

基于通道增益的可变比特率点云压缩

江照意, 邹文钦, 宋超, 杨柏林*

(浙江工商大学计算机科学与技术学院 杭州 310018)
(ybl@zjgsu.edu.cn)

摘 要: 针对现有基于深度学习的点云压缩方法需要训练多个网络, 耗费大量的时间和空间资源的缺陷, 提出一种基于通道增益的可变比特率点云压缩方法. 首先在网络的编码端利用层次化结构, 通过每个层级提取点云特征和应用偏移注意力机制, 有效地捕捉输入点云的关键特征信息; 然后引入增益单元评估和缩放各个隐向量通道的重要度, 消除向量通道间的信息冗余, 仅需训练单个网络即可实现可变比特率压缩; 为了预测特征向量的概率分布, 对特征向量进行超先验编码, 构造高斯熵模型, 通过熵编码进一步降低编码量; 最后在解码端采用子点卷积进行上采样重构原始点云, 避免顶点的局部聚集, 提高点云的重构质量. 实验结果表明, 在 ShapeNet 数据集上, 以率失真曲线的 BD-rate 作为性能评价指标, 与 VRR 和 Draco 方法相比, 平均比特率分别降低 48.66% 和 63.56%; 压缩性能得到了显著的提升.

关键词: 可变比特率; 点云压缩; 注意力机制; 子点卷积

中图法分类号: TP391

DOI: 10.3724/SP.J.1089.2024.20089

Variable Rate Compression of Point Cloud Based on Channel Gain

Jiang Zhaoyi, Zou Wenqin, Song Chao, and Yang Bailin*

(School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou 310018)

Abstract: Existing point cloud compression methods based on deep learning require training multiple networks, which consumes a large amount of time and space resources. To address this issue, a variable rate point cloud compression network model based on channel gain is proposed. First, the encoding end of the network uses a hierarchical structure to extract point cloud features and apply offset attention mechanisms at each level, effectively capturing the key feature information of the input point cloud. Secondly, gain units are introduced to evaluate and scale the channel of each hidden vector, eliminating information redundancy between vector channels, and requiring only one network to achieve variable rate compression. Then, to predict the probability distribution of feature vectors, a hyperprior encoder is performed on the feature vectors to construct a Gaussian entropy model, further reducing the coding volume through entropy coding. Finally, the decoder uses sub-point convolutions for upsampling to reconstruct the original point cloud, avoiding local point aggregation and improving the reconstruction quality. For the ShapeNet dataset, with BD-rate on the rate-distortion curve as the performance evaluation index, our method reduces the average bitrate by 48.66% and 63.56% compared with VRR and Draco methods, respectively. The experimental results show that the compression performance has been significantly improved compared to current methods.

收稿日期: 2022-11-30; 修回日期: 2023-04-07. 基金项目: 国家自然科学基金(62172366); 浙江省自然科学基金(LY21F020013, LY22F020013). 江照意(1975—), 男, 博士, 副教授, 硕士生导师, CCF 会员, 主要研究方向为计算机图形学、数字几何处理、机器学习、计算机视觉; 邹文钦(1998—), 男, 硕士研究生, 主要研究方向为数字几何处理、深度学习; 宋超(1976—), 男, 博士, 副教授, 硕士生导师, CCF 会员, 主要研究方向为计算机图形学、机器学习、计算机视觉; 杨柏林(1974—), 男, 博士, 教授, 博士生导师, CCF 会员, 论文通信作者, 主要研究方向为计算机图形学、虚拟现实、机器学习.

Key words: variable bitrate; point cloud compression; attention mechanism; sub-point convolutions

近年来, 3 维点云在虚拟现实、3 维视觉和自动驾驶中的应用需求不断增加. 由于点云具有庞大的数据量, 导致对其的存储和传输需要耗费很多的内存和带宽, 因此, 点云的压缩方法一直受到相关研究人员的关注.

随着深度学习在图像和视频压缩中得到成功应用, 基于深度学习的方法也逐渐成为点云压缩的研究热点. 根据输入点云的数据类型, 基于深度学习的点云压缩方法可以分为基于体素的方法和基于点的方法. 基于体素的方法通过体素化将输入点云转化为体素, 再应用规则形式的 3 维卷积抽取点云特征进行压缩. 基于点的方法直接将原始点云作为网络输入, 采用与 PointNet^[1]和 PointNet++^[2]类似的方式编码点云. 然而, 现有的基于深度学习的点云压缩方法, 通过设置不同的损失函数系数进行率失真(rate-distortion, RD)优化, 针对每个系数重新训练网络才能得到不同的压缩比特率. 因此, 该类方法存在显著缺陷, 即需要存储不同系数对应的网络模型参数, 耗费大量的时间和空间资源. 于是, 可变比特率的压缩方法的研究受到人们的关注, 期待通过训练一次网络模型达到可变比特率的目标. 早期的可变比特率压缩方法用于图像压缩, Toderici 等^[3]提出一种基于长短期记忆(long short term memory, LSTM)网络的可变比特率图像压缩方法; 后续的方法通过改进变分自编码框架, 实现可变比特率图像压缩. 而在点云压缩研究中, 基于深度学习的可变比特率压缩方法甚少. Muzaddid 等^[4]首次提出可变比特率点云压缩方法, 直接用 PointNet 作为基本模块, 将原始输入点云编码为固定长度的向量, 利用权重熵损失函数训练网络, 将特征信息集中于向量的低维分量, 通过截取向量通道的方式达到控制不同压缩比特率的目的. 但是, 这种方式会丢失部分信息, 尤其是在压缩比特率较小的情况下; 此外, 其利用 PointNet 提取点云特征, 难以捕获点云的局部结构信息.

