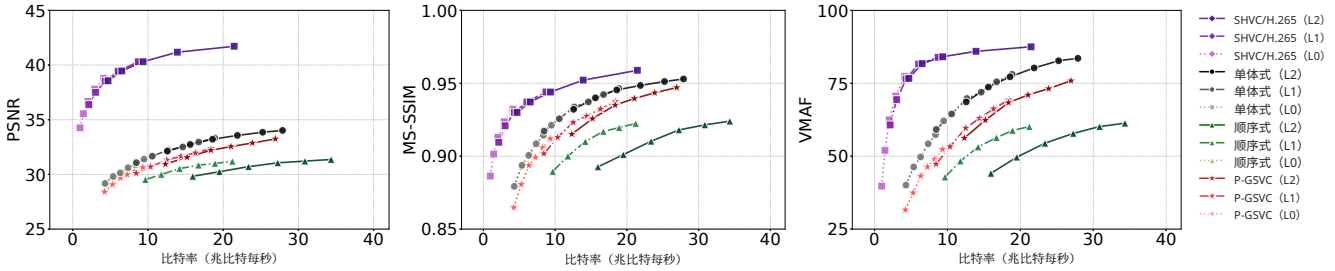


图8：P-GSVC 与基线方法在 UVG 数据集上的量化率失真结果。

1200 帧每秒的速度（2 位有效数字）同时重建全部三层。尽管编码成本高得难以承受，但在视频点播等离线场景中，编码只需执行一次，成本可由多次观看分摊。此外，我们的实现尚未针对速度进行优化。性能分析表明，90% 以上的计算耗时都集中在迭代式高斯优化上，可通过在多个图形处理器上并行计算来加速（ [38] 报告的多图形处理器加速比为 5.33× ）。我们目前正在开发更快的编码器版本，以便实际部署。

5 结论

本文介绍了 P-GSVC，这是首个面向图像和视频、基于渐进式二维高斯泼溅的可伸缩高斯表示框架。我们指出并解决了渐进式高斯表示优化中的一项挑战：增量式逐层训练会因跨层优化冲突而导致质量下降。通过在各层进行联合、增量和循环训练，P-GSVC 表明，我们能够稳定地渐进式重建图像和视频，且相较于逐层顺序式训练，质量有显著提升。这项工作将高斯泼溅视为一种用于编码图像和视频、以支持自适应媒体传输的替代基元，架起了传统可伸缩编解码器与基于神经网络的编解码器之间的桥梁。

