DeepRAHT：面向点云属性压缩的预测式 RAHT 学习

Chunyang Fu[1], TaiQin[2], Shiqi Wang[1]*， Zhu Li[3]l中国香港特别行政区香港城市大学[2]中国北京大学[3]美国密苏里大学堪萨斯城分校，堪萨斯城
chunyang.fu@my.cityu.edu.hk, qintai@stu.pku.edu.cn,shiqwang@cityu.edu.hk, zhu.li@ieee.org

arXiv:2601.12255v1 [eess.IV] 18 Jan 2026

摘要

区域自适应层次变换（RAHT）是一种有效的点云属性压缩（PCAC）方法。然而，其在深度学习中的应用尚缺乏研究。本文提出了一种基于稀疏张量的有损点云属性压缩端到端 RAHT 框架，称为 DeepRAHT。RAHT 变换在学习式重建过程中执行，无需手动进行 RAHT 预处理。我们还引入预测式 RAHT 以降低比特率，并设计了基于学习的预测模型来提升性能。此外，我们设计了一种将游程编码应用于熵模型的比特率代理，实现了无缝变速率编码并增强了鲁棒性。DeepRAHT 是一种可逆且失真可控的框架，能够确保其性能下界，并具有显著的应用潜力。实验表明，与基线方法相比，DeepRAHT 是一种性能更高、速度更快且鲁棒性更强的解决方案。

项目页面 — https://github.com/zb12138/DeepRAHT

引言

三维点云是一组由几何（即点的位置）和属性（例如颜色、法向量和反射率）组成的点。作为三维数据的一种常见表示形式，点云已广泛应用于混合现实、自动驾驶车辆和高分辨率制图等各种实际应用中（MPEG 3DG 2016）。然而，海量点的无结构和无序特性给现有压缩技术带来了挑战。点云几何与属性压缩可以联合进行（Alexiou、Tung 和 Ebrahimi 2020；Yu 等人 2025年），但近期研究中的主流方法（Wang 等人 2025a、b；Guo 等人 2024；Fang 等人 2022年）以及基于 MPEG 几何的点云压缩标准 G-PCC（Schwarz 等人 2018年）都是先压缩几何，再基于重建的几何压缩属性。近年来，点云几何压缩取得了长足进展（Fu 等人 2022；Song 等人 2023b；Wang 等人 2025a；Wang 和 Gao 2025年；You 等人 2025年），尤其是在各种基于深度学习的方法中。然而，属性压缩尚未得到充分研究。本文研究点云属性的有损压缩，并假设几何已经完成重建。

点云属性至关重要，应用广泛。例如，颜色提供最基本的视觉信息，反射率用于 LiDAR 目标检测，而法向量则用于表面重建和渲染。此外，近年来流行的高斯泼溅数据（Kerbl 等人，2023年）也与点云有关，因为高斯基元的参数可以视为属性。有损点云属性压缩的一种重要方法是区域自适应分层变换（RAHT）（De Queiroz 和 Chou，2016年），该方法已被采纳为 G-PCC 标准（MPEG 3DG 2023年）的核心方法，并展现出优异的复杂度和性能。然而，关于 RAHT 的深度学习研究寥寥无几。3DAC（Fang 等人，2022年）是首个使用深度熵模型对 RAHT 变换系数进行编码的框架，但它只是先使用人工设计的 RAHT（即与原始实现相同）生成变换系数，然后设计深度熵模型来学习系数的分布。由于人工设计的 RAHT 不可微，3DAC 只能优化比特率。3DAC 也忽略了预测式 RAHT，而这正是当前 G-PCC RAHT 标准（MPEG 3DG 2023年）中的一项关键改进。预测可以显著降低变换系数的不确定性，因此，对残差而非系数进行编码，能够大幅降低比特率。然而，RAHT 中预测机制的学习长期以来基本未受关注。缺少端到端框架限制了 RAHT 在深度学习中的应用，例如与 RAHT 及其他组件（如去块任务）进行联合学习。

本文提出了首个端到端可微 RAHT 框架 DeepRAHT，并集成了可学习预测模型。在我们的方法中，多尺度RAHT变换是在学习重建过程中执行的，无需预先手动进行 RAHT 变换。端到端设计使我们的框架具有很高的应用潜力，能够联合优化 RAHT 和其他组件。DeepRAHT 与 MPEG G-PCC 具有相同的框架，并据此构建

*通讯作者 版权所有 © 2026年，人工智能促进协会（www.aaai.org）。保留所有权利。

性能下界。此外，DeepRAHT 完全可逆，且失真可控。我们的主要贡献如下：

- 我们实现了首个端到端可微 RAHT 框架，其结构与 G-PCC 参考软件（即 tmc13v14，又称 G-PCCv14）完全相同，并为其性能确立了一个下界。
- 我们开发了一种融入祖先尺度上下文的可学习预测模型。该模型验证了 DeepRAHT 的可学习性，与 G-PCCv14 相比，BD 码率提升超过 24%。
- 我们提出了一种码率代理，使用游程编码作为熵编码器，取代大多数研究采用的熵瓶颈（Ball´e 等，2018年）。游程编码在我们的框架中更为稳健，并可实现无缝变速率编码。

相关工作

传统点云属性压缩方法

点云属性压缩（PCAC）的关键在于通过几何结构挖掘属性相关性。主流点云属性压缩方法主要采用预测、投影和变换技术。预测树（Waschb¨usch 等，2004年）是基于预测的方法中最早的相关方法。G-PCC（MPEG 3DG 2023）的预测变换分支也会在按细节层级划分的细化层级中进行预测。基于投影的方法，例如 MPEG V-PCC（Graziosi 等，2020年），会将三维点集转换为二维图像，从而能够应用现有的图像和视频压缩标准（如 JPEG 和 H.265）（Mekuria、Blom 和 Cesar，2016年；Li 等，2020年）。基于变换的方法着重于在频域中压缩属性，因为此时能量更为集中。图傅里叶变换（GFT）及其变体（Cohen、Tian 和 Vetro，2016年；Shao 等，2017年；Xu 等，2020年；Song 等，2023a年；Zhang 等，2024）已被证明有效，但进行特征分解时复杂度过高。Queiroz 等人（De Queiroz 和 Chou，2016年）提出了一种高效的自适应 Haar 小波变换，称为区域自适应层次变换（RAHT），它避免了 GFT 中的特征分解，并被纳入 MPEG G-PCC 标准。我们的框架实现了 RAHT 的并行化，从而降低了 DeepRAHT 的复杂度。

基于学习的点云属性压缩方法

Alexiou等人（Alexiou、Tung和Ebrahimi，2020年）率先使用稠密三维卷积压缩点云属性。随后，Deep-PCAC（Sheng等人，2022年）提出了另一种基于点卷积的方法。FoldingNet（Quach、Valenzise和Dufaux，2020年）将点云投影到二维图像，并应用图像编解码器进行压缩。LVAC（Isik等人，2022年）利用隐式神经表示（INR）进行点云属性压缩。Fang等人（Fang等人，2022年）提出了3DAC，这是首个直接编码RAHT系数的基于学习的方法；然而，其所使用的RAHT不可微，并且缺少预测组件。与此同时，SparsePCAC（Wang和马，2022年）通过利用堆叠的稀疏卷积层取得了进展。尽管有这些创新，它们的有损压缩性能仍落后于早期的G-PCC测试模型（即G-PCCv14）。

近年来，一些方法取得了显著进展，并声称其性能优于G-PCCv14。3CAC（Wang、Ding 和 Ma，2023年）与 CNet（Nguyen 和 Kaup，2023年）是较早用于无损点云属性压缩的尝试。TSC-PCAC（Guo 等人，2024）通过引入基于 Transformer 的模块进行跨通道上下文回归，增强了 SparsePCAC。Li 等人（Li 等人，2024）提出了首个基于图字典学习的网络。Mao 等人提出了用于有损点云属性压缩的 SPAC（Mao 等人，2025年），这是一种利用残差网络和多个切片的基于采样的框架。近期，Unicorn（Wang 等人 2025b）成为点云压缩领域最先进的深度学习框架，采用平均池化获取多个尺度，并通过稀疏卷积对属性残差进行编码。这些方法在大多数数据集上都展现出更优的结果，但在特定情况下仍暴露出一些稳健性问题。

方法

概述

给定一个包含 N 个点、记作 $P=(p,a)$ 的三维体素化点云，$p\in\mathbb{N}^{N\times 3}$ 是点的位置集合，$a\in\mathbb{R}^{N}$ 是相应的属性[1]。假设输入点云为 P_0，我们的目标是在已知 p_0 的情况下对 a_0 进行有损压缩。在 DeepRAHT 中，首先通过步幅为 $2\times 2\times 2$ 的求和池化对 P_0 进行 s 次池化，从而生成 s 个尺度 $\{P_1,...,P_m,...,P_s\}$。将 A_m 定义为 P_m 中该节点所包含点的属性之和，因此根据求和池化的定义，

$$A_{m+1,i}=\sum_{j\in\mathcal{N}_i}A_{m,j},\quad m=0,1,...,s-1, \tag{1}$$

其中，i 是 P_{m+1} 中的父节点，经过池化后在 P_m 中最多有八个子节点 j。$A_0=a_0$ 是输入属性。

DeepRAHT 的整体框架如图 1 所示。首先进行 s 次求和池化，以生成多尺度点云。编码过程从最后一个尺度 s 开始，按自顶向下的方式进行。对每个尺度上的点云应用变换模型，而预测模型则为可选项（例如，为便于说明，在尺度 m 中禁用了预测）。在 T 模型中，使用 Haar 模块将 A_m 分解为交替系数 AC_m 和直接系数 DC_m。熵编码器 EM 对 AC_m 或其残差进行编码和解码，而 DC_m（不包括根节点 DC_s）等同于 A_{m+1}，已在更高尺度进行编码，因此被丢弃。