为了解决上述问题, 本文提出一种基于通道增益的可变比特率点云压缩网络模型, 仅需要训练单个网络模型, 即可实现不同比特率的点云压缩. (1) 为了减少信息丢失, 在编码端利用多层次结构和偏移注意力机制在底层抽取局部信息, 捕获精细的几何结构, 还能在高层捕获全局的关键信息. 其中, 偏移注意力机制能够很好地捕获无序点云的全局空间上下文信息, 有利于编码端对点云特征信息的聚合. (2) 引入增益单元(gain unit,

GU)用于评估和缩放各个通道的重要度, 消除特征通道间的信息冗余, 实现可变比特率压缩. (3) 在解码端引入逆增益单元(inverse gain unit, IGU), 将量化后的增益特征映射至增益前的数值区间, 避免缩放后压缩性能的退化. 其中, GU 和 IGU 总是成对出现. (4) 在主解码端, 采用子点卷积模块对特征向量进行上采样重构原始点云, 子点卷积模块能够有效地减少重构过程中的随机噪声点的产生, 减小网络复杂度, 提高重构精度.

1 相关工作

随着 3 维扫描技术的快速发展, 3 维点云数据的获取越发迅速、便捷, 目前已可以获取细节丰富的复杂几何物体的点云模型. 3 维点云的应用需求不断增加, 面向点云模型的分析、压缩、注意力机制、上采样等相关工作受到了广泛的关注.

1.1 基于深度学习的点云压缩

点云压缩是点云数据存储和传输中的重要问题, 压缩性能影响点云的应用. 传统的点云压缩标准最早由 MPEG^[5]提出, 包括基于八叉树^[6]的点云几何压缩和基于投影视频的点云压缩. 随着深度学习在图像压缩中的广泛使用, 基于深度学习的点云压缩也引起了越来越多的关注, 根据压缩内容的不同, 分为几何压缩和属性压缩. 此外, 几何压缩包含基于体素的方法和基于点的方法 2 类. 基于体素的方法用于压缩体素化的点云, 代表性的方法有: Quach 等^[7]利用 3 维卷积对体素化点云进行压缩; Wang 等^[8]采用稀疏卷积分析体素化点云, 在精度和速度上都优于使用 3 维卷积的压缩方法. 基于点的方法直接将原始点云作为网络输入, 可以避免体素化过程产生的精度损失. Yan 等^[9]将 PointNet 编码器融入自动编码器框架中, 用于直接压缩原始点云; Wiesmann 等^[10]使用 KPConv^[11]学习点云局部的特征描述子, 充分利用了点的局部信息; Gao 等^[12]提出多尺度的图神经网络采样模块, 在编码端增强邻居点的特征聚合; He 等^[13]将点云的局部密度融合到特征信息中, 取得了比较好的压缩效果. 本文方法属于基于点的压缩方法.

1.2 可变比特率压缩

基于深度学习的可变比特率压缩算法早期应用于图像压缩中. Toderici 等^[3]首次利用 LSTM 在单一网络模型下实现可变比特率的图像压缩, 然

而 LSTM 的计算量巨大, 性能较低. 针对该问题, 研究人员提出了许多改进方法. Cai 等^[14]提出一种多尺度分解的网络模型, 每个尺度对应不同的比特率; Chen 等^[15]将缩放因子嵌入自动编码器框架中, 借助缩放因子控制压缩率的变化; Yang 等^[16]提出一种可调制的自动编码器网络, 将可调制网络与自动编码器相结合, 提出一种控制 RD 操作曲线的有效方法.

与图像压缩相比, 基于深度学习的可变比特率点云压缩方法很少, 现有的大部分点云压缩方法仍然通过训练多个模型来得到对应的不同压缩率. 最近, Muzaddid 等^[4]提出一种可变比特率的点云压缩方法, 将 PointNet 中的特征提取模块作为编码器, 全连接层作为解码器, 引入权重熵损失函数, 通过裁剪向量的通道控制可变比特率; 但是, 这种裁剪方式导致解码端重构精度下降. 本文借鉴 GU 的思想消除特征的通道冗余, 信息损失少.

1.3 点云注意力机制

注意力机制最早在自然语言处理领域广泛使用, 近年来在计算机视觉领域中取得成功, 并受到 3 维点云相关研究人员的关注^[17], 如点云分类、分割、上采样和压缩等任务都应用了注意力机制. Engel 等^[18]提出 Point Transformer 来捕获全局点云语义信息, 用于点云分类和分割任务; Zhang 等^[19]提出稀疏窗口注意力机制和相对注意力, 来聚合粗粒度的局部信息和细粒度的全局形状信息, 用于点云分类任务; Qiu 等^[20]采用偏移通道多尺度注意力用于点云上采样. 当前, 也有一些方法将 Transformer 的思想引入点云压缩, 取得了更好的压缩性能. 本文引入偏移注意力机制提取无序点云的全局空间上下

文信息, 有利于聚合点云的特征信息.

1.4 点云上采样

点云压缩过程中, 解码端通常是一个上采样过程, 上采样的精度影响解码的质量. 就点云压缩而言, 解码时, 基于体素的方法可以采用 3 维反卷积实现, 而基于点的压缩方法由于点云数据的不规则和无序性, 普通的 3 维反卷积并不适用. 近年来出现了许多关于点云上采样的研究工作. Yu 等^[21]利用多分支的多层感知机(multilayer perceptron, MLP)对复制多份的特征进行变换, 达到上采样的目的; Li 等^[22]借助生成对抗网络的思想进行上采样; Qian 等^[23]利用图卷积网络和筛选操作编码点云的局部信息, 完成上采样任务. 在点云压缩中, Liang 等^[24]提出 TransPCC 方法, 利用线性插值的方式得到上采样的点云, 但需要额外存储编码端下采样过程中的中间信息. 本文引入子点卷积进行上采样, 能够有效地维持上采样的精度.