参考文献

[1] 2000。信息技术——活动图像及相关音频信息的通用编码——第 2 部分：视频。亦以 ITU-T 建议 H.262 发布。[2] 2004。信息技术——视听对象编码——第 2 部分：视觉。[3] Eirikur Agustsson 和 Radu Timofte。2017。NTIRE 2017 单幅图像超分辨率挑战赛：数据集与研究。载于 *2017 IEEE*计算机视觉与模式识别会议研讨会，*CVPR 2017* 研讨会，*2017*年*7*月*21—26*日，美国檀香山，夏威夷。IEEE计算机学会，1122–1131。doi:10.1109/CVPRW. 2017.150[4] Yash Bhalgat、Jinwon Lee、Markus Nagel、Tijmen Blankevoort 和 Nojun Kwak。2020。LSQ+：通过可学习偏移量和更好的初始化改进低比特量化。*CoRR* abs/2004.09576（2020）。arXiv:2004.09576 https://arxiv.org/ abs/2004.09576[5] Andrew Bond、Jui-Hsien Wang、Long Mai、Erkut Erdem 和 Aykut Erdem。2025。GaussianVideo：基于分层高斯泼溅的高效视频表示。*CoRR* abs/2501.04782（2025）。arXiv:2501.04782 doi:10.48550/ARXIV. 2501.04782[6] Jill M. Boyce、Yan Ye、Jianle Chen 和 Adarsh K. Ramasubramonian。2016。SHVC 概述：高效视频编码标准的可伸缩扩展。*IEEE*电路与系统视频技术汇刊26, 1（2016），20–34。doi:10.1109/TCSVT.2015.2461951[7] Qian Cao、Dongdong Zhang 和 Chengyu Sun。2024。基于神经表示的质量可伸缩视频编码。载于多媒体建模——第*30*届国际会议，*MMM 2024*，荷兰阿姆斯特丹，*2024*年*1*月*29*日至*2*月*2*日，论文集，第一部分（《计算机科学讲义》，第*14554*卷），Stevan Rudinac、Alan Hanjalic、Cynthia C. S. Liem、Marcel Worring、Björn Þór Jónsson、Bei Liu与Yoko Yamakata（编）。Springer，396–409。doi:10.1007/978-3-031-53305-1_30[8] Yihang Chen、Mengyao Li、Qianyi Wu、Weiyao Lin、Mehrtash Harandi与Jianfei Cai。2025年。PCGS：三维高斯泼溅的渐进式压缩。CoRR abs/ 2503 08511 2025 2503 08511 10 48550 2503 08511 。（）。[9] arXiv: 。doi: 。/ARXIV. 。伊士曼柯达公司。。柯达无损真彩色图像套件。http: // r0k.us/graphics/kodak/。24幅图像，真彩色，分辨率 768 × 512。[10] Zhiwen Fan、Kevin Wang、Kairun Wen、Zehao Zhu、Dejia Xu与Zhangyang Wang。2024年。LightGaussian：无界三维高斯压缩，实现15倍缩减与 200+ 帧每秒。载于神经信息处理系统进展38：2024年神经信息处理系统年会，NeurIPS 2024，加拿大不列颠哥伦比亚省温哥华，2024年12月10日至15日，Amir Globersons、Lester Mackey、Danielle Belgrave、Angela Fan、Ulrich Paquet、Jakub M. Tomczak与Cheng Zhang（编）。http://papers.nips.cc/paper_files/paper/2024/hash/ fd881d3b625437354d4421818f81058f-Abstract-Conference.html[11] Robert Gray。1984年。矢量量化。IEEE Assp Magazine 1, 2（1984年），4–29。[12] Caroline N. Haddad。2009年。Cholesky 分解。载于《最优化百科全书》第二版，Christodoulos A. Floudas与Panos M. Pardalos（编）。Springer，374 377 10 1007 978 0 387 74759 0_67 – 。doi: 。/ - - - - Bernhard Kerbl、[13] Georgios Kopanas、Thomas Leimkühler与George Drettakis。2023年。用于实时辐射场渲染的三维高斯泼溅。ACM 图形学汇刊 42, 4 2023 139 1 139 14 10 1145 3592433 （），：–：。doi: 。/ Inseo Lee、[14] Youngyoon Choi与Joonseok Lee。。GaussianVideo：通过高斯泼溅实现高效的视频表示与压缩。载于IEEE/CVF计算机视觉与模式识别研讨会，CVPR 研讨会2025，美国田纳西州纳什维尔，2025年6月11日至15日。计算机视觉基金会 / IEEE，4471–4480。https://openaccess.thecvf.com/content/CVPR2025W/PBVS/html/ Lee_GaussianVideo_ Efficient_Video_Representation_and_Compression_by_ Gaussian_Splatting_CVPRW _2025_paper.html[15] Zhi Li、Christos Bampis、Julie Novak、Anne Aaron、Kyle Swanson、Anush Moorthy与JD Cock。2018年。视频多方法评估融合：旅程仍在继续。Netflix 技术博客 25, 1（2018年）。[16] Ming Lu、Zhihao Duan、Fengqing Zhu与Zhan Ma。2024年。深度分层视频压缩。载于第三十八届AAAI 人工智能会议，AAAI 2024，第三十六届人工智能创新应用会议，IAAI 2024，第十四届人工智能教育进展研讨会，EAAI 2014，2024年2月20日至27日，加拿大温哥华，Michael J. Wooldridge、Jennifer G. Dy与Sriraam Natarajan（编）。AAAI 出版社，8859–8867。doi:10.1609/AAAI.V38I8.28733 [17] Alexandre Mercat、Marko Viitanen与Jarno Vanne。2020年。UVG 数据集：用于视频编解码器分析与开发的50/120帧每秒4K序列。载于第11届ACM 多媒体系统会议论文集，MMSys 2020，土耳其伊斯坦布尔，2020年6月8日至11日，Laura Toni、Ali C. Begen、Özgü Alay与Christian Timmerer（编）。ACM，297 302 10 1145 3339825 3394937 – 。doi: 。/ 。Ben Mildenhall、[18] Pratul P. Srinivasan、Matthew Tancik、Jonathan T. Barron、Ravi Ramamoorthi与Ren Ng。2022年。NeRF：将场景表示为用于视图合成的神经辐射场。ACM通讯 65, 1（2022年），99–106。doi:10.1145/3503250[19] Alberto Presta、Enzo Tartaglione、Attilio Fiandrotti、Marco Grangetto与Pamela C. Cosman。2025年。基于方差感知掩码的高效渐进式图像压缩。载于IEEE/CVF计算机视觉应用冬季会议，WACV 2025，美国亚利桑那州图森，2025年2月26日至3月6日。IEEE，7692–7700。doi:10.1109/WACV61041.2025.00747[20] Heiko Schwarz、Detlev Marpe与Thomas Wiegand。2007年。H.264/AVC 标准的可扩展视频编码扩展概述。IEEE电路与系统视频技术汇刊 17, 9 2007 1103 1120 10 1109 2007 905532 （），–。doi: 。/TCSVT. 。Yuang Shi。。网络化扩展现实中基于D高斯的沉浸式媒体流传输。载于第16届ACM 多媒体系统会议论文集，MMSys 2025，南非斯泰伦博斯，2025年3月31日至4月4日，Herman Arnold Engelbrecht、Silvia Rossi与Simon N. B. Gunkel（编）。ACM，356–360。doi:10.1145/3712676.3719673[21] 2025 3