与此同时，编码过程中也在进行重建。iHaar 模块执行逆 Haar 变换，以使用解码后的 AC_m 和重建的 DC_m 重建 A_m，其中 DC_m 是根据更高尺度 A_{m+1} 重建的。随后，重建的 A_m

[1]这里讨论单个属性通道，这种情况更为一般，其他通道与之类似。

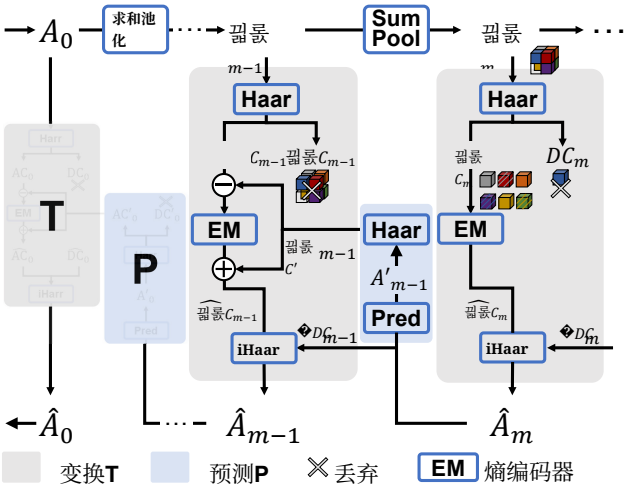


图 1：DeepRAHT 概览。 A_0 是输入属性 ˆ， A_0 是重建属性。

符号说明	描述
A	属性求和
w	节点中的点数
$a = A/w$	平均属性
$g = A/\sqrt{w}$	归一化属性
$g_{\mathrm{L}}, \ldots, g_{\mathrm{HHH}}$	变换系数

表1：符号说明。

ˆ 用于以类似方式重建 DC_{m-1} 。在 P 模型中， A_m 也会预测 AC'_{m-1}，用于对 AC_{m-1} 进行残差编码。T 和 P 过程重复进行，直到编码 A_0。

解码过程也从尺度 s 开始，与重建过程一致，并且 A_0 表示解码后的点云属性。表1为符号说明。

变换模型

本节介绍 Haar 和 iHaar 模块及其基于卷积的实现。区域自适应分层变换（RAHT）是 Haar 小波变换的一种变体，最初由 De Queiroz 和 Chou 于 2016年提出（De Queiroz and Chou 2016）。本文采用 RAHT 的二进制版本（Jonathan Taquet 2020年），该版本改善了原始 RAHT 中高频系数的去相关性。

对于每个 $2\times2\times2$ 体素，Haar 变换将八个节点分解为一个直流系数和七个交替系数。图2展示了一个示例，其中节点的索引为 j，尺度为 m。请注意，其中一个节点是透明的，表示它不包含任何点。将 g_m 记为归一化属性，作为输入的节点特征，

$$g_{m,j} = A_{m,j}/\sqrt{w_{m,j}},\quad j \in \mathcal{N}_i. \tag{2}$$

$w_{m,\,j}$ 是节点 p_0 所包含的点数。$A_{m,\,j}$ 是这些点的属性之和，而 i 是输出直流分量（节点 LLL）在 P_{m+1}中的索引，定义见方程 (1)。

首先应用 Z 轴分解，并以此作为示例进行说明。

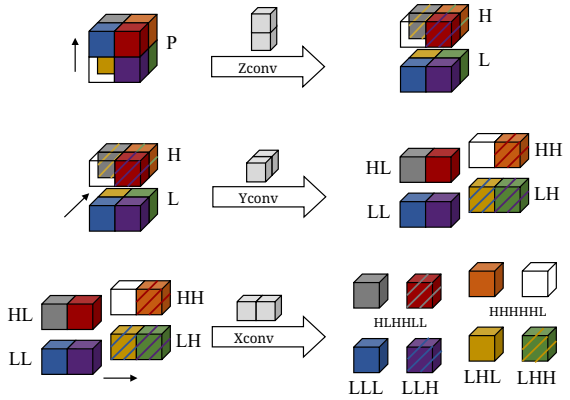


图 2：二进制 RAHT 分解。

下节点对和上节点对（例如紫色和红色）中的归一化属性{v1},{v2}按照以下公式进行变换，

$$\begin{bmatrix} g_{\mathrm{L}} \\ g_{\mathrm{H}} \end{bmatrix} = \frac{1}{\sqrt{w_1+w_2}} \begin{bmatrix} \sqrt{w_1} & \sqrt{w_2} \\ -\sqrt{w_2} & \sqrt{w_1} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, \tag{3}$$

其中，权重 w_1 是下节点（例如紫色节点）所包含的点的数量， w_2 是上节点（例如红色节点）的对应数量。g_{L} 和 g_{H} 分别是低频节点和高频系数节点（即阴影节点）的输出系数，它们将作为后续分解的归一化属性。输出节点的权重被设为 $w_{\mathrm{L}} = w_{\mathrm{H}} = w_1 + w_2$。请注意，如果节点对中的一个节点为空，则非空节点的归一化属性将直接传递至低频部分，以便继续进行后续分解。高频部分的对应系数设为零：$g_{\mathrm{L}} = g_{\text{nonempty}}$、$g_{\mathrm{H}} = 0$、 $w_{\mathrm{H}} = 0$。完成该分解后，输入系数被转换为一半低频系数和一半高频系数。类似地，根据公式（3），依次对 L、H 节点进行两次 Y轴分解，并对 LL、LH、HL、HH 节点进行四次 X轴分解。

三个轴分解后， g_m 被转换为八个系数组 g_{LLL}、g_{LLH}、……、g_{HHH}。请注意，由于存在空节点，这些系数组的数量并不相等。 g_{LLL} 组构成直流系数（直流分量），并且可以证明，直流分量等同于下一尺度的归一化属性（证明见补充材料），

$$DC_{m,\mathcal{N}_i} \equiv g_{\mathrm{LLL}_{m,\mathcal{N}_i}} = A_{m+1,i}/\sqrt{w_{m+1,i}} = g_{m+1,i}, \tag{4}$$

其中， $w_{m+1,\,i} = \sum_{j\in\mathcal{N}_i} w_{m,\,j}$ 和 $A_{m+1,\,i}$ 由方程 (1) 定义。在这方面，就直流分量而言，Haar变换等价于方程 (1) 中描述的步幅为 $2\times2\times2$ 的求和池化，随后进行平方根归一化。由该方程可知， DC_m 等价于 A_{m+1}，因此在编码时丢弃 DC_m，因为 A_{m+1} 已在更高尺度上编码。具体而言，最终尺度上的根直流分量（即 DC_s）会被直接编码。其余七个系数组成交替系数（交流分量），并将使用熵编码器进行压缩。

在重建过程中，根据方程 (4)，可以利用从更高尺度解码得到的属性重建直流分量

$$\widehat{DC}_m = \widehat{A}_{m+1}/\sqrt{w_{m+1}}, \tag{5}$$

交流分量由熵模型从比特流中解码。iHaar 模块以相反的方式运行$\widehat{\ }$，最终得到重建属性 A_m。

G-PCC 实现（MPEG 3DG 2023）通过基于 C++的手动框架实现了上述 RAHT，但该实现不可微。我们使用 Minkowski 稀疏张量（Choy、Gwak 和 Savarese 2019）构建可微的二进制 RAHT。稀疏张量 $\{C, F\}$ 具有坐标字段 $C \in \mathbb{N}_0^{N\times 3}$和关联特征 $F \in \mathbb{R}^{N\times k}$。因此，对于 A_0，稀疏张量 $\boldsymbol{A}_0$[2] 可定义为 $\{p_0, a_0\}$。于是，方程（1）等价于

$$\boldsymbol{A}_{m+1} = \text{SumPooling}(\boldsymbol{A}_m, k = 2^3, s = 2^3), \tag{6}$$

其中，k 表示核大小，s 表示步幅。求和池化也可用于计算节点所包含的点数。定义初始权重稀疏张量 $\boldsymbol{w}_0 = \{p_0, \mathbf{1}\}$，则

$$\boldsymbol{w}_{m+1} = \text{SumPooling}(\boldsymbol{w}_m, k = 2^3, s = 2^3). \tag{7}$$

为对 Haar 分解进行建模，我们使用稀疏卷积对每个轴分解进行建模。以 Z 轴为例，详情如图 3 中的 Z Haar 所示，其中 Zconv 是一种稀疏卷积，其定义如下：

$$\text{Zconv} \equiv \text{Conv}(i = 1, o = 2, k = s = (1, 1, 2)), \tag{8}$$

其中，i 表示输入通道，o 表示输出通道。卷积核的初始权重预先设定为单位矩阵 $\boldsymbol{I}_2$。这样，便得到节点 g_1 和 g_2，并将其映射到两个输出通道。Yconv 和 Xconv 的定义类似，但核大小和步幅分别为 (1, 2,1) 和 (2, 1, 1)。

将 f $\in$ {P, L, H, LL .., HH, LLL ...HHH}（省略默认尺度 m）记作图 2 中的频率，并将 w_{f}记作每个节点的权重（点数）。 $w_{\text{L}} \ldots$、 w_{HH}可通过下式得出：

$$\boldsymbol{w}_1, \boldsymbol{w}_2 = \text{HaarConv}(\boldsymbol{w}_{\text{f}}) \tag{9}$$

$$\boldsymbol{w}_{\text{fL}} = \boldsymbol{w}_1 + \boldsymbol{w}_2 \tag{10}$$

$$\boldsymbol{w}_{\text{fH}} = \begin{cases} \mathbf{0} & \text{if } w_1 = 0 \text{ or } w_2 = 0 \\ \boldsymbol{w}_{\text{fL}} & \text{otherwise,} \end{cases} \tag{11}$$

其中 HaarConv $\in$ {Zconv、Yconv、Xconv} 如图 2 所示。注意，$\boldsymbol{w}_{\text{P}} = \boldsymbol{w}_m$ 作为初始权重。fL 和 fH 分别表示基频对应的低频和高频（例如，若 f = P，则 fH = H；若 f = H，则 fL = HL）。

