arXiv:2303.11473v2 [eess.IV] 5 Jul 2023

夹心式视频压缩：利用神经网络封装器高效拓展标准编解码器的应用范围

Berivan Isik[*] 、*Onur G. Guleryuz*[†]、*Danhang Tang*[†]、*Jonathan Taylor*[†] 和 *Philip A. Chou*[†]

[*]斯坦福大学， [†]Google Research berivan.isik@stanford.edu,
{oguleryuz,danhangtang,jontaylor,philchou}@google.com

摘要

我们提出夹心式视频压缩——一种在标准视频编解码器两侧封装神经网络的视频压缩系统。夹心框架由神经网络预处理器和后处理器组成，中间夹有标准视频编解码器。我们联合训练这些网络，以优化率失真损失函数，旨在显著提升其在各种压缩场景中的表现，超越标准编解码器。在这种设置下进行端到端训练，需要为标准视频编解码器构建可微代理，其中包含结合运动补偿的时域处理、帧间／帧内模式决策以及环路滤波。我们针对视频编解码器的关键组件提出可微近似，并证明夹心结构的神经编码不仅能在标准编解码器基础上显著提升压缩效果，还能在两个重要场景中显著改善率失真性能。通过低分辨率 HEVC 传输高分辨率视频时，夹心系统相比标准 HEVC 可提升 6.5 dB。更重要的是，采用广为人知的感知相似度指标 LPIPS 衡量时，我们发现，在质量相同的情况下，其码率相比 HEVC 降低了 30%。最后，我们表明，由参数量极小、轻量级网络构成的预处理器和后处理器，也能取得接近上述结果的效果。[1]

1. 引言

过去十年间，神经图像压缩方法[6]逐渐超越了 JPEG2000 和 HEVC 等传统图像压缩器 [23]，并取代它们成为最先进的图像压缩器。神经图像压缩器之所以能够取得成功，是因为它们能够执行复杂的非线性变换，而这些变换是通过反向传播 [5]进行端到端学习的。近来，首批学习式视频压缩器已被提出 [20] ，神经视频压缩器的改进研究也随之加速 [3, 13, 19, 28]。现有策略主要侧重于学习如何预测光流、利用预测的光流对先前重建帧进行扭曲，以及进行残差补偿；这些过程均由端到端训练的神经网络完成。这导致网络变得非常复杂，拥有数千万个参数 [13, 19], ，每像素需要数千万次浮点运算，并且在高端图形处理器上训练需要数周时间 [16]。

在这项工作中，我们提出了一种计算效率更高的方法，利用神经网络改进任意标准视频编解码器。具体而言，我们将一个标准视频编解码器夹在



图 1。使用灰度编解码器（HEVC4:0:0）的玩具示例。夹心结构用于通过只能传输灰度的编解码器传输全色视频。图中展示了在时间 t、$t+1$ 和 $t+2$ 的神经编码帧经标准编解码器解码后的结果（压缩瓶颈）、后处理器生成的最终重建结果，以及原始源视频。码率=00.50 每像素比特数，峰值信噪比=393dB，帧/秒=30。请注意，夹心结构在瓶颈表示上建立了时间一致的类似调制的模式，预处理器通过这些模式编码颜色，随后由后处理器解调，从而得到全彩色结果。

轻量级神经网络预处理器和后处理器，并对其进行端到端优化。我们的方法利用了标准编解码器现有的高度优化计算实现，以及网络节点对标准比特流进行的无缝网络传输，这些节点已经熟悉此类流量。所提出的神经网络预处理器和后处理器能够学习如何对标准编解码器的输入和输出执行非线性变换，以实现最佳率失真性能；与此同时，标准视频编解码器则通过对神经预处理器的输出执行常规视频编码操作来承担繁重的工作。我们的工作主要旨在使标准编解码器适用于超出其设计直接适用范围的压缩场景。为此，我们关注的不仅是

[1]第一作者在 Google 实习期间完成的工作。[1]发表于 2023 年国际图像处理会议（ICIP）。

显著的改进，也包括理解神经网络如何实现这些改进。由于这些网络必须生成与原始视觉场景相关的、可观看的8位视频，因此通常可以观察到它们的运行方式，即便只能进行粗略观察（例如，参见图1中的玩具示例）。