2 本文方法

本文提出一种基于通道增益的可变比特率点云压缩方法, 其框架与通用的自编码器网络相似. 如图 1 所示, 主编码器将原始点云 P 变换为坐标向量 X 和特征向量 F , 包含 3 个模块, 每个模块对上一层的输入点云进行下采样操作和注意力处理; 主解码器也由 3 个对应的模块组成, 用于对解压出来的坐标 $\hat{X}$ 和特征向量 $\hat{F}$ 进行重构点云 $\hat{P}$; 特征向量 F 被超先验编码器变换为超先验向量 Z , 用于构造超先验熵模型.

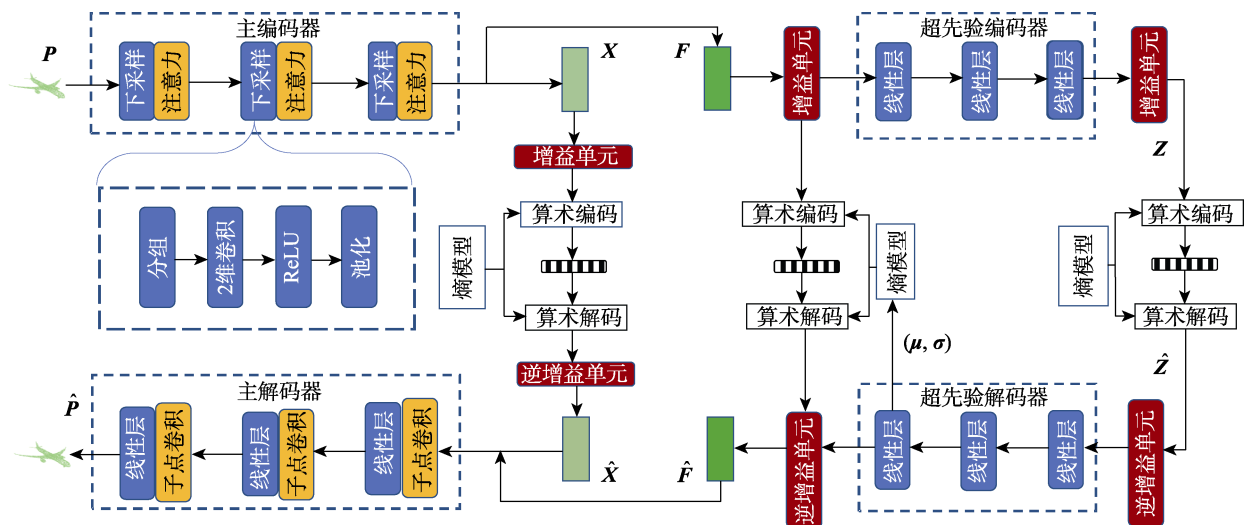


图 1 本文方法框架

本文使用 3 组增益单元对分别作用于网络框架中的 3 种向量, 以控制可压缩率. 此外, 考虑下采样后的坐标信息显著影响重构效果, 将主编码器输出的 $\mathbf{X}$ 和 $\mathbf{F}$ 分开压缩. 其中, $\mathbf{X}$ 经过坐标编码器和坐标解码器, 采用全分解熵模型进行编码与解码; 隐向量 $\mathbf{F}$ 通过超先验分支估计高斯熵模型的参数, 再根据高斯熵模型进行压缩处理.

2.1 可变比特率控制

现有基于深度学习的点云压缩方法通过训练多个模型得到不同的压缩率, 消耗大量的训练时间和存储模型参数所需要的空间. 受文献[25]的启发, 考虑网络中隐向量各个通道的重要性不同, 引入 GU, 通过网络学习优化向量各个通道的缩放系数, 达到控制可变比特率的目的.

GU 的计算公式为 $\bar{\mathbf{f}}_s = G(\mathbf{f}, s) = \mathbf{f} \odot \boldsymbol{\mu}_s$. 其中, $\bar{\mathbf{f}}_s$ 表示增益后的隐向量; $G(\cdot)$ 表示增益操作; $\odot$ 表示通道级别的相乘操作; $\mathbf{f} \in \mathbb{R}^{c \times N}$ 表示增益单元输入的隐向量, c 和 N 分别表示通道的维度和隐向量的个数; $\boldsymbol{\mu}_s = \{\mu_{s,0}, \mu_{s,1}, \dots, \mu_{s,c-1}\}$ 表示增益向量, $\mu_{s,i} \in \mathbb{R}$, s 表示增益向量在增益矩阵中的索引, 所有增益向量组成一个增益矩阵 $\boldsymbol{\mu} \in \mathbb{R}^{c \times n}$, n 表示增益向量的个数. 于是, 隐向量的每个通道都有一个对应的增益值, 与增益向量的通道相对应. 因此, 增益矩阵可以通过网络训练得到每个通道对应的增益值, 即通道的缩放系数.

隐向量经过 GU 处理后称为增益特征, 需要进行量化操作, 用于后续的熵编码, 量化公式为

$$\hat{\mathbf{f}}_s = Q(\bar{\mathbf{f}}_s) = \text{round}(\bar{\mathbf{f}}_s).$$

其中, $\hat{\mathbf{f}}_s$ 表示量化后的增益特征; $Q(\cdot)$ 表示量化处理过程.