同样，方程（9）可用于计算变换系数 $\boldsymbol{g}_{\text{f}}$。因此，正向 Haar 变换的公式（3）可通过以下方式实现：

$$\boldsymbol{g}_1, \boldsymbol{g}_2 = \text{HaarConv}(\boldsymbol{g}_{\text{f}}) \tag{12}$$

$$\boldsymbol{g}_{\text{fL}} = \alpha * \boldsymbol{g}_1 + \beta * \boldsymbol{g}_2 \tag{13}$$

$$\boldsymbol{g}_{\text{fH}} = \begin{cases} \mathbf{0} & \text{if } w_1 = 0 \text{ or } w_2 = 0 \\ -\beta * \boldsymbol{g}_1 + \alpha * \boldsymbol{g}_2 & \text{otherwise,} \end{cases} \tag{14}$$

[2]粗体表示稀疏张量，斜体表示其特征 $A = \boldsymbol{A}.F$。

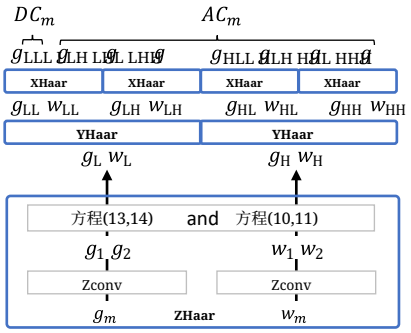


图 3：Haar模块的细节。

其中：

$$\alpha = \sqrt{w_1/(w_1 + w_2)}, \beta = \sqrt{w_2/(w_1 + w_2)}. \tag{15}$$

注意，$\boldsymbol{g}_{\text{P}} = \boldsymbol{g}_m$。Haar模块的细节如图 3 所示。方程 (9)~(15) 重复执行，将 f 从 P 转换为 HH，直到$DC_m = g_{\text{LLL}}$和 $AC_m = \{g_{\text{LLH}}, \ldots, g_{\text{HHH}}\}$获得为止。

iHaar 模块采用转置卷积来建模 HaarConv 的逆运算，即，

$$\text{iZconv} \equiv \text{ConvT}(i = 2, o = 1, k = s = (1, 1, 2)). \tag{16}$$

反向Haar变换的实现如下：

$$\boldsymbol{g}_1 = \alpha * \boldsymbol{g}_{\text{fL}} - \beta * \boldsymbol{g}_{\text{fH}} \tag{17}$$

$$\boldsymbol{g}_2 = \beta * \boldsymbol{g}_{\text{fL}} + \alpha * \boldsymbol{g}_{\text{fH}} \tag{18}$$

$$\boldsymbol{g}_{\text{f}} = \text{iHaarConv}([\boldsymbol{g}_1, \boldsymbol{g}_2]), \tag{19}$$

其中，α、β 定义见方程 (15)，iHaarConv $\in${iXconv、iYconv、iZconv}，f 从 HH 映射到 P。最后，$\widehat{\text{属性}}$求和通过 $A_m = g_{\text{P}} * \sqrt{w_m}$重建。

预测模型

在 DeepRAHT 中，我们从最粗层 P_s 编码到最细层 P_0，因此可以引入预测模型。G-PCC 第 14 版引入了预测式 RAHT（S´ebastien Lasserre 2019），该方法基于逆距离加权预测，利用父节点邻居对当前节点的属性进行上采样。在当前的 G-PCC 第 23 版中，Wang 等人（Wang 等人 2023年）提出利用已解码的兄弟节点邻居来提高预测准确率。然而，如果将这种自回归技术应用于基于深度学习的批处理方法，将显著增加解码时间。因此，我们仅使用父尺度来设计预测模型，如 G-PCC 第 14 版所示。我们首先介绍 IDW 的学习型实现，然后提出预测补偿模块，以提高预测性能。实验结果表明，基于学习的 Pred 模型甚至优于基于兄弟节点的预测。

IDW预测 我们为逆距离加权（IDW）预测设计了一种稀疏卷积。为确保稳定性，IDW在平均属性域中执行。假设 $\boldsymbol{a}'_{m-1}$ = IDW($\widehat{\boldsymbol{a}}_m$)是尺度 $m-1$的预测平均属性，其实现方式如下：

$$\text{IDW}(\widehat{\boldsymbol{a}}_m) \equiv \text{Conv}(\text{Unpool}(\widehat{\boldsymbol{a}}_m), k = 3^3, s = 1^3) \tag{20}$$

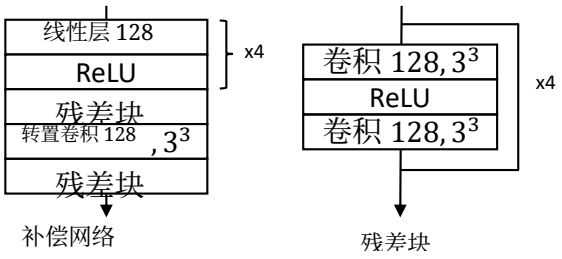


图 4：预测补偿模块。

其中，$\widehat{\boldsymbol{a}}_m \equiv \boldsymbol{A}_m / w_m$ 是平均属性，$\boldsymbol{A}_m$是从父尺度重建的属性，Unpool是一种步幅为2的反池化操作。Conv是一种归一化卷积，其卷积核权重之和为1，且各权重占比根据其与核中心的距离预先设定。具体而言，卷积核中中心、面、边和角的权重占比分别设为4、3、2和1。

预测补偿 如前所述，我们不利用兄弟节点来辅助预测。相反，我们提出了一种补偿模块，通过利用来自祖父节点（即 $m+1$）尺度的上下文信息来增强 Pred 预测。这种方法避免了自回归问题，因此不会大幅增加解码时间。我们使用祖父尺度与父尺度之间的预测误差来补偿当前预测。结合方程 (20)，补偿后的预测为：

$$\boldsymbol{a}'_{m-1} = \mathrm{Comp}(\widehat{\boldsymbol{a}}_m - \mathrm{IDW}(\widehat{\boldsymbol{a}}_{m+1})) + \mathrm{IDW}(\widehat{\boldsymbol{a}}_m), \quad (21)$$

其中，Comp 是补偿模块，如图 4 所示（省略输入和输出的线性层）。预测补偿模块根据经验设计，由多个堆叠的线性层和卷积组成，隐藏层维度为 128，核大小为3^3,，其中包括步幅为 2 的转置卷积。

预测的平均属性 $\boldsymbol{a}'_{m-1}$ 得到预测的属性求和结果 $\boldsymbol{A}'_{m-1} = \boldsymbol{a}'_{m-1} w_{m-1}$，同时也应用 Haar 变换来预测交流系数 AC'_{m-1} =Haar($\boldsymbol{A}'_{m-1}$)，如 P模型所示。最后，交流分量的残差为：

$$r_{m-1} = AC_{m-1} - AC'_{m-1} \quad (22)$$

已编码。

请注意，根据预测效果可以移除预测补偿；预测效果可通过均方误差进行评估，但需要付出 s 比特的代价来向解码器发送信号。这一机制提升了 Deep-RAHT 的鲁棒性，并确保其性能下界为 G-PCCv14。在实验中几乎没有发生移除的情况，这表明预测补偿模块始终是实用的。

熵编码器

大多数基于深度学习的压缩方法都会在其熵模型中使用瓶颈（Ball´e 等，2018年）。然而，它受数据方差的影响很大。为了解决这一问题，我们采用零值游程编码（MPEG 3DG 2020年）（详情请参阅补充材料）作为熵编码器。零值游程编码适用于压缩含有大量零值的数据。RAHT 的系数残差高度集中在零值附近，因此零值游程编码表现更佳、收敛更快，也更稳健。由于游程编码不可微，我们提出一种基于拉普拉斯分布的码率代理，以估计其概率模型，

$$q(r) = \int_{r-0.5}^{r+0.5} \mathcal{L}_{\mu,\sigma}(r)\, dr, \quad (23)$$

其中，均值 μ 和标准方差 σ 由实验确定。游程编码的实际比特率约等于交叉熵 $R \approx \alpha H(p,\ q)$，其中 p 是量化残差 Q(r/qs)的分布，而 α 表示游程编码相较于基于假设的拉普拉斯分布的代理方法所取得的性能提升比例。 Q 是一个 STEROUND (Bengio、L´eonard 和 Courville 2013)，且 qs 是量化步长。请注意，DeepRAHT 中的 Haar 变换完全可逆，因此失真源于对残差的量化，而量化失真仅取决于量化步长。

学习

由于 Haar、iHaar、Pred 以及 EM模块均可端到端微分，因此最终的重建误差可以直接通过 $\ell_{recon} = \|a_0 - \widehat{a}_0\|_2^2$获得。在预测模型中，预测损失 $\ell_{pred} = \sum_m \|(a_m - a'_m)\|_2^2$ 也有助于收敛。另一方面，比特率损失可以由码率代理 $\ell_{bits} = -\sum_m \log_2 q(r_m/qs)$ 得到，其中 $q(r)$定义见公式（23）。因此，总损失为：

$$\ell = \ell_{bits} + \lambda(\ell_{recon} + \ell_{pred}). \quad (24)$$

λ 是用于平衡失真损失和比特率损失的权重。

实验

实验设置

数据集 我们在 MPEG（MPEG 3DG 2024）推荐的 RWTT（De Queiroz 和 Chou 2016年）数据集上训练模型，该数据集包含 568 个真实世界物体。评估在 Owlii（Xu、Lu 和 Wen 2017年）、8iVSLF（Krivoku´ca、Chou 和 Savill 2018年）以及 MPEG CTC样本（MPEG 3DG 2022）数据集的第一帧上进行，这些数据集是大多数 PCACworks 中常用的测试集。

评估指标 采用峰值信噪比（PSNR）评估重建质量，并使用每点比特数（每点比特数）评估压缩率。采用 Bjøntegaard比特率差（BD-BR）评估相同质量下的总体比特率节省。