图2概述了我们的方法。对于第 i 个视频片段，其帧为 $t = 0, 1, 2, ...$，源图像 $\{I_i(t)\}$ 由神经预处理器以时间上相互独立的方式处理，得到“瓶颈”图像 $\{L_i(t)\}$。瓶颈图像包含表示源图像的神经潜在编码，但仍采用 4:4:4（或 4:2:0）格式，可由标准视频编解码器进行压缩。标准视频编解码器将瓶颈图像压缩为重建的瓶颈图像 $\{\hat{L}_i(t)\}$，同样采用 4:4:4 格式。最后，重建的瓶颈图像经神经后处理器处理，得到重建的源图像 $\{\hat{I}_i(t)\}$。为训练神经网络预处理器和后处理器，我们用可微代理（以黄色阴影显示）替代标准视频编解码器，并最小化总率失真拉格朗日函数 $\sum_{i,\ t} D_i(t) + \lambda R_i(t)$，其中 $D_i(t)$ 和 $R_i(t)$ 分别是视频片段 i 中帧 t 的失真和码率。评估时，使用标准视频编解码器，而非代理。

我们的框架受到先前夹心式图像压缩研究的启发 [9, 10], , 该研究同样旨在提升相较于标准编解码器的率失真性能，同时利用其实现方面的便利性。端到端学习式图像编解码器在率失真性能上优于标准图像编解码器 [6, 7, 12, 26] ，但需要过度参数化的网络和熵模型，因而会消耗大量时间、功率和资源。[9, 10] 提出了一种解决方案，即在标准图像编解码器周围包裹神经网络。要端到端训练夹心结构，需要为标准图像编解码器构建一个可微代理。作者训练夹心模型时，将标准图像编解码器替换为可微 JPEG 代理：该代理使用 8×8 离散余弦变换（DCTs）和直通量化[2] 作为量化步骤的代理，并使用可微函数近似熵编码步骤的码率。研究者展示了，经过训练后，预处理器和后处理器能够通过彼此通信：它们发送以空间调制模式形式呈现的神经编码，以表示原始输入中的细节。这些模式易于通过标准编解码器压缩，同时能够抵御其造成的退化。与单独使用标准编解码器相比，当输入特征与标准编解码器的预期适用对象不同时，性能提升尤其显著。具体而言，[9] 在编码多光谱图像或法线贴图图像时取得了较大提升；而 [10] 在使用低动态范围（LDR）或低分辨率（LR）编解码器编码高动态范围或高分辨率图像时也取得了较大提升。

我们的动机和方法与 [9, 10] 类似，但在本文中，我们实现了一项根本性突破：将夹心方法从图像扩展到视频。这一突破面临诸多挑战。首先，对视频帧分别进行预处理和后处理可能会带来问题，因为瓶颈图像中的神经编码需要在时间上保持连贯，视频编解码器才能利用运动补偿。目前完全无法确定类似调制的高频模式能否通过运动补偿得到良好表示。其次，标准视频编解码器 [11,17, 24, 25, 27, 29] 比图像编解码器复杂得多，涉及的不仅是运动估计和运动补偿，还包括模式选择、环路滤波以及许多其他方面。在本文中，我们探讨了这些问题，并展示了夹心式视频压缩的结果，这些结果与此前夹心式图像压缩的结果相呼应，其中包括多光谱应用（使用灰度视频编解码器压缩彩色视频）和超分辨率应用（使用标准分辨率视频编解码器压缩高分辨率视频）。此外，我们仅用约 100K 个参数的网络（预处理器与后处理器合计），就实现了优于标准编解码器的效果；与最先进的神经视频压缩器相比，复杂度降低了两个数量级。在我们之前，只有少数研究尝试将标准视频编解码器夹在神经处理器之间。其中一些研究，例如 [4,8, 18], , 采用的方式使前后处理器可以独立使用，因此没有充分利用前后处理器之间可用的通信。我们的贡献可总结如下：