受到量化的影响, 增益操作的逆过程无法直接使用增益值的倒数, 为了将 $\hat{\mathbf{f}}_s$ 的数据区间映射到与隐向量 $\mathbf{f}$ 相同的数值区间, 引入 IGU 对 $\hat{\mathbf{f}}_s$ 进行逆向操作. 类似地, IGU 用逆增益矩阵 $\boldsymbol{\mu}' \in \mathbb{R}^{c \times n}$ 表示, 该矩阵中包含多个逆增益向量, 表示为 $\boldsymbol{\mu}'_s = \{\mu'_{s,0}, \mu'_{s,1}, \dots, \mu'_{s,c-1}\}$, $\mu'_{s,i} \in \mathbb{R}$, IGU 的内部计算定义为 $\mathbf{f}'_s = \text{IG}(\hat{\mathbf{f}}_s, s) = \hat{\mathbf{f}}_s \odot \boldsymbol{\mu}'_s$. 其中, $\text{IG}(\cdot)$ 表示逆增益操作.

增益向量与对应的逆增益向量总是成对出现, 训练网络时, 每对增益向量 $\{\boldsymbol{\mu}_s, \boldsymbol{\mu}'_s\}$ 匹配一个拉格朗日系数 λ_s , 该系数是损失函数的超参数, 取自一个给定的拉格朗日系数集合; 同时, 一个拉格朗

日系数对应于 RD 曲线上一个点, 从而得到一系列离散点; 还可以对相邻的 2 个离散失真点进行指数插值, 得到一条连续的 RD 曲线. 通过上述方法, 将所有增益向量对联合训练网络模型一次, 即实现可变比特率的点云压缩.

值得注意的是, 一组增益单元对由多个增益向量对构成. 本文共使用 3 组增益单元对, 分别作用于网络模型中 3 种隐向量: 坐标向量、超先验向量和特征向量. 由于下采样后的坐标信息显著影响上采样阶段的重构质量, 因此将坐标向量嵌入单独的高维隐空间进行压缩. 为了进一步扩大可压缩率的范围, 使用一组增益单元对用于消除坐标向量各通道间的信息冗余. 此外, 通过超先验编码器将特征向量映射成超先验向量, 用于估计高斯熵模型的参数, 并根据高斯熵模型压缩特征向量. 为了控制压缩率, 在压缩超先验向量和特征向量时各自使用 1 组增益单元对进行处理; 联合 3 组增益单元对, 使得可变比特率的控制更加灵活和精确.

2.2 编码模块

2.2.1 主编码器

主编码器输入原始点云, 包含 3 个子模块, 每个子模块对上一层的输入进行下采样操作和注意力处理. 下采样操作中的下采样率是一个超参数, 用于控制最终的压缩比特率和重构精度. 如图 1 所示, 首先通过最远点采样法对输入的点云进行下采样, 得到更加稀疏的点集, 通过查询采样点的邻居点进行分组; 然后通过几个共享的 MLP 和局部池化操作对所有分组进行特征嵌入. 但是, 由于 MLP 本身的局限性, 对空间中的无序点云提取特征时会出现局部特征聚合问题. 为了解决该问题, 在下采样操作后引入偏移注意力模块来聚合空间中点的全局上下文信息. 偏移注意力模块的结构如图 2 所示, 输入特征为 $\mathbf{F}_{\text{in}} \in \mathbb{R}^{N \times C}$, 注意力权重的计算公式为

$$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q} \times \mathbf{K}^T}{\sqrt{C}}\right), \mathbf{A} \in \mathbb{R}^{N \times N}.$$

其中, $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ 分别表示查询矩阵、键值矩阵和值矩阵; C 表示输入的通道数; Softmax 函数用于注意力权重的标准化; $\mathbf{A}$ 表示经过 $\mathbf{Q}$ 和 $\mathbf{K}$ 逐元素相乘和标准化后的加权和, 即注意力权重. 于是, 加权后的注意力特征的计算公式为

$$\mathbf{F}_{\text{sa}} = \mathbf{A}\mathbf{V}, \mathbf{F}_{\text{sa}} \in \mathbb{R}^{N \times C}.$$

考虑 $\mathbf{F}_{\text{sa}}$ 和 $\mathbf{F}_{\text{in}}$ 之间的差异性, 对二者进行逐

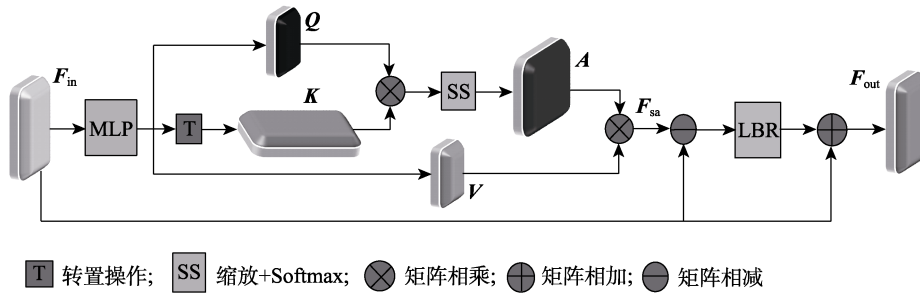


图 2 偏移注意力模块结构

元素相减得到偏移特征，并对偏移量进行卷积和标准化操作(linear+batchnorm+ReLU, LBR)，得到最终的偏移注意力特征

$$F_{\text{output}} = \text{LBR}(F_{\text{sa}} - F_{\text{in}}) + F_{\text{in}}, F_{\text{output}} \in \mathbb{R}^{N \times C}.$$

经过偏移注意力之后，输出特征的维数保持不变。

2.2.2 坐标编码器

坐标编码器将采样后的瓶颈点云坐标嵌入高维坐标隐空间中，由一维卷积模块组成。由于点的坐标信息对于重构质量有很大的影响，坐标信息在压缩过程中需要尽可能地保持精确。为了减小计算量，该模块采用简单的卷积层编码点云的坐标信息，计算公式为 $X' = \text{GR}(\psi(X))$ 。其中， X 表示瓶颈坐标信息； ψ 表示卷积层；GR 表示进行组正则化和 ReLU 激活操作(group regularization and ReLU, GR)。