基线方法 G-PCC（MPEG 3DG 2023）是 MPEG 提供的应用最广泛的标准。比较中采用 Octree-RAHT分支上的最新公开版本（tmc13v23）。3DAC（Fang 等人，2022年）是首个对手动 RAHT 系数进行编码的基于学习的方法。TSC-PCAC（Guo 等人，2024）是一种基于

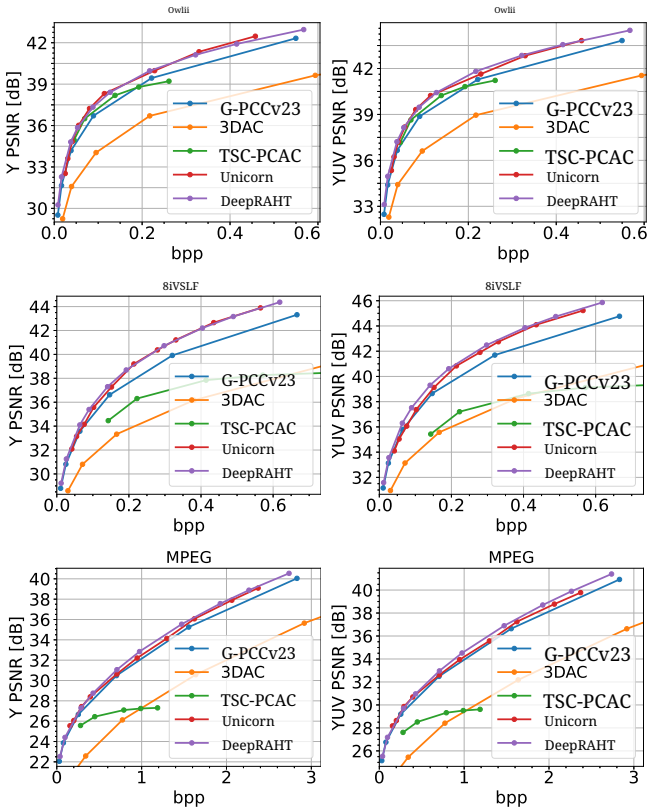


图 5：各数据集上的平均率失真曲线。

Transformer 和通道间上下文回归的自编码器方法。Unicorn（Wang 等人 2025b）是目前最先进的深度学习点云压缩框架。所有基于学习的方法均在 RWTT 上训练。

实现细节 变换和预测由 PyTorch 实现，游程编码则由 C++ 编写。点云颜色在 YUV 空间中进行压缩。所有方法的训练和测试均使用 Core i9 第 13 代 CPU 和一块 NVIDIA 4090 GPU（24 GB 内存）。总尺度 s 设为点云的几何精度，预测从 $s-2$开始，直到 0。我们将 $qs=8$、$\lambda=1/255$的批量大小设为 1，并使用学习率为 0.0001 的 Adam 优化器进行训练。验证损失的详细信息见补充材料。

实验结果

BD-BR 定量增益 DeepRAHT 与基线方法的定量比较见表 2。DeepRAHT 优于所有基线方法，平均而言，相比 G-PCCv23 提升了 16.4%，相比 Unicorn 的 YUV BD-BR 提升了 7.3%。平均率失真曲线如图 5 所示。在较宽的比特率范围内，DeepRAHT 均优于所有基线方法。同时，DeepRAHT 在色度方面也有显著提升：与 Unicorn 相比，U 分量的 BD-BR 增益为 20.5%，V 分量为 20.8%。详细的率失真曲线见补充材料。值得注意的是，

锚点	G-PCCv23 Y	G-PCCv23 YUV	**3DAC** Y	**3DAC** YUV	TSC-PCAC Y	TSC-PCAC YUV	Unicorn Y	Unicorn YUV
篮球	-23.9	-24.8	-69.1	-68.7	-26.4		-31.0	-7.8 -11.0
舞者	-24.4	-25.5	-72.2	-71.4	-19.2		-26.6	-5.8 -10.9
练习	-14.3	-16.8	-64.9	-64.5		-0.4	-2.7	-10.0 -13.5
模型	-13.0	-13.0	-62.6	-61.9		0.2	9.2	8.0 7.1
Owlii 平均值	-18.9	-20.0	-67.2	-66.6	-11.4	-12.8	-3.9	-7.1
Thaidancer	-13.2	-12.3	-62.9	-63.6	-34.8		-47.4	8.0 3.0
拳击手	-23.0	-25.9	-76.4	-75.9	-82.5		-89.8	-20.5 -25.7
长裙	-17.3	-16.5	-67.7	-68.3	-51.8		-61.7	1.1 -6.6
loot	-16.6	-20.0	-73.0	-73.0	-66.0		-73.1	-6.9 -12.0
红与黑	-14.5	-13.3	-76.5	-76.6	-62.6		-72.3	-8.6 -20.3
士兵	-16.1	-17.2	-68.7	-67.7	-46.5		-66.6	-0.8 -3.8
8iVSLF 平均值	-16.8	-17.5	-70.9	-70.9	-57.4		-68.5	-4.6 -10.9
埃及人	-8.6	-8.1	-43.0	-44.0	-83.8		-91.6	NA NA
立面	-16.1	-15.3	-76.9	-78.1	-63.2		-63.6	8.0 7.1
房屋	-4.1	-3.8	-79.1	-81.7	-74.9		-85.4	-6.9 -6.5
Shiva	-2.3	-1.5	-41.6	-44.0	-50.9		-58.6	-10.4 -10.6
ULB	-15.3	-14.7	NA	不适用	-69.6	-76.3	NA	NA
女王	-29.4	-29.6	-64.9	-65.8	-70.7	-79.6	NA	NA
雕像	-9.1	-7.9	-60.8	-62.5	-50.9		-57.1	-4.5 -5.9
MPEG平均值	-12.1	-11.6	-61.1	-62.7	-66.3	-73.2	-3.4	-4.0
平均值	-15.9	-16.4	-66.4	-66.7	-45.0		-51.5	-4.0 -7.3

不适用：该方法无法压缩数据。

表 2：与基线方法相比的 BD-BR 增益（%）。

DeepRAHT 更加稳健，成功完成压缩，并且在所有数据上均优于 G-PCC；而其他基于学习的方法在处理某些大型或稀疏点云时表现不佳。

定性可视化。可视化对比如图 6 所示。可以观察到，与 G-PCC 相比，DeepRAHT 显著减少了伪影。Unicorn 生成的重建结果更加平滑，但会不可逆地丢失细节，例如拳击手的衬衫纹理以及红与黑的面部细节，而 DeepRAHT 能保留更多细节。这一优势源于 DeepRAHT 的可逆性，其失真仅取决于量化。更多对比结果见补充材料。

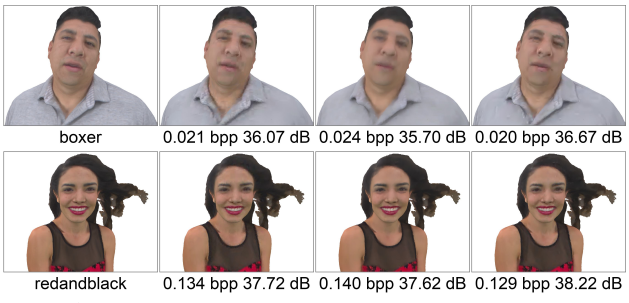


图 6：G-PCC、Unicorn 和 DeepRAHT 的真实值（GT）及重建结果。图中给出了每点比特数和 Y 峰值信噪比。

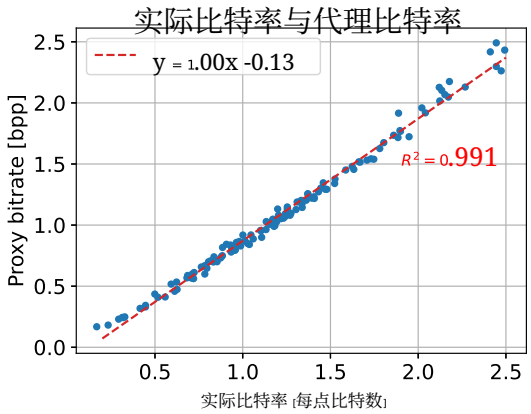


图 7：游程编码与代理比特率。

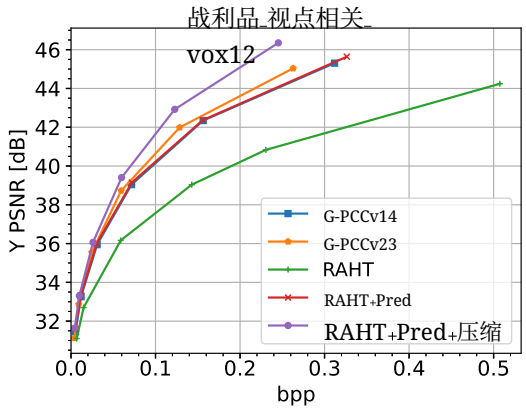


图8：对战利品进行消融研究视点相关。

游程编码的码率代理。我们在熵模型中提出了一种码率代理，用于预测游程编码的码率。通过拟合实际数据，我们获得了公式（23）中的参数 $\alpha = 0.425, \mu = 0, \sigma = 0.2$。测试数据的图 7 展示了实际比特率以及预测值。决定系数为 0.991，表明所提出的码率代理具有很高的准确性。

变速率编码 变速率编码要求使用单个神经网络模型，以不同的速率和质量实现有损压缩。相比之下，3DAC 和 TSC 需要针对不同速率训练多个模型，因为它们所使用的熵模型高度依赖数据的方差。Unicorn 旨在通过引入可调量化层应对这一挑战；然而，单个模型只能适应大约三种比特率。我们提出使用游程编码来解决这一问题，并证明该方法在拉普拉斯分布下对方差各异的数据具有很强的鲁棒性。因此，图 5 中的速率点是通过在一个最高速率检查点上调整 $qs = \{8, 10, 12, 16, 24, 32, 48, 64, 128, 224\}$ 得到的，无需额外训练。