(1) 我们提出了一种夹心式视频压缩框架。由于神经网络结构简单，该框架能够更高效地进行训练和推理，同时仍能利用神经网络学习非线性变换的灵活性。

(2) 我们提出了一种可微视频编解码器代理，可在训练循环中有效使用。

(3) 我们证明，在低分辨率传输中，夹心式 HEVC 相比标准 HEVC 4:4:4 提升了 6.5 dB（对灰度图像进行彩色编码时，相比 4:0:0 提升了 8 dB）。此外，使用感知相似度指标学习感知图像块相似度（LPIPS）进行优化和测试时，我们观察到，在质量相同的情况下，码率相比标准 HEVC 4:4:4 降低了 $\sim 30\%$。

2. 方法

2.1. 视频编解码器代理

我们的 video codec proxy 如图2所示，它将一组输入图像（例如一组图像，或 GOP）映射为一组输出图像，并为每幅图像生成相应的码率。我们的 video codec proxy 的细节可概括如下。

帧内编码。视频编解码器代理使用来自 [9, 10] 的图像编解码器代理，模拟对图像组中的第一帧（$t = 0$），即 I 帧进行编码。

运动补偿。视频编解码器代理模拟通过对前一帧进行运动补偿，来预测图像组中的后续每一帧（$t > 0$），即 P 帧；前一帧可以是 I 帧，也可以是 P 帧。运动补偿使用“真实值”稠密运动流场来执行，该流场由最先进的光流估计器 UFlow [15], 在原始源图像 $I_i(t)$ 和 $I_i(t-1)$ 之间运行得到。视频代理只需将这个真实值运动流场应用于前一帧重建得到的 $\hat{}$ ~ 瓶颈图像 $L_i(t-1)$，从而获得瓶颈图像 $L_i(t)$ 的帧间预测 $L_i(t)$。这模拟了标准编解码器在瓶颈图像上进行的运动补偿。请注意，我们的运动补偿代理实际上并不依赖于 $L_i(t)$，因此，$\hat{}$ ~ 尽管它是一种扭曲，但它是一个从 $L_i(t-1)$ 到 $L_i(t)$ 的线性映射，且具有常数雅可比矩阵。与尝试使用同时依赖 $L_i(t-1)$ 和 $L_i(t)$ 的可微函数相比，这使优化变得容易得多；例如，UFlow 本身会寻找并应用一个从 $\hat{}$ $L_i(t-1)$ 到 $L_i(t)$ 的扭曲。众所周知，此类函数的雅可比矩阵波动很大，会导致训练困难。

预测模式选择。我们的视频编解码器代理还会模拟帧间/帧内预测模式的决策，从而更好地处理视频中的时间遮挡/未遮挡区域。首先，通过对当前帧进行初步压缩并对其进行低通滤波，来模拟帧内预测。这样可以模拟滤波过程，尽管并非采用通常的方向滤波来