2.3 解码模块

2.3.1 主解码器模块

主解码器模块的作用是：根据解压得到下采样后的坐标向量 $\hat{X}$ 和特征向量 $\hat{F}$ 进行上采样，重构点云；主解码器由线性层和子点卷积模块组成。与主编码器模块类似，主解码器有 3 个对应的上采样阶段，在每个上采样阶段，将点的坐标向量和特征向量根据上采样率进行扩张，并将扩张后的坐标和特征作为下个阶段的输入。上采样率也是超参数，通常与主编码模块的下采样率相对应。

为了实现点云上采样，现有方法使用线性插值进行特征扩维，或者用简单的反卷积操作扩维。但是，由于插值方法需要用到编码端每个阶段的中间信息，严重影响编码的效率，因此插值方法并不适用于压缩任务。此外，反卷积上采样只能用于规则的数据，对于不规则且无序的空间点云，无法直接使用反卷积。为了解决上述问题，在主解码器中引入子点卷积^[13]进行上采样，子点卷积如图 3 所示。其中，输入是特征 f_{in} ，维数为 $N \times d_{\text{in}}$ ， U 表示上采样率；周期筛选^[21]用来调整特征的维数。

首先将输入特征按通道划分为 U 组，每组的维数均为 d_{in}/U ，再分别对各组进行 2 维卷积操作实现扩维，将通道维数扩展至 d_{out} 。合并所有组的通道，特征的维度为 $N \times U d_{\text{out}}$ ，通过周期筛选操作，特征的维度变换 $UN \times d_{\text{out}}$ ，得到上采样后的输出特征。与之前的插值方法和反卷积方法不同，子点卷积模块只用到瓶颈信息进行上采样，能够有效地提高压缩效率；同时，分组卷积不仅可以降低计算量，也可以有效地避免顶点的局部聚集现象。

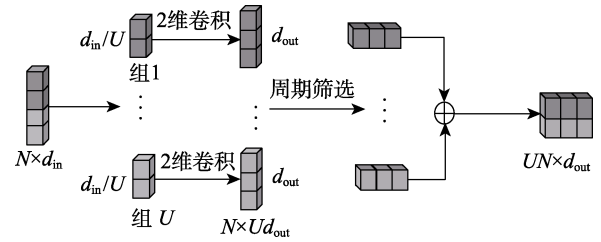


图 3 子点卷积示意图

主解码器将子点卷积运用于各个阶段的上采样过程。如图 4 所示，坐标向量和特征向量分别进行上采样，采样公式为

$$\begin{cases} \mathbf{x}_{i+1} = d(\mathbf{x}_i) + \Delta \mathbf{x}_i \\ \mathbf{f}_{i+1} = d(\mathbf{f}_i) + \Delta \mathbf{f}_i \end{cases}.$$

其中， $\mathbf{x}_i \in \mathbb{R}^{N \times 3}$ 表示输入坐标向量； $\mathbf{f}_i \in \mathbb{R}^{N \times d}$ 表示特征向量； $\Delta \mathbf{x}_i$ 表示点的坐标偏移； $\Delta \mathbf{f}_i$ 表示特征偏移； $d(\cdot)$ 表示复制操作。 $\mathbf{f}_i$ 通过子点卷积操作进行上采样，再通过 MLP 将其映射到 3 维坐标空间，得到坐标偏移量 $\Delta \mathbf{x}_i$ ；同时，将输入坐标直接复制 U 份，作为上采样后的基础坐标，加上坐标偏移得到上采样的点云坐标。类似地，使用子点卷积将 $\mathbf{f}_i$ 转换为上采样的特征偏移 $\Delta \mathbf{f}_i$ ，并将输入特征复制 U 份，加上特征偏移量得到上采样的点云特征。

2.3.2 坐标解码器

坐标解码器与坐标编码器是对称的，用来对

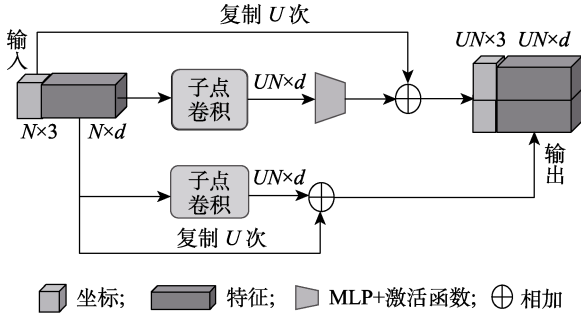


图 4 上采样过程示意图

压缩后的坐标向量进行解码. 与坐标编码器类似, 坐标解码器由 2 个共享的 MLP、组标准化层和 ReLU 激活函数构成.

2.4 熵模型

为了进一步压缩数据量, 量化后的隐向量需要进行熵编码. 当前, 使用最广泛的熵编码方法是算术编码, 关键是设计一个高效的概率模型, 用于估计隐向量的概率分布. 该模块中, 特征向量使用超先验熵模型进行编码, 坐标向量和超先验向量使用全分解熵模型进行编码.

特征向量中, 借助超先验向量捕捉特征向量的空间依赖关系, 更加精确地估计出特征向量的概率分布. 针对点云, 本文设计了一组简单有效的超先验编码器和解码器, 用于辅助超先验熵模型的参数生成. 超先验编码器和解码器都由线性层和 ReLU 激活函数构成. 超先验编码器对特征向量进行特征提取, 得到超先验向量 $\mathbf{Z}$, 再通过全分解熵模型学习出 $\mathbf{Z}$ 的概率分布, 用于编码 $\mathbf{Z}$; 该解码器根据 $\mathbf{Z}$ 输出高斯熵模型的参数, 估计出特征向量 $\mathbf{Y}$ 的概率分布. 由于引入超先验信息, 高斯熵模型比全分解熵模型能够更加精确地匹配特征向量的实际分布, 有利于提升压缩性能.