[22] Yuang Shi、Géraldine Morin、Simone Gasparini与Wei Tsang Ooi。2025年。LapisGS：面向自适应流媒体的分层渐进式三维高斯泼溅。载于国际三维视觉会议，*3DV 2025*，新加坡，*2025*年*3*月*25*日至*28*日。IEEE，991–1000。doi:10.1109/3DV66043.2025.00096[23] Yuang Shi、Géraldine Morin、Simone Gasparini与Wei Tsang Ooi。2026年。Sketch&Patch++：高效的结构感知三维高斯表示。*CoRR* abs/ 2601 05394 2026 2601 05394 10 48550 2601 05394 。（）。[24] arXiv: 。doi: 。/ARXIV. 。Yuang Shi、Ruoyu Zhao、Simone Gasparini、Géraldine Morin与Wei Tsang Ooi。2024年。通过基于神经网络的表示实现体积视频压缩。载于第*16*届沉浸式混合与虚拟环境系统国际研讨会论文集，*MMVE 2024*，意大利巴里，*2024*年*4*月*15*日至*18*日。ACM，85–91。doi:10.1145/3652212.3652220[25] Weronika Smolak-Dyzewska、Dawid Malarz、Kornel Howil、Jan Kaczmarczyk、Marcin Mazur与Przemyslaw Spurek。2024年。VeGaS：视频高斯泼溅。*CoRR* abs/ 2411 11024 2024 2411 11024 10 48550 2411 11024 。（）。[26] arXiv: 。doi: 。/ARXIV. 。Yang-Tian Sun、Yihua Huang、Lin Ma、Xiaoyang Lyu、Yan-Pei Cao与Xiaojuan Qi。2024年。泼溅视频：用于多样化处理的视频高斯表示。载于神经信息处理系统进展*38*：*2024*年神经信息处理系统年会，*NeurIPS 2024*，加拿大不列颠哥伦比亚省温哥华，*2024*年*12*月*10*日至*15*日，Amir Globersons、Lester Mackey、Danielle Belgrave、Angela Fan、Ulrich Paquet、Jakub M. Tomczak与Cheng Zhang（编）。http://papers.nips.cc/paper_files/paper/2024/hash/5a461bdff86cc07e976bb6c518810398-Abstract-Conference.html[27] Yuan-Chun Sun、Yuang Shi、Cheng-Tse Lee、Mufeng Zhu、Wei Tsang Ooi、Yao Liu、Chun-Ying Huang与Cheng-Hsin Hsu。2025年。LTS：面向动态多层三维高斯泼溅场景的DASH流媒体系统。载于第*16*届*ACM* 多媒体系统会议论文集，*MMSys 2025*，南非斯泰伦博斯，*3*月 *31* *4, 2025* 10 1145 3712676 3714445 -*4*月 。ACM。doi: 。/ 。[28] David S. Taubman与Michael W. Marcellin。。*JPEG* - 图像压缩基础、标准与实践。Kluwer国际工程与计算机科学丛书，第 642 10 1007 978 1 4615 0799 4 卷。Kluwer。doi: 。/ - - - - Longan Wang、Yuang Shi与Wei Tsang Ooi。。GSVC：通过二维高斯泼溅实现高效的视频表示与压缩。载于第*35*届数字音频与视频网络及操作系统支持研讨会论文集，*NOSSDAV 2025*，南非斯泰伦博斯，*2025*年*3*月*31*日至*4*月*4*日。ACM， 15 21 10 1145 3712678 3721876 – 。doi: 。/ 。Chang Wu、Guancheng Quan、Gang He、Xin-Quan Lai、Yunsong Li、Wenxin Yu、Xianmeng Lin与Cheng Yang。2024年。QS-NeRV：基于视频神经表示的实时质量可扩展解码。载于第*32*届*ACM*国际多媒体会议论文集，*MM 2024*，澳大利亚维多利亚州墨尔本，*2024*年*10*月*28*日至*11*月*1*日，Jianfei Cai、Mohan S. Kankanhalli、Balakrishnan Prabhakaran、Susanne Boll、Ramanathan Subramanian、Liang Zheng、Vivek K. Singh、Pablo César、Lexing Xie与Dong Xu（编）。ACM，2584–2592。doi:10. 