[2] 直通量化器是一种由软件实现的函数 $y = Q(x)$，它在前向过程中执行常规量化，但在反向过程中假定其梯度为 dy/dx 1，而非 0。

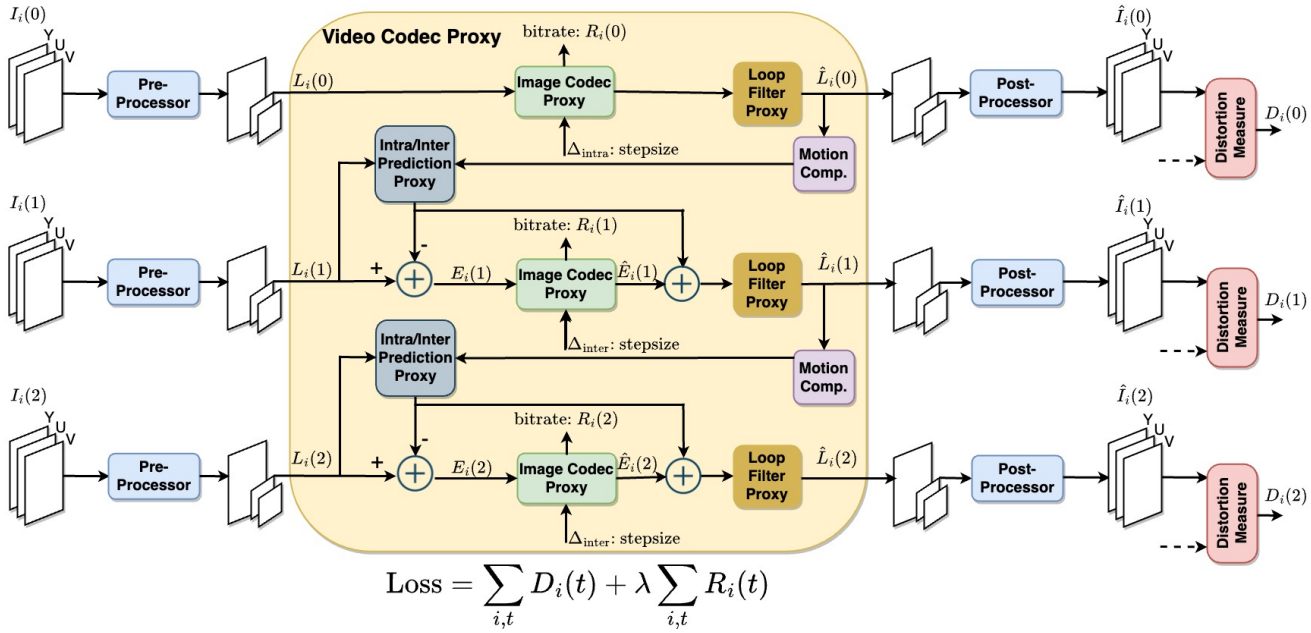

图2。三明治架构　　训练期间。阴影框（视频编解码器代理）在推理时会替换为标准视频编解码器　　.

根据相邻块预测每个块。初步压缩可确保帧内预测代理在极低码率下不会获得不当的优先选择。对于每个块，都会比较帧内预测（通过空间滤波得出）与帧间预测（通过运动补偿得出），并根据哪种预测最接近输入块来确定预测模式。

残差编码。预测图像由帧内预测块和帧间预测块组合而成，将其从瓶颈图像中减去，得到预测残差。随后，使用 [9, 10] 的图像编解码器代理压缩残差图像。再将压缩后的残差加回预测图像，得到瓶颈图像的“滤波前”重建图像 $\hat{}\, L_i(t)$。

环路滤波。“滤波前”重建图像随后由环路滤波器进行滤波，以获得最终的瓶颈图像重建图像 $\hat{L}_i(t)$。该环路滤波器由一个小型 U-Net((8);(8, 8))[21] 实现，每次处理一个通道；它仅使用视频编解码器代理，基于自然视频训练一次，针对四个码率点进行训练，并采用率失真（ℓ_2）训练损失，以模拟常见的环路滤波器。所得的滤波器串联组合在所有模拟中均保持固定；也就是说，针对四个码率点分别独立训练后，环路滤波器不再继续训练。

码率代理。与 [9, 10], 类似，每帧的可微码率可表示为 $R(E_i(j)) = a \sum_{k,\ l} \log(1 + |e_l^{(k)}|/\Delta)$，其中 $e_l^{(k)}$ 是 DCT 系数第 k 个块中的第 l个系数，而 a的取值使得 $R(E_i(j))$等于 JPEG 以均匀步长 Δ对 $E_i(j)$进行编码时的实际码率。 [22]中给出了另一种代理方法。

2.2. 前处理器与后处理器

前处理器与后处理器通过与 U-Net 并行的 1×1 卷积网络（逐像素运行的多层感知机）实现。其基本思路是，1×1 卷积网络负责学习恰当的色彩空间转换和色调映射，而 U-Net 则添加或重新移除作为神经编码的空间调制模式或水印。我们报告了使用两种不同复杂度 U-Net 的结果。作为 U-Net 编码器/解码器滤波器，高复杂度网络采用 U-Net((32, 64, 128, 256); (512, 256, 128, 64, 32))（$\sim$ 8M 个参数）。低复杂度的精简网络采用 U-Net((32); (32, 32))（$\sim$ 57 千个参数），参数量减少了 $\sim$ **99%**。