2.5 损失函数

神经网络模型使用标准的 RD 损失作为损失函数, 用于权衡失真与比特率的重要性, 其定义为 $L = R + \lambda D$. 其中, D 表示失真损失; R 表示比特率损失; λ 表示拉格朗日系数.

比特率损失 R 包含对 2 个全分解模型和 1 个高斯熵模型估计的概率分布进行算术编码的比特位数之和, 计算公式为 $R = R_c + R_f + R_{hp}$. 其中, R_c , R_f , R_{hp} 分别表示坐标隐向量、特征隐向量和超先验隐向量的编码比特率.

对于失真损失 D , 通常仅需考虑输入点云和重构点云之间的重构误差 D_p . 然而, 由于编码器包含多个层级, 为了避免误差累积, 还需计算多层

级损失 D_{mul} ; 另外, 瓶颈坐标向量的重构严重影响最终点云的重构, 还需考虑坐标向量 $\mathbf{X}$ 和解码后的 $\hat{\mathbf{X}}$ 之间的瓶颈损失 D_x . 因此, 总的失真损失的计算公式为 $D = D_p + \sigma D_{mul} + \rho D_x$. 其中, σ 和 ρ 分别表示各自的权重系数; $D_p = L_{CD}(\mathbf{P}, \hat{\mathbf{P}})$, $\mathbf{P}$ 表示输入点云, $\hat{\mathbf{P}}$ 表示重构点云; $D_{mul} = \sum_{k=0}^{K-1} L_{CD}(\mathbf{P}_k, \hat{\mathbf{P}}_k)$, $\mathbf{P}_k$, $\hat{\mathbf{P}}_k$ 分别表示第 k 层级的下采样点云和上采样点云; $D_x = L_{CD}(\mathbf{X}, \hat{\mathbf{X}})$, $\mathbf{X}$ 表示瓶颈坐标, $\hat{\mathbf{X}}$ 表示解码瓶颈坐标. 各类失真损失都使用倒角距离(chamfer distance, CD)来度量^[26].

3 实验结果与分析

3.1 实验设置

选用公开数据集 ShapeNet^[27]训练整个网络, 该数据集共包含 55 个类别的点云集合, 且顶点坐标均标准化到 $[-1, 1]$ 中. 数据集的具体信息参照文献[23]. 实验中, 选取的下采样率和上采样率分别为 $[1/4, 1/2, 1/2]$ 和 $[2, 2, 4]$; 嵌入层的特征维度设置为 8. 优化训练损失函数时, 设置权重 $\sigma = 0.01$, 权重 $\rho = 0.1$. 在可变比特率控制中, 增益向量对的数量 n 与拉格朗日系数的数量相等, 取 $n=6$. 本文中, 将损失函数中对应的拉格朗日系数集合设置为 $\lambda = \{10, 50, 100, 500, 1000, 4000\}$. 在训练的过程中, 每次迭代时, 从 0~5 随机选择一个索引 s , 根据该索引, 分别从 μ , μ' 和 λ 中选择对应的 μ_s , μ'_s 和 λ_s ; 然后, 由选中的 λ_s 计算损失函数, 对选中的 μ_s 和 μ'_s 进行联合训练和优化. 网络使用 Adam 优化器训练, 初始学习率为 0.001, 训练的 Batch 大小为 32, 共训练 100 个 Epoch. 实验平台为显存为 8GB 的显卡, 型号为 NVIDIA GeForce GTX 1080 Ti GPUs.

3.2 性能比较

3.2.1 比特率-失真曲线比较

为了验证本文方法的有效性, 与可变比特率点云压缩方法 VRR^[4]和开源方法 Draco^[28]进行实验. 使用峰值信噪比(peak signal-to-noise ratio, PSNR)和 CD 衡量压缩点云的重构精度, 采用逐点比特率(bit per point, BPP)衡量压缩比特率的大小. 实验结果如图 5 所示, RD 曲线的结果为测试集中所有样本的平均值, 在相同的 BPP 下, PSNR 越高且 CD 越低, 表示重构的精度越好. 可以看出: (1) 本文方法获取的 BPP 取值范围更大, 具有更宽的比特率调整范围; 与 VRR 和 Draco 方法相比, 该方法

的重构精度均有显著提升. (2) 在 BPP 取值较低时, VRR 的性能下降十分明显, 因为该方法通过通道裁剪的方式控制比特率, 低比特率下被裁剪的信息过多, 影响解码端重构精度; 而且, VRR 编码端利用 PointNet 提取特征, 无法聚合局部信息, 重构质量较低. (3) 由于 Draco 是一种非深度学习方法, 基于先验知识手工构建特征无法自适应地消除冗余性,

性能最差. (4) 本文方法采用通道增益的方式, 在调整比特率时能够自适应地保留关键信息, 注意力机制和子点卷积模块的应用进一步提高了压缩性能, 以 RD 曲线的 BD-rate 作为性能评价指标, 与 VRR 和 Draco 方法相比, 该方法的平均比特率分别降低 48.66% 和 63.56%. 实验结果表明, 与现有方法相比, 本文方法的压缩性能得到显著提升.