1145/3664647.3680586[31] Xingyu Xie、Pan Zhou、Huan Li、Zhouchen Lin 和 Shuicheng Yan。2024年。Adan：用于更快优化深度模型的自适应 Nesterov 动量算法。*IEEE* 模式分析与机器智能汇刊。46，12（2024年），9508–9520。doi:10.1109/TPAMI.2024.3423382[32] Vickie Ye、Ruilong Li、Justin Kerr、Matias Turkulainen、Brent Yi、Zhuoyang Pan、Otto Seiskari、Jianbo Ye、Jeffrey Hu、Matthew Tancik 和 Angjoo Kanazawa。2024年。gsplat：一个用于高斯泼溅的开源库。*CoRR* abs/2409.06765（2024年）。arXiv:2409.06765 doi:10.48550/ARXIV.2409.06765[33] Neil Zeghidour、Alejandro Luebs、Ahmed Omran、Jan Skoglund 和 Marco Tagliasacchi。2022年。SoundStream：端到端神经音频编解码器。*IEEE/ACM* 音频、语音与语言处理汇刊。30（2022年），495–507。doi:10. 1109/TASLP.2021.3129994[34] Gai Zhang、Xinfeng Zhang 和 Lv Tang。2024年。面向人类与机器的统一可扩展深度图像压缩框架。*ACM* 多媒体计算、通信与应用汇刊。 20, 10 2024 314 1 314 22 10 1145 3678472 ( )，：– :。doi: . / Jiahui Zhang [35]、Fangneng Zhan 、 Muyu Xu 、 Shijian Lu 和 Eric P. Xing。2024年。FreGS：采用渐进式频率正则化的三维高斯泼溅。载于*IEEE/CVF* 计算机视觉与模式识别会议，*CVPR 2024*，西雅图，华盛顿州，美国，*6*月 *16 22, 2024* 21424 21433 10 1109 52733 2024 02024 -。IEEE，–。[36] doi: . /CVPR . . Richard Zhang、Phillip Isola、Alexei A. Efros、Eli Shechtman 和 Oliver Wang。2018年。深层特征作为感知度量的惊人效能。载于*IEEE* 计算机视觉与模式识别会议论文集（*CVPR*）。586–595。[37] Xinjie Zhang、Xingtong Ge、Tongda Xu、Dailan He、Yan Wang、Hongwei Qin、Guo Lu、Jing Geng 和 Jun Zhang。2024年。GaussianImage：利用二维高斯泼溅实现每秒1000帧的图像表示与压缩。载于计算机视觉——*ECCV 2024* 第*4*卷，*2024*年，*18 29 15067* - 欧洲会议，意大利米兰，*9*月*-10*月，论文集第*IX*部分（《计算机科学讲义》，第 卷），Ales Leonardis、Elisa Ricci、Stefan Roth、Olga Russakovsky、Torsten Sattler 和 Gül Varol（编）。Springer，327–345。doi:10.1007/978-3-031-72673-6_18[38] Hexu Zhao、Haoyang Weng、Daohan Lu、Ang Li、Jinyang Li、Aurojit Panda 和 Saining Xie。2024年。关于扩展三维高斯泼溅训练规模。载于欧洲计算机视觉会议。Springer，14–36。[39] Lingting Zhu、Guying Lin、Jinnan Chen、Xinjie Zhang、Zhenchao Jin、Zhao Wang 和 Lequan Yu。2025年。大图像即高斯：利用分层二维高斯泼溅实现高质量大图像表示。载于*AAAI-25*，由人工智能促进协会赞助，*2025*年*2*月*25*日至*3*月*4*日，美国宾夕法尼亚州费城，Toby Walsh、Julie Shah 和 Zico Kolter（编）。AAAI 出版社，10977–10985。doi:10.1609/AAAI.V39I10.33193[40] Brent Zoomers、Maarten Wijnants、Ivan Molenaers、Joni Vanherck、Jeroen Put 和 Nick Michiels。2025年。PRoGS：高斯泼溅的渐进式渲染。载于*IEEE/CVF* 冬季计算机视觉应用会议，*WACV 2025*，美国亚利桑那州图森，*2025*年*2*月*26*日至*3*月*6*日。IEEE，3118–3127。doi:10.1109/ WACV61041.2025.00308