2.3. 损失函数

如第1节所述，视频编解码器代理中的预处理器和后处理器以及环路滤波器经过训练，以最小化总率失真拉格朗日函数$\sum_{i,\ t} D_i(t) + \lambda R_i(t)$，其中 $D_i(t)$和$R_i(t)$分别是视频片段 i中第 t 帧的失真和码率。具体而言，码率项旨在促使预处理器和后处理器生成具有时间一致性的神经编码，因为按照我们的“真实值”运动流场移动的神经编码能够得到良好的预测，从而减小残差和码率项。正如图1所示，我们确实观察到生成的神经编码似乎具有一定程度的时间一致性。请注意，从输入图像经过预处理器、视频编解码器代理、后处理器和损失函数的整体映射是可微的。

3. 实验

3.1. 设置与数据集规范

我们生成了一个视频数据集，其中包含来自 AV2通用测试条件 [1] 的视频的 10 帧视频片段，以及使用 UFlow [15]计算得到的相应运动流。我们使用批量大小为 8，即每个批次包含 8 个视频片段。在数据集生成步骤中，每个片段都经过处理，使其包含 10 帧、每帧大小为 256x256，并选自帧/秒为 20-40 的视频。HEVC 使用 x265 实现（IPP.., 单参考帧，开启率失真优化量化和环路滤波器）。我们在三种设置下将夹层模型与标准编解码器 HEVC 进行比较：(1) YUV 4:0:0格式，采用单通道（灰度）瓶颈表示，如图 4 所示；



图 3。HEVC 低分辨率视频的夹心结构：压缩瓶颈中的帧、夹心结构的重建结果、原始源视频以及 HEVC 低分辨率视频。请注意人群和建筑物的清晰度有了显著提升。虽然此处未展示，但由于神经预处理器嵌入了信息，夹心结构系统相较于经过神经后处理的 HEVC 低分辨率视频也取得了显著提升。

(2) YUV 4:4:4 格式的低分辨率（LR）瓶颈表示，如图 5 所示；
(3) 使用学习感知图像块相似度（LPIPS）——文献中常用的一种感知相似度指标 [30, 14]的 YUV 4:4:4 格式。每种设置下，模型训练 1000 轮次，学习率为 1e-4，并在 120 个测试视频片段上进行测试。所有率失真图均基于整个测试集。使用 ℓ_2 范数时，我们以 YUV 峰值信噪比报告结果；使用 LPIPS 时，则以 “LPIPS (RGB)PSNR”报告结果。由于 LPIPS 是为 RGB 设计的，因此在后一种情况下，我们将最终解码的 YUV 视频转换为 RGB，然后计算 LPIPS。为使结果处于大致相近的尺度，我们为 LPIPS 推导了一个固定的线性缩放因子，使得对于图像向量 x，y，

$$sLPIPS(x,y) \sim ||x-y||^2, \text{if } ||x-y||^2 < \tau \qquad (1)$$

其中 τ 是一个较小的阈值，s 是 LPIPS 线性缩放器。一个视频片段的 LPIPS 损失是 10 帧的 LPIPS 损失的平均值。

3.2. 结果

YUV 4:0:0。图 1 和图 4 展示了夹层式 HEVC YUV 4:0:0 编解码器的主观和定量结果。从图 1 中的瓶颈帧可以看出，夹层模型能够保持帧间的时间一致性，使标准编解码器 HEVC 能够利用调制模式，获得更优的率失真点。从定量结果来看，图 4 显示，夹层模型相比 HEVC YUV400 有显著提升（提高 8 dB）。此外，图中还列出了 HEVC 444 及其神经增强版本作为参考。后者在压缩方面取得了有意义但相对较小的提升，码率比前者降低了 5%。

低分辨率（LR）。图3和图5展示了采用夹心结构的 HEVC YUV 4:4:4低分辨率编解码器的主观比较和定量结果。标准低分辨率编解码器传输的是由原始视频经线性下采样（双三次插值）得到的低分辨率视频。解码视频经线性

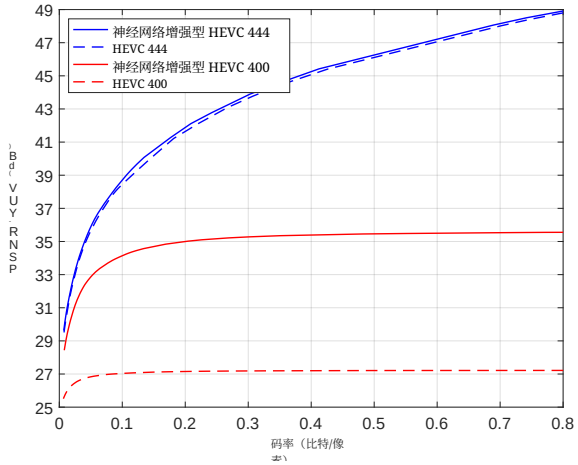

图 4。YUV 4:0:0夹层编码的率失真性能 .