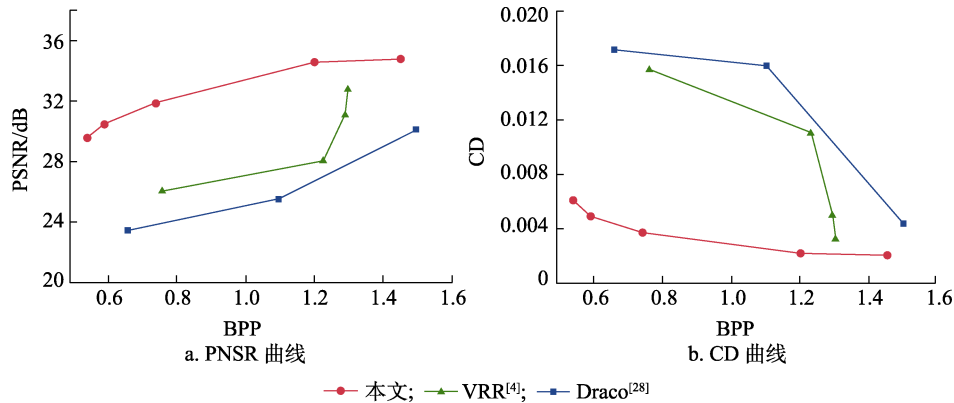
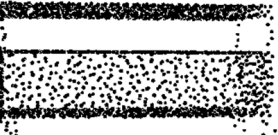
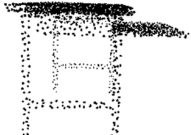
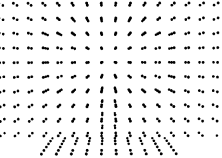
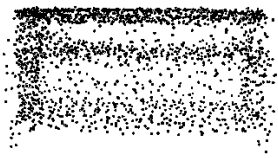
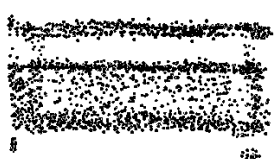


图 5 不同方法压缩性能对比

在测试集中随机抽取几个不同类别的点云模型进行压缩并重构, 可视化结果如表 1 所示, 其中, 测试的点云数据为显示器、长椅和桌子模型. 可以看出, 在 BPP 取值接近甚至更低的情况下, 本文方

法能够保留更多的局部细节, 重构效果在视觉上更佳; VRR 方法在电脑和桌子的测试点云上出现了较多明显的噪声, 重构点云和原始点云的空间分布差异较大; 在较低的比特率下, Draco 方法的量化步

表 1 不同方法点云模型重构的可视化结果

方法	点云模型		
	显示器	长椅	桌子
GT			
Draco ^[28]	 BPP: 1.62 PSNR: 30.60 dB	 BPP: 1.68 PSNR: 30.08 dB	 BPP: 1.82 PSNR: 28.34 dB
VRR ^[4]	 BPP: 1.32 PSNR: 32.08 dB	 BPP: 1.40 PSNR: 30.77 dB	 BPP: 1.45 PSNR: 29.37 dB
本文	 BPP: 1.36 PSNR: 33.89 dB	 BPP: 1.42 PSNR: 34.15 dB	 BPP: 1.40 PSNR: 33.58 dB

长较大, 重构点云较为稀疏, 视觉效果欠佳。

3.2.2 空间复杂度比较

空间复杂度是衡量深度学习网络模型性能的重要指标。为了比较固定比特率和可变比特率压缩方法的空间复杂度, 将本文方法分成移除和保留 GU 的 2 种情况进行定量对比。其中, 移除 GU 的网络是固定比特率方法, 需要训练多个网络得到不同比特率下的 RD 曲线, 模型参数多, 占用更多内存空间; 而保留 GU 的网络只需要训练一个网络即可实现可变比特率的压缩, 大大降低了模型的参数数量和存储空间, 唯一的代价是需要存储少量的 GU 数据。实验结果如表 2 所示。可以看出, 可变比特率方法的模型参数数量和空间复杂度大约是固定比特率版本的 1/5, 可变比特率的方法能够有效地节省内存, 降低计算复杂度。

表 2 空间复杂度比较

比特率	模型参数量/KB	空间复杂度/MB
可变	25.6	54.6
固定	128.0	273.0

3.3 消融实验

通过消融实验, 验证本文方法中各个模块的有效性。以包含 PointNet++编码器^[8]、全分解熵模型和简单的一维反卷积解码器^[23]等基本模块的网络作为基准网络, 用于消融实验的模块包括编码端的偏移注意力模块、超先验熵模型和解码端的子点卷积模块。在基准网络上, 逐个加入上述各个模块, 在相同的训练超参数下, 重新训练加入各个模块后的网络; 然后分别测试重构模型的 PSNR 和 CD; 所有模块都是在 ShapeNet 数据集上进行训练和测试。实验结果如表 3 所示。可以看出, 基准网络的 PSNR 值最低, CD 值最高, 重构效果最差; 加入各个消融模块后, PSNR 值逐步升高, CD 值逐渐降低, 重构精度都有一定程度的提升; 值得注意的是, 加入子点卷积模块后, PSNR 值的提升最显著, 说明上采样阶段的稳定性和准确性对于重构质量的影响更重要, 比反卷积上采样方法效果更好。实验结果表明, 在基准网络中加入所有消融模块后,

表 3 消融实验结果

模块	PSNR/dB	CD
基准	30.63	0.484
基准+偏移注意力	31.40	0.363
基准+偏移注意力+子点卷积	34.62	0.262
基准+偏移注意力+子点卷积+超先验	34.73	0.197