复杂度 在8iVSLF数据集的所有比特率和帧上评估复杂度并取平均值，该数据集中的每个点云平均包含3251505个点。为避免内存溢出，通过KD树将测试点云划分为每个包含 $2 * 10^6$ 个点的块。如表3所示，DeepRAHT在编码和解码方面均快于所有基线方法。DeepRAHT的GPU内存占用仅为8 GB，这表明如果增加块中的点数，其速度还可大幅提升。值得注意的是，我们的模型仅需单个检查-

方法	**3DAC**	TSC-PCAC	Unicorn	DeepRAHT
编码时间	38.45 秒	7.86 秒	20.86 秒	6.03 秒
解码时间	51.71 秒	26.87 秒	14.99 秒	5.74 秒
模型大小	1MB×**5**	148MB×5	65MB×3	88 MB×1
GPU显存	10 GB	22GB	16 GB	**8 GB**

表3：在 8iVSLF 上测试的复杂度比较。

点，模型大小为88 MB，处于适中水平。这种效率得益于我们相对浅层的网络架构和高效的稀疏卷积，为实际应用提供了显著优势。

消融研究 图 8 展示了在 loot viewdep 上进行消融实验的结果。RAHT 表示在不使用任何预测的情况下，采用 De Queiroz 和 Chou（2016年）提出的原始 RAHT，这也是 3DAC 使用的版本。RAHT+Pred 指仅使用逆距离加权预测的 DeepRAHT，与原始 RAHT 相比，其 BD 率节省约为 48.2%。RAHT+Pred 的结构和算法与 G-PCCv14 相同，因此其性能与 G-PCCv14 非常接近；G-PCCv14 作为 DeepRAHT 的下界，确保了其稳健性。采用预测补偿模块的 DeepRAHT（记为 RAHT+Pred+Comp）相比 G-PCCv14 进一步取得了 24.6% 的 BD 率增益。此外，如预测模型一节所述，DeepRAHT 在预测时无需使用兄弟上下文，性能仍优于 G-PCCv23 16.6%。这些结果证明了 DeepRAHT 的预测模型和预测补偿模块的有效性。

结论

本文研究了 RAHT 中预测的学习。我们实现了一种端到端可微的预测式 RAHT，称为 DeepRAHT，使整个框架能够进行联合训练。我们提出了一种基于学习的环内预测模型，利用来自父尺度和祖父尺度的上下文信息，取得了显著的性能提升。我们还设计了一种基于拉普拉斯分布的码率代理，已被证明有助于估计游程编码的比特率，并且在 RAHT系数编码方面更加高效和稳健。与 G-PCCv23 相比，DeepRAHT 实现了 16% 的比特率节省，并且在多个数据集上优于现有的基于学习的压缩方法，展现出在高性能、低复杂度和卓越稳健性方面的显著优势。未来的工作将重点拓展该方法，以用于激光雷达和动态点云的压缩。

致谢

本工作部分获得香港研究资助局（RGC）优配研究金（GRF）拨款 11200323、国家自然科学基金委员会／研究资助局联合研究计划项目 N CityU198/24，以及创新科技署拨款 GHP/044/21SZ 的支持。

参考文献

Alexiou, E.; Tung, K.; and Ebrahimi, T. 2020年. 面向点云压缩的神经网络方法. In《数字图像处理应用 XLIII》，第 11510 卷，18–37. 国际光学工程学会.

Ball´e, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018年. 采用尺度超先验的变分图像压缩. In 国际表示学习会议 .

Bengio, Y.; L´eonard, N.; and Courville, A. 2013. 用于条件计算的随机神经元梯度估计或传播. arXiv 预印本 arXiv:1308.3432. Choy, C.; Gwak, J.; and Savarese, S. 2019. 4D 时空卷积网络：Minkowski 卷积神经网络. In IEEE/CVF 计算机视觉与模式识别会议论文集，3075–3084. Cohen, R. A.; Tian, D.; and Vetro, A. 2016年. 使用图变换压缩稀疏点云属性. In 2016年 IEEE国际图像处理会议（ICIP），1374–1378. De Queiroz, R. L.; and Chou, P. A. 2016年. 使用区域自适应分层变换压缩三维点云. 《IEEE图像处理汇刊》，25(8)：3947–3956. Fang, G.; Hu, Q.; Wang, H.; Xu, Y.; and Guo, Y. 2022年. 3 dac：学习点云属性压缩. In IEEE/CVF 计算机视觉与模式识别会议论文集，14819–14828. Fu, C.; Li, G.; Song, R.; Gao, W.; and Liu, S. 2022年. Octattention：用于点云压缩的基于八叉树的大规模上下文模型. In AAAI 人工智能会议论文集，第 36 卷，625–633. Graziosi, D.; Nakagami, O.; Kuma, S.; Zaghetto, A.; Suzuki, T.; a nd Tabatabai, A. 2020年. 点云压缩标准化工作的概述：基于视频（V-PCC）和基于几何（G-PCC）. 《信号与信息处理学会亚太地区分会汇刊》，9：e13. Guo, Z.; Zhang, Y.; Zhu, L.; Wang, H.; and Jiang, G. 2024. TSC-PCAC：基于体素变换器和稀疏卷积的三维广播点云属性压缩. 《IEEE广播汇刊》. Isik, B.; Chou, P. A.; Hwang, S. J.; Johnston, N.; and Toderici, G. 2022. Lvac：使用基于坐标的网络进行学习式体积属性压缩. 《信号处理前沿》，2：1008812. Jonathan Taquet, S. L. 2020. 关于二进制 RAHT 的报告. 技术报告 m54266, ISO/IEC JTC1/SC29/WG11 MPEG，BlackBerry. Kerbl, B.; Kopanas, G.; Leimk¨uhler, T.; and Drettakis, G. 2023. 用于实时辐射场渲染的三维高斯泼溅. ACM 图形学汇刊，42(4)：139–1.

Krivoku´ca, M.; Chou, P. A.; and Savill, P. 2018. 8i 体素化表面光场（8iVSLF）数据集。输入文档 m42914，ISO/IEC JTC1/SC29/WG11 MPEG，卢布尔雅那。Li, L.; Li, Z.; Liu, S.; and Li, H. 2020. 面向基于视频的点云压缩的高效投影帧填充。IEEE 多媒体汇刊，23：2806–2819。Li, X.; Dai, W.; Li, S.; Li, C.; Zou, J.; and Xiong, H. 2024. 基于 p-拉普拉斯嵌入图字典学习的三维点云属性压缩。IEEE 模式分析与机器智能汇刊，46(2)：975–993。Mao, X.; Yuan, H.; Guo, T.; Jiang, S.; Hamzaoui, R.; and Kwong, S. 2025. SP AC：面向稠密点云的基于采样的渐进式属性压缩。IEEE 图像处理汇刊，34：2939–2953。Mekuria, R.; Blom, K.; and Cesar, P. 2016. 面向远程沉浸式视频的点云编解码器设计、实现与评估。IEEE 视频技术电路与系统汇刊，27(4)：828–842。MPEG 3DG. 2016. 点云压缩（PCC）的用例。技术报告 N16331, ISO/IEC JTC1/SC29/WG11 MPEG。MPEG 3DG. 2020. 关于通过移除字典和游程编码修改属性系数熵编码的报告。技术报告 m54265，ISO/IEC JTC1/SC29/WG11 MPEG。MPEG 3DG. 2022. G-PCC 通用测试条件。技术报告 N0427，ISO/IEC JTC 1/SC 29/WG 7 MPEG。MPEG 3DG. 2023. G-PCC 第二版编解码器说明。技术报告 N00506，ISO/IEC JTC1/SC29/WG7 MPEG。MPEG 3DG. 2024. 基于 AI 的点云压缩通用测试条件。技术报告 N01058, ISO/IEC JTC1/SC29/WG7 MPEG。Nguyen, D. T.; and Kaup, A. 2023. 使用学习得到的条件概率模型进行点云几何与属性无损压缩。IEEE 视频技术电路与系统汇刊，33(8)：4337–4348。Quach, M.; Valenzise, G.; and Dufaux, F. 2020. 基于折叠的点云属性压缩。载于《2020 IEEE国际图像处理会议（ICIP）》，3309–3313。IEEE。Schwarz, S.; Preda, M.; Baroncini, V.; Budagavi, M.; Cesar, P.; Chou, P. A.; Cohen, R. A.; Krivoku´ca, M.; Lasserre, S.; Li, Z.; et al. 2018. 新兴的 MPEG 点云压缩标准。IEEE 新兴与精选电路和系统期刊，9(1)：133–148。Shao, Y.; Zhang, Z.; Li, Z.; Fan, K.; and Li, G. 2017. 使用拉普拉斯稀疏度优化图变换进行三维点云属性压缩。载于《2017 IEEE 视觉通信与图像处理会议（VCIP）》，1–4。Sheng, X.; Li, L.; Liu, D.; Xiong, Z.; Li, Z.; and Wu, F. 2022. Deep-PCAC：一种端到端深度有损点云属性压缩框架。IEEE 多媒体汇刊，24：2617–2632。