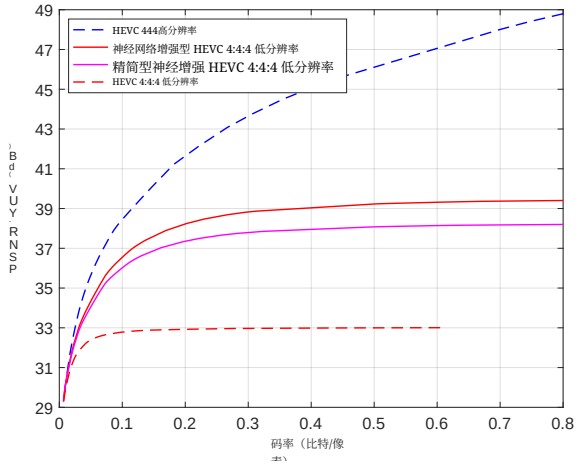

图 5。YUV 4:4:4低分辨率（LR）夹层编码的率失真性能。（HEVC 444 HR 指 HEVC YUV 4:4:4高分辨率。）

上采样（Lanczos3）。夹心结构采用完全相同的系统，只是使用生成的瓶颈表示。图5中，夹层模型采用 HEVC YUV 4:4:4低分辨率编解码器，相较于标准HEVC YUV 4:4:4低分辨率编解码器提升了 6.5 dB；图3所示的主观比较也体现了这一点，夹心结构在标准 HEVC YUV 4:4:4低分辨率编解码器产生模糊结果的区域，能够输出细节丰富得多的画面。

感知相似度指标 - LPIPS。最后，我们使用 TensorFlow [2] 对 [30] 的实现，以 LPIPS 为目标训练夹心结构 HEVC YUV 4:4:4，并将其与以 LPIPS 为目标的标准 HEVC YUV 4:4:4 进行比较。结果如图 6 和图 7 所示。请注意，在广泛的范围内，夹层模型相比标准 HEVC 可实现 $\sim 30\%$ 的码率改善。通过观看解码视频，我们观察到，夹心系统在获得稳定码率降低的同时，视觉差异几乎无法察觉（例如图 6）。由于视频片段通常描绘的是同一场景，只存在轻微的几何变换和颜色变换，因此所采用的



图 6。采用 LPIPS 的 HEVC：夹心结构重建结果中的帧间帧、原始源视频，以及 HEVC 重建结果（右下角的误差放大了 10 倍）。请注意夹心结构结果的码率降低。尽管绝对误差更大，我们仍未观察到具有感知意义的差异。

片段平均 LPIPS 损失预计对该场景具有鲁棒性，类似于 [14] 的稳健化集成 LPIPS 损失。

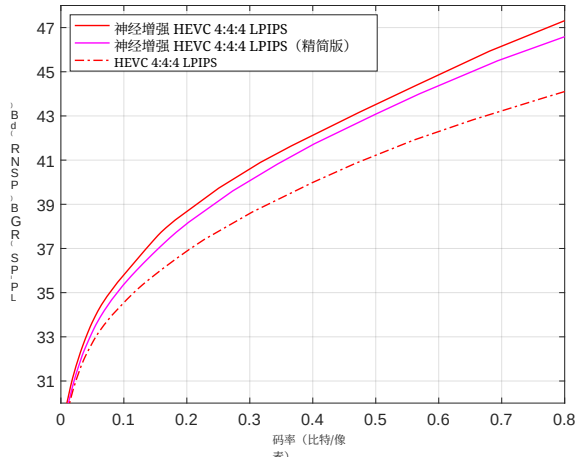

图 7。采用 LPIPS 训练和测试的 HEVC YUV 4:4:4 夹心结构的率失真性能。在广泛的码率范围内，神经增强 HEVC YUV 4:4:4 的性能比标准 HEVC 高 ∼ 30%。