注: 粗体表示最优值。

PSNR 和 CD 都取最优值, 验证了各个消融模块的有效性。

4 结 语

本文提出一种基于深度学习的可变比特率点云压缩方法, 利用通道 GU, 只需要训练单个网络模型, 实现了可变比特率控制。在编码端引入偏移注意力机制, 在特征提取过程中能够有效地聚合点云的全局和局部信息; 在解码端引入子点卷积模块, 能够减轻上采样过程中的顶点聚集现象和随机噪声点的产生。实验结果表明, 在 ShapeNet 数据集上, 与 VRR 和 Draco 方法相比, 本文方法的压缩性能得到显著提升。但是, 由于未考虑点云中的顶点分密度不均匀, 本文方法压缩 Lidar 点云时表现不佳。另外, 未来工作将进一步探索动态点云可变比特率控制方法, 以及更有效的点云上采样方法。

参考文献(References):

- [1] Charles R Q, Su H, Mo K C, *et al.* PointNet: deep learning on point sets for 3D classification and segmentation[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 77-85
- [2] Charles R Q, Li Y, Su H, *et al.* PointNet++: deep hierarchical feature learning on point sets in a metric space[C] //Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2017: 5105-5114
- [3] Toderici G, O'Malley S M, Hwang S J, *et al.* Variable rate image compression with recurrent neural networks[C] //Proceedings of the 4th International Conference on Learning Representations. San Juan, Puerto Rico 2016
- [4] Muzaddid M A A, Beksi W J. Variable rate compression for raw 3D point clouds[C] //Proceedings of the International Conference on Robotics and Automation. Los Alamitos: IEEE Computer Society Press, 2022: 8748-8755
- [5] Schwarz S, Preda M, Baroncini V, *et al.* Emerging mpeg standards for point cloud compression[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2019, 9(1): 133-148
- [6] Schnabel R, Klein R. Octree-based point-cloud compression[C] //Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics. Goslar: Eurographics Association, 2006: 111-121
- [7] Quach M, Valenzise G, Dufaux F. Improved deep point cloud geometry compression[C] //Proceedings of the 22nd International Workshop on Multimedia Signal Processing. Los Alamitos: IEEE Computer Society Press, 2020: 1-6

- [8] Wang J Q, Ding D D, Li Z, *et al.* Multiscale point cloud geometry compression[C] //Proceedings of the Data Compression Conference. Los Alamitos: IEEE Computer Society Press, 2021: 73-82
- [9] Yan W, Shao Y T, Liu S, *et al.* Deep autoencoder-based lossy geometry compression for point clouds[OL]. [2022-11-30]. <https://arxiv.org/abs/1905.03691>
- [10] Wiesmann L, Milioto A, Chen X Y L, *et al.* Deep compression for dense point cloud maps[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 2060-2067
- [11] Thomas H, Charles R Q, Deschaud J E, *et al.* KPConv: flexible and deformable convolution for point clouds[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 6410-6419
- [12] Gao L Y, Fan T Y, Wan J Q, *et al.* Point cloud geometry compression via neural graph sampling[C] //Proceedings of the IEEE International Conference on Image Processing. Los Alamitos: IEEE Computer Society Press, 2021: 3373-3377
- [13] He Y, Ren X L, Tang D H, *et al.* Density-preserving deep point cloud compression[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 2323-2332
- [14] Cai C L, Chen L, Zhang X Y, *et al.* Efficient variable rate image compression with multi-scale decomposition network[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 29(12): 3687-3700
- [15] Chen T, Ma Z. Variable bitrate image compression with quality scaling factors[C] //Proceedings of the International Conference on Acoustics, Speech and Signal Processing. Los Alamitos: IEEE Computer Society Press, 2020: 2163-2167
- [16] Yang F, Herranz L, Van De Weijer J, *et al.* Variable rate deep image compression with modulated autoencoder[J]. IEEE Signal Processing Letters, 2020, 27: 331-335
- [17] Lu D N, Xie Q, Wei M Q, *et al.* Transformers in 3D point clouds: a survey[OL]. [2022-11-30]. <http://arxiv.org/abs/2205.07417>
- [18] Engel N, Belagiannis V, Dietmayer K. Point transformer[J]. IEEE Access, 2021, 9: 134826-134840
- [19] Zhang C, Wan H C, Shen X Y, *et al.* PVT: point - voxel transformer for point cloud learning[J]. International Journal of Intelligent Systems, 2022, 37(12): 11985-12008
- [20] Qiu S, Anwar S, Barnes N. PU-transformer: point cloud up-sampling transformer[C] //Proceedings of the 16th Asian Conference on Computer Vision. Heidelberg: Springer, 2022: 326-343
- [21] Yu L Q, Li X Z, Fu C W, *et al.* PU-net: point cloud upsampling network[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 2790-2799
- [22] Li R H, Li X Z, Fu C W, *et al.* PU-GAN: a point cloud upsampling adversarial network[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 7202-7211
- [23] Qian G C, Abualshour A, Li G H, *et al.* PU-GCN: point cloud upsampling using graph convolutional networks[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2021: 11678-11687
- [24] Liang Z J, Liang F. TransPCC: towards deep point cloud compression via transformers[C] //Proceedings of the International Conference on Multimedia Retrieval. New York: ACM Press, 2022: 1-5
- [25] Cui Z, Wang J, Gao S Y, *et al.* Asymmetric gained deep image compression with continuous rate adaptation[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2021: 10527-10536
- [26] Huang L L, Wang S L, Wong K, *et al.* Octsqueeze: octree-structured entropy model for lidar compression[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2020: 1310-1320
- [27] Chang A X, Funkhouser T, Guibas L, *et al.* ShapeNet: an information-rich 3d model repository[OL]. [2022-11-30]. <https://arxiv.org/abs/1512.03012>
- [28] Frank G, Michael H, Ondrej S, *et al.* Google/draco: a library for compressing and decompressing 3D geometric meshes and point clouds[EB/OL]. [2022-11-30] <https://github.com/google/draco>