Song, F.; Li, G.; Yang, X.; Gao, W.; and Liu, S. 2023a. 具有区域感知优化变换的块自适应点云属性编码。IEEE 视频技术电路与系统汇刊，33(8)：4294–4308。Song, R.; Fu, C.; Liu, S.; and Li, G. 2023b. 面向学习式点云压缩的高效分层熵模型。载于《IEEE/CVF 计算机视觉与模式识别会议论文集》，14368–14377。S´ebastien Lasserre, D. F. 2019. 关于改进 RAHT 以利用属性相关性。技术报告 m47378，ISO/IEC JTC1/SC29/WG11 MPEG，BlackBerry。Wang, J.; Ding, D.; and Ma, Z. 2023. 使用跨尺度、跨组和跨颜色预测进行点云属性无损压缩。载于《2023 年数据压缩会议（DCC）》，228–237。Wang, J.; and Ma, Z. 2022. 基于稀疏张量的点云属性压缩。载于《2022年 IEEE 第五届多媒体信息处理与检索国际会议（MIPR）》，59–64。IEEE。Wang, J.; Xue, R.; Li, J.; Ding, D.; Lin, Y.; and Ma, Z. 2025a. 一种采用通用多尺度条件编码的多功能点云压缩器——第一部分：几何。IEEE 模式分析与机器智能汇刊，47(1)：269–287。Wang, J.; Xue, R.; Li, J.; Ding, D.; Lin, Y.; and Ma, Z. 2025b. 一种采用通用多尺度条件编码的多功能点云压缩器——第二部分：属性。IEEE 模式分析与机器智能汇刊，47(1)：252–268。Wang, K.; and Gao, W. 2025. Unipcgc：通过高效统一方法实现实用的点云几何压缩。载于《AAAI 人工智能会议论文集》，第39卷，12721–12729。Wang, W.; Xu, Y.; Zhang, K.; and Zhang, L. 2023. 用于 G-PCC 的同伴上采样变换域预测。载于《2023 年 IEEE 多媒体与展览国际会议（ICME）》，708–713。Waschb¨usch, M.; Gross, M. H.; Eberhard, F.; Lamboray, E.; and W¨urmlin, S. 2004. 点采样模型的渐进式压缩。载于 PBG，95–102。Xu, Y.; Hu, W.; Wang, S.; Zhang, X.; Wang, S.; Ma, S.; Guo, Z.; and Gao, W. 2020. 用于动态点云属性压缩的预测式广义图傅里叶变换。IEEE 视频技术电路与系统汇刊，31(5)：1968–1982。Xu, Y.; Lu, Y.; and Wen, Z. 2017. Owlii 动态人体网格序列数据集。输入文档 m41658，ISO/IEC JTC1/SC29/WG11 MPEG，澳门。You, K.; Chen, T.; Ding, D.; Asif, M. S.; and Ma, Z. 2025. Reno：用于 3D 激光雷达点云的实时神经压缩。载于《计算机视觉与模式识别会议论文集》，22172–22181。Yu, P.; Zhang, Y.; Liang, F.; Li, H.; and Guo, Y. 2025. 面向点云快速有损压缩的分层失真学习。IEEE 多媒体汇刊，1–16。

Zhang, J.; Chen, Y.; Liu, G.; Gao, W.; 和 Li, G. 2024. 高效的点云属性压缩框架：使用属性引导图傅里叶变换。载于 ICASSP 2024 - 2024 IEEE声学、语音与信号处理国际会议 (ICASSP)，8426–8430。

DeepRAHT 补充材料

Chunyang Fu[1], Tai Qin[2], Shiqi Wang[1], Zhu Li[3]
[1]香港城市大学，中国香港特别行政区[2]北京大学，中国北京[3]密苏里大学堪萨斯城分校，美国堪萨斯城 chunyang.fu@my.cityu.edu.hk, qintai@stu.pku.edu.cn,shiqwang@cityu.edu.hk, zhu.li@ieee.org

直流等价性的证明

我们利用方程（4）通过求和池化高效地生成下一尺度的直流分量。证明如下。根据变换模型中的定义，我们只关注低频变换系数，并假设节点 g_1、g_2 合并为 g_{L_1}，节点 g_3、g_4 合并为 g_{L_2}，依此类推。根据公式（3），

$$\begin{aligned} g_{\mathrm{L}_1} &= \frac{\sqrt{w_1}}{\sqrt{w_1+w_2}} g_1 + \frac{\sqrt{w_2}}{\sqrt{w_1+w_2}} g_2, \\ g_{\mathrm{L}_2} &= \frac{\sqrt{w_3}}{\sqrt{w_3+w_4}} g_3 + \frac{\sqrt{w_4}}{\sqrt{w_3+w_4}} g_4. \end{aligned} \tag{25}$$

根据方程（13），节点 g_{L_1} 和 g_{L_2} 中的权重分别更新为 w_1+w_2 和 w_3+w_4。假设它们合并为 g_{LL_1}，根据方程（3），

$$\begin{aligned} g_{\mathrm{LL}_1} &= \frac{g_{\mathrm{L}_1}\sqrt{w_1+w_2}}{\sqrt{w_1+w_2+w_3+w_4}} + \frac{g_{\mathrm{L}_2}\sqrt{w_3+w_4}}{\sqrt{w_1+w_2+w_3+w_4}} \\ &= \frac{\sum_{j=1}^{4} g_j\sqrt{w_j}}{\sqrt{\sum_{j=1}^{4} w_j}}. \end{aligned} \tag{26}$$

同样，假设节点 g_5、g_6、g_7、g_8 合并为g_{LL_2}，则有，

$$g_{\mathrm{LL}_2} = \frac{\sum_{j=5}^{8} g_j\sqrt{w_j}}{\sqrt{\sum_{j=5}^{8} w_j}}. \tag{27}$$

最终的直流分量由 g_{LL_1} 和 g_{LL_2}合并得到，

$$\begin{aligned} g_{\mathrm{LLL}} &= \frac{\sqrt{\sum_{j=1}^{4} w_j}}{\sum_{j=1}^{8}\sqrt{w_j}} g_{\mathrm{LL}_1} + \frac{\sqrt{\sum_{j=5}^{8} w_j}}{\sum_{j=1}^{8}\sqrt{w_j}} g_{\mathrm{LL}_2} \\ &= \frac{\sum_{j=1}^{8} g_j\sqrt{w_j}}{\sqrt{\sum_{j=1}^{8} w_j}}. \end{aligned} \tag{28}$$

注意，$g_j \equiv A_j/\sqrt{w_j}$的层索引由方程 (28) 得出，

$$g_{\mathrm{LLL}_{m,\mathcal{N}_i}} = \frac{\sum_{j=1}^{8} A_{m,j}}{\sqrt{\sum_{j=1}^{8} w_j}} = \frac{A_{m+1,i}}{\sqrt{w_{m+1,i}}} = g_{m+1,i}, \tag{29}$$

where $w_{m+1,i} = \sum_{j=1}^{8} w_{m,j}$, $A_{m+1,i} = \sum_{j=1}^{8} A_{m,j}$.

版权所有 © 2026年，人工智能促进协会 (www.aaai.org)。保留所有权利。

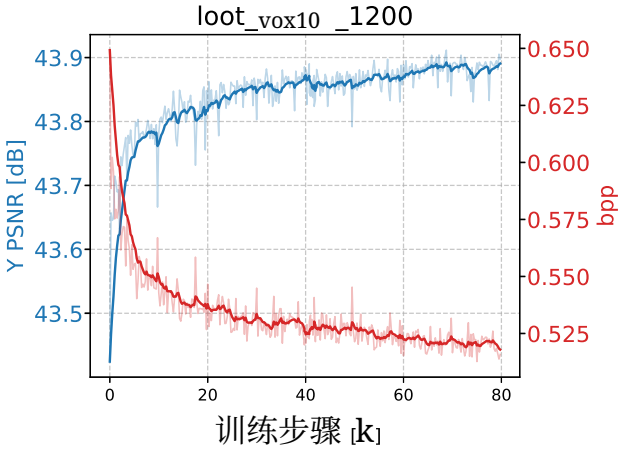


图 9：验证数据的 Y 峰值信噪比和比特率。

B 零游程编码

零值游程编码（MPEG 3DG 2020）是游程编码的一种变体。游程编码是一种无损压缩方法，通过存储（游程，值）条目，高效地编码数据序列，这些条目表示连续零值（游程）和非零值（值）的长度。例如，输入序列为 A000BC0D 时，对应的零值游程编码为 0A3B0C1D。值采用指数哥伦布编码进行二值化，游程则通过一元编码、截断莱斯编码和指数哥伦布编码级联进行二值化。完成二值化后，码字通过算术编码进行压缩。

C 训练过程

训练过程中在 loot vox10 上测试的验证损失如图 9 所示。可以观察到，在训练开始时，PSNR 和比特率都已得到良好初始化。这验证了 DeepRAHT 在未经训练时的性能。同时，随着训练的进行，验证数据的重建质量不断提升，比特率则逐渐降低，这证明了损失函数和码率代理的有效性。相比之下，3DAC 只是 RAHT 系数的一个熵模型，因此在其训练过程中，重建质量不会发生变化。

D R-D曲线详情

R -D 曲线如下所示，展示了每个测试数据 ow.