4. 结论

在本研究中，我们通过精心设计视频编解码器代理，并以端到端方式训练神经网络预处理器和后处理器，将三明治架构扩展到视频压缩。所提出的夹层模型在多种设置下均优于标准编解码器 HEVC，包括 YUV 4:0:0 和 YUV 4:4:4 LR 格式；在 ℓ_2 和 LPIPS 失真条件下，分别实现了 8 dB （4:0:0）、 6.5 dB （4:4:4 LR）的增益，以及 ∼ 30% 的码率（LPIPS）改进。我们还表明，采用仅含 57K 个参数的精简轻量网络，也能获得与上述结果非常接近的表现。在这些场景中，三明治系统不仅能实现显著优于标准视频编解码器的率失真性能，还能通过使用轻量网络，在提升性能的同时保持计算效率。我们的结果清楚地表明，三明治系统能够让标准编解码器适用于其原始设计范围之外的压缩场景，从无缝提升分辨率到针对领先的感知质量指标进行优化，且均能取得显著改进。

5. 参考文献

[1] AOM 通用测试条件 v2.0。http://aomedia.org/docs/CWG-B075o_AV2_CTC_v2.pdf。访问日期：2022-07-11。[2] M. Abadi、A. Agarwal、P. Barham、E. Brevdo、Z. Chen、C. Citro、G. S. Corrado、A. Davis、J. Dean、M. Devin、S. Ghemawat、I. Goodfellow、A. Harp、G. Irving、M. Isard、Y. Jia、R. Jozefow- icz、L. Kaiser、M. Kudlur、J. Levenberg、D. Mané、R. Monga、S. Moore、D. Murray、C. Olah、M. Schuster、J. Shlens、B. Steiner、I. Sutskever、K. Talwar、P. Tucker、V. Vanhoucke、V. Vasudevan、F. Viégas、O. Vinyals、P. Warden、M. Wattenberg、M. Wicke、Y. Yu 和 X. Zheng。TensorFlow：异构系统上的大规模机器学习，2015。软件可从 tensorflow.org 获取。[3] E. Agustsson 、 D. Minnen 、 N. Johnston、J. Balle、S. J. Hwang 和 G. Toderici。用于端到端优化视频压缩的尺度空间流。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第 8503–8512 页，2020。[4] Y. Andreopoulos。采用 AVC、VP9 和 AV1 进行视频编码的神经网络预处理与后处理。载于*AOM* 研究研讨会，2022。[5] J. Ballé、P. A. Chou、D. Minnen、S. Singh、N. Johnston、E. Agustsson、S. J. Hwang 和 G. Toderici。非线性变换编码。*IEEE* 信号处理精选主题期刊，15(2)：339–353，2020。[6] J. Ballé、V. Laparra 和 E. P. Simoncelli。面向感知质量的非线性变换编码端到端优化。载于*2016* 年图像编码研讨会（*PCS*），第 1–5 页。IEEE，2016。[7] J. Ballé、D. Minnen、S. Singh、S. J. Hwang 和 N. Johnston。采用尺度超先验的变分图像压缩。*arXiv* 预印本 *arXiv:1802.01436*，2018。[8] P. Eusébio、J. Ascenso 和 F. Pereira。基于深度学习的预处理与后处理图像编码框架优化。载于欧洲信号处理会议（*EUSIPCO*），第 506–510 页，2021。[9] O. G. Guleryuz、P. A. Chou、H. Hoppe、D. Tang、R. Du、P. David- son 和 S. Fanello。夹心式图像压缩：在标准编解码器外封装神经网络。载于*2021* 年 *IEEE* 国际图像处理会议（*ICIP*），第 3757–3761 页。IEEE，2021。[10] O. G. Guleryuz、P. A. Chou、H. Hoppe、D. Tang、R. Du、P. David- son 和 S. Fanello。夹心式图像压缩：提高标准编解码器的分辨率和动态范围。载于*2022* 年图像编码研讨会（*PCS*），第 175–179 页，2022。[11] J. Han、B. Li、D. Mukherjee、C.-H. Chiang、A. Grange、C. Chen、H. Su、S. Parker、S. Deng、U. Joshi 等。AV1 技术综述。*IEEE* 论文集，109(9)：1435–1462，2021。[12] Y. Hu、W. Yang、Z. Ma 和 J. Liu。学习端到端有损图像压缩：一项基准测试。*IEEE* 模式分析与机器智能汇刊，2021。[13] Z. Hu、G. Lu 和 D. Xu。FVC：一种面向特征空间深度视频压缩的新框架。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第 1502–1511 页，2021。[14] M. Kettunen、E. Härkönen 和 J. Lehtinen。E-LPIPS：通过随机变换集成实现稳健的感知图像相似度。*CoRR*，abs/1906.03973，2019。