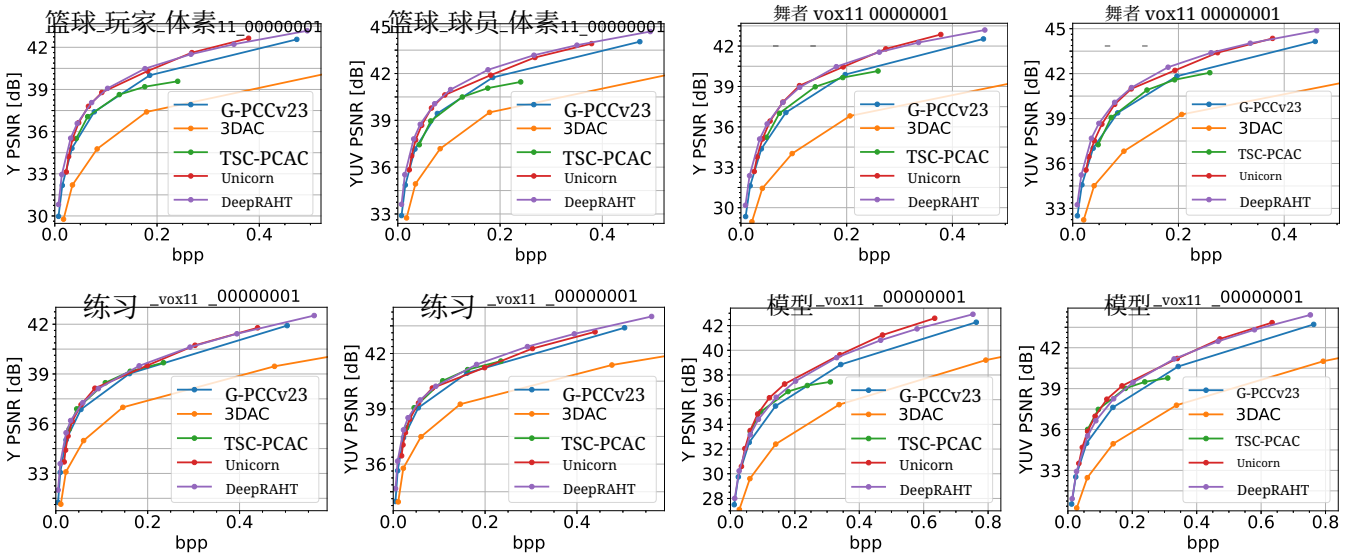


图 10：Owlii数据集的详细率失真比较。

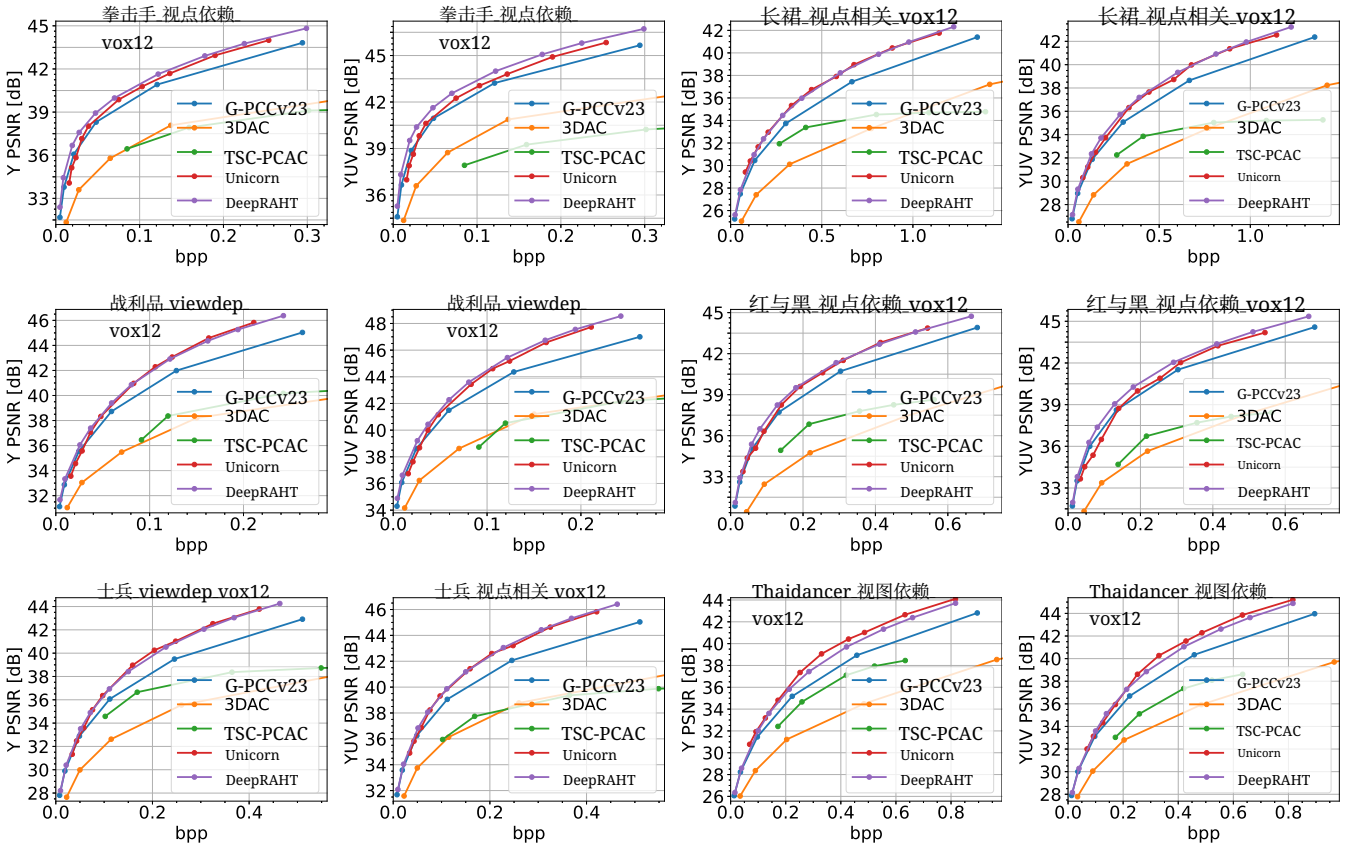


图 11：8iVSLF数据集的详细率失真比较。

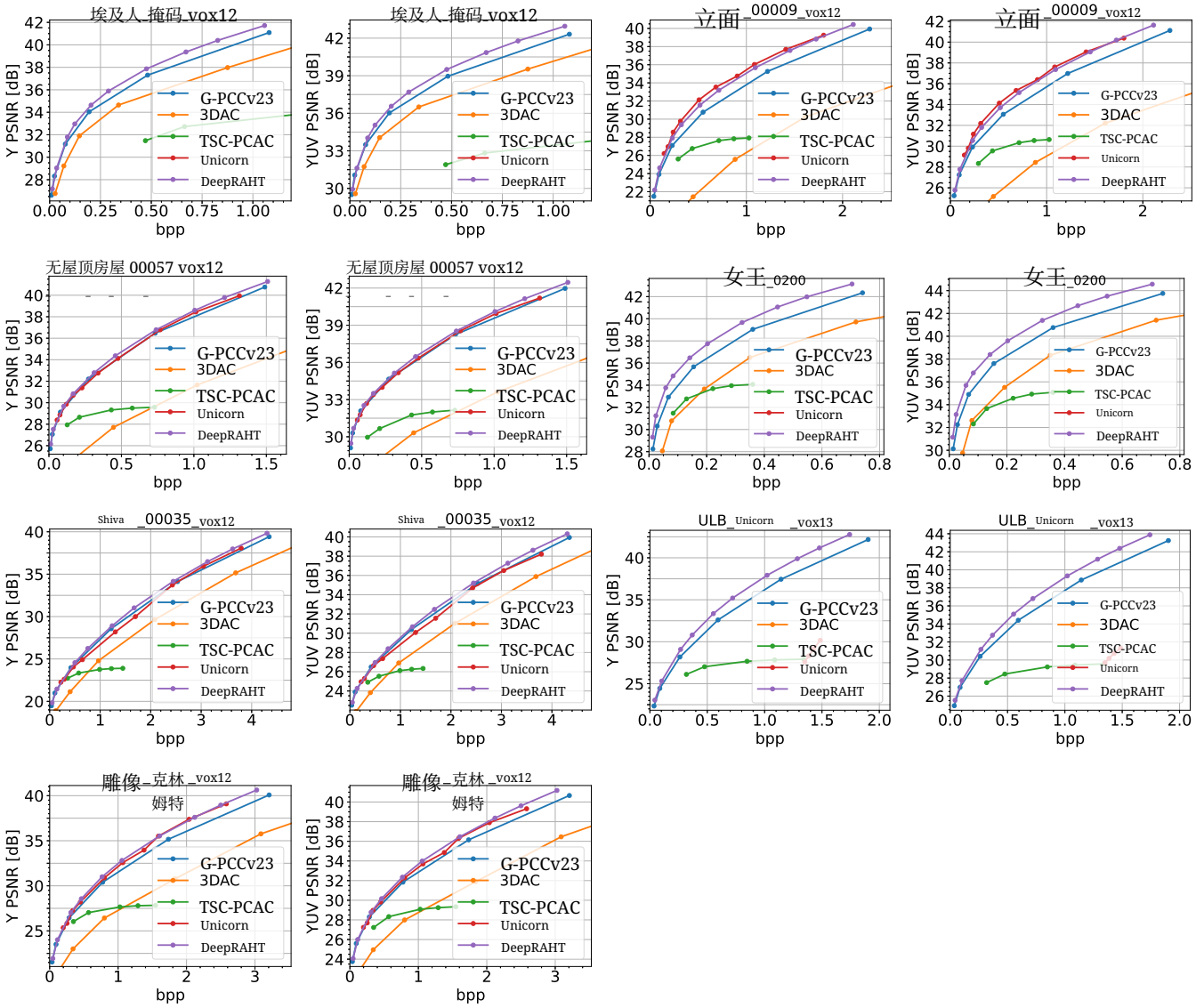


图 12：MPEG 样本的详细率失真比较。

E 色度 R-D曲线详情

下面展示了每个测试数据的色度率失真曲线。

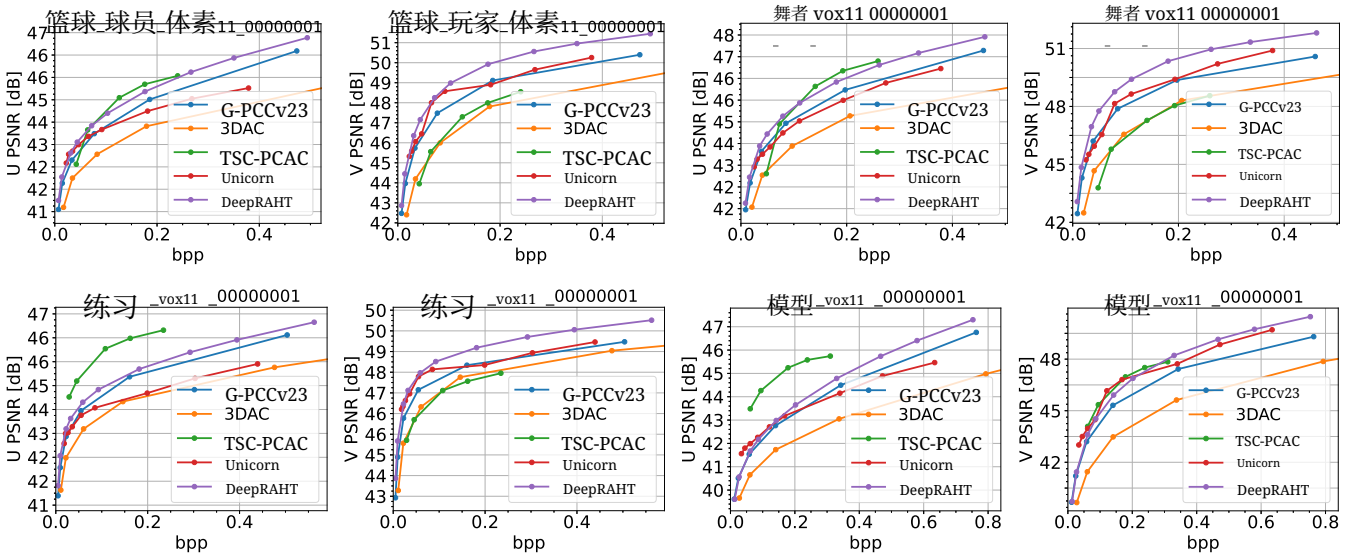


图13：Owlii数据集的详细色度率失真比较。

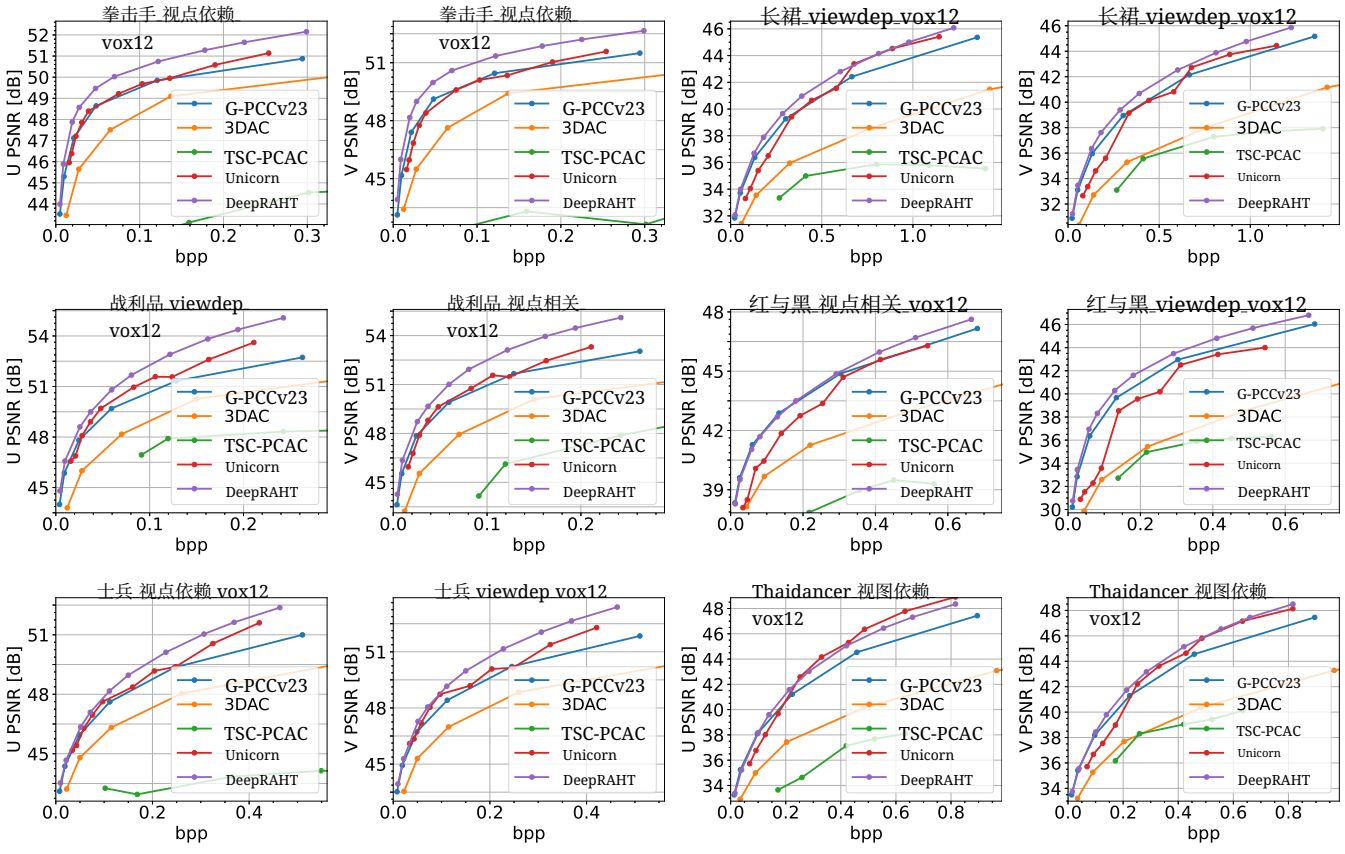


图 14：8iVSLF数据集的详细色度率失真比较。

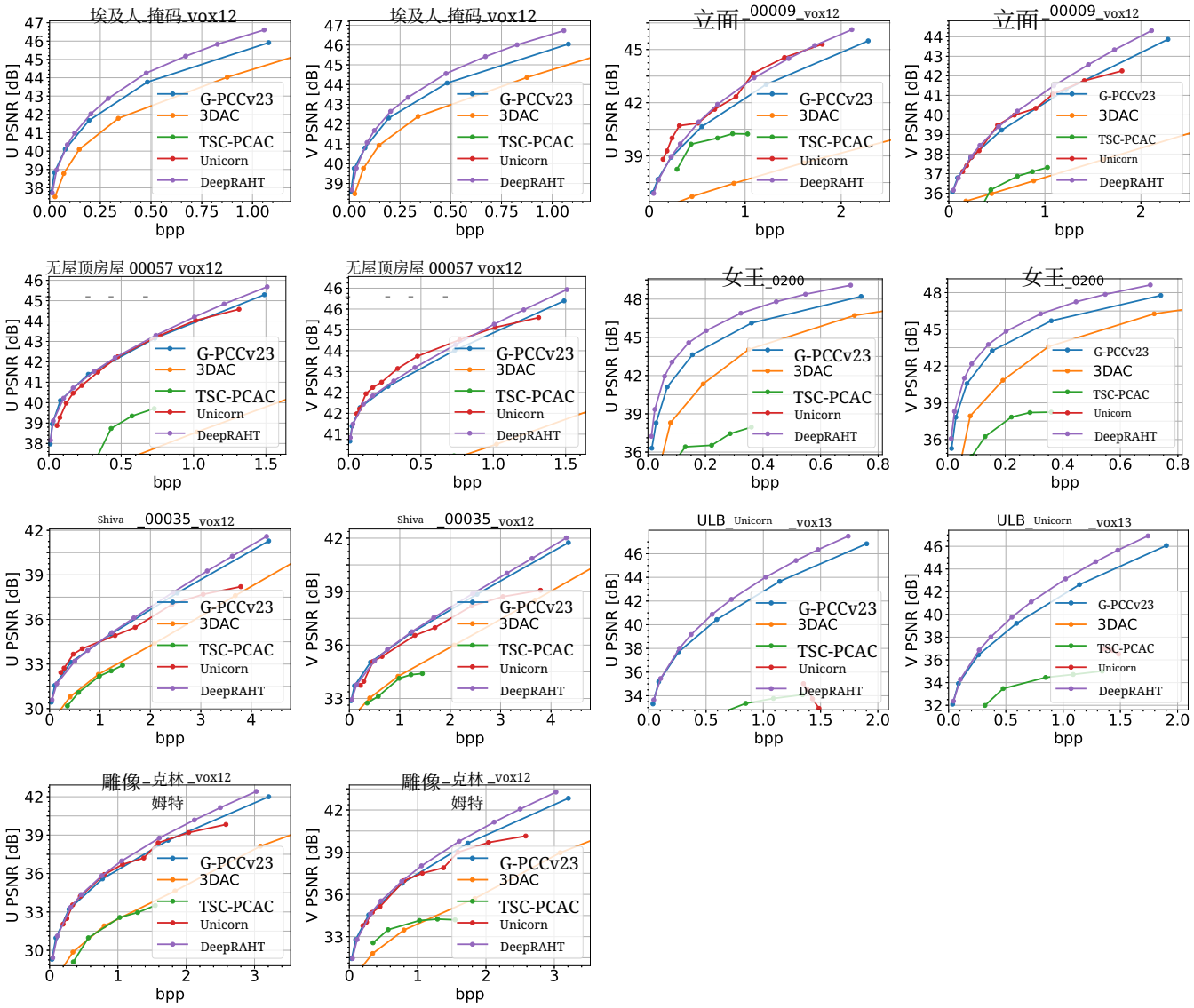


图15：MPEG样本的详细色度率失真比较。

F 更多定性可视化

图16展示了更多定性可视化结果。左侧为较低质量的重建结果，右侧为较高质量的重建结果。采用 PCQM（Meynet 等，2020 年）作为额外的质量指标。



图16：更多定性可视化　　结果。每个重建点云下方显示了 Y、U、V PSNR、每点比特数和 1-PCQM。　　d.

参考文献

Meynet, G.；Nehm´e, Y.；Digne, J.；以及 Lavou´e, G. 2020年。PCQM：一种用于彩色三维点云的全参考质量指标。载于《2020年第十二届多媒体体验质量国际会议》（QoMEX），第1–6页。IEEE。MPEG 3DG。2020年。关于通过字典移除和游程编码修改属性系数熵编码的报告。技术报告 m54265，ISO/IEC JTC1/SC29/WG11 MPEG。