[15] S. K. Lodha、A. Pang、R. E. Sheehan 和 C. M. Wittenbrink。UFlow：流体流动中的不确定性可视化。载于第七届 *IEEE Visualization'96* 年会论文集，第 249–254 页。IEEE，1996。[16] G. Lu、W. Ouyang、D. Xu、X. Zhang、C. Cai 和 Z. Gao。Dvc：一种端到端深度视频压缩框架。载于 *2019* 年 *IEEE/CVF* 计算机视觉与模式识别会议（*CVPR*），第 10998–11007 页，2019。[17] D. Mukherjee、J. Bankoski、A. Grange、J. Han、J. Koleszar、P. Wilkins、Y. Xu 和 R. Bultje。最新开源视频编解码器 VP9：概述与初步结果。载于 *2013* 年图像编码研讨会（*PCS*），第 390–393 页。IEEE，2013。[18] K. Qiu、L. Yu 和 D. Li。用于视频编码联合优化及预处理与后处理的编解码器仿真网络。*IEEE*开放期刊：电路与系统，2：648–659，2021。[19] O. Rippel、A. G. Anderson、K. Tatwawadi、S. Nair、C. Lytle 和 L. Bourdev。Elf-vc：高效的学习式灵活码率视频编码。载于*IEEE/CVF* 国际计算机视觉会议论文集，第 14479–14488 页，2021。[20] O. Rippel、S. Nair、C. Lew、S. Branson、A. G. Anderson 和 L. Bourdev。学习式视频压缩。载于*IEEE/CVF* 国际计算机视觉会议论文集，第 3454–3463 页，2019。[21] O. Ronneberger、P. Fischer 和 T. Brox。U-Net：用于生物医学图像分割的卷积网络。载于医学图像计算与计算机辅助干预——*MICCAI 2015*，第 234–241 页，Cham，2015。[22] A. Said、M. K. Singh 和 R. Pourreza。用于基于神经网络的视频编解码器增强的可微比特率估计。载于*2022* 年图像编码研讨会（*PCS*），第 379–383 页，2022。[23] 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，22(12)：1649–1668，2012。[24] G. J. Sullivan、P. N. Topiwala 和 A. Luthra。H.264/AVC 高级视频编码标准：概述及保真度范围扩展介绍。数字图像处理应用 *XXVII*，5558：454–474，2004。[25] V. Sze、M. Budagavi 和 G. J. Sullivan。高效视频编码（HEVC）。载于集成电路与系统、算法及架构，第 39 卷，第 40 页。施普林格，2014。[26] G. Toderici、D. Vincent、N. Johnston、S. Jin Hwang、D. Minnen、J. Shor 和 M. Covell。利用循环神经网络进行全分辨率图像压缩。载于*IEEE* 计算机视觉与模式识别会议论文集，第 5306–5314 页，2017。[27] P. Tudor。MPEG-2 视频压缩。电子与通信工程期刊，7(6)：257–264，1995。[28] R. Yang、Y. Yang、J. Marino 和 S. Mandt。用于神经网络视频压缩的分层自回归建模。*arXiv* 预印本 *arXiv:2010.10258*，2020。[29] J. Zhang、C. Jia、M. Lei、S. Wang、S. Ma 和 W. Gao。AVS 视频编码标准的最新进展：AVS3。载于*2019* 年图像编码研讨会（*PCS*），第 1–5 页。IEEE，2019。[30] R. Zhang、P. Isola、A. A. Efros、E. Shechtman 和 O. Wang。深度特征作为感知度量的惊人有效性。载于*IEEE* 计算机视觉与模式识别会议论文集，第 586–595 页，2018。