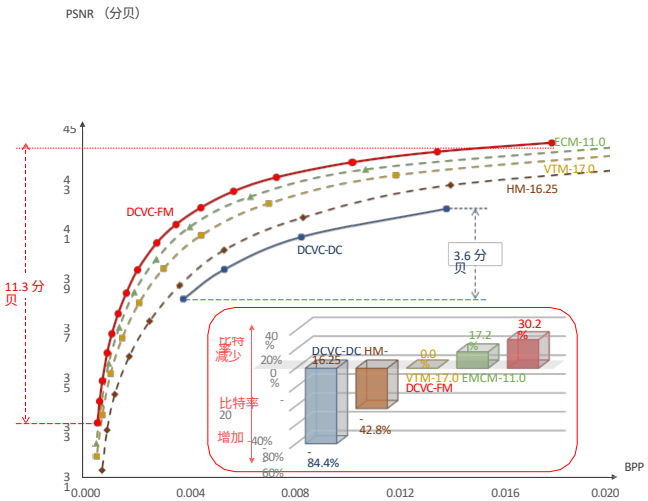
采用特征调制的神经视频压缩

李家豪、李斌、卢彦 微软亚洲研究院
{li.jiahao, libin, yanlu}@microsoft.com

摘要

新兴的基于条件编码的神经视频编解码器（NVC）显示出优于常用的基于剩余编码的编解码器，最新的 NVC 已经宣称优于最好的传统编解码器。然而，仍有一些关键问题阻碍着神经视频编解码器的实用性。在本文中，我们提出了一种功能强大的基于条件编码的 NVC，通过特征调制解决了两个关键问题。首先是如何在单一模型中支持宽质量范围。以往具有这种功能的 NVC 平均只能支持约 3.8 dB 的 PSNR 范围。为了解决这一限制，我们对当前图像的潜在特征进行了调制。

通过可学习的量化标度器，我们可以对每个帧进行量化。在训练中，我们专门设计了均匀量化粒度采样机制，以提高编码和量化的协调性。这使得量化缩放器的学习效果更好，并帮助我们的 NVC 支持约 11.4 dB 的 PSNR 范围。其次是如何使 NVC 在长预测链下仍然有效。我们发现，之前的 SOTA NVC 在使用较大的周期内设置时存在明显的质量下降问题。为此，我们建议通过周期性刷新机制来调节时间特征，从而提高质量。值得注意的是，在单帧内设置下，我们的编解码器比以前的 SOTA NVC 节省了 29.7% 的比特率，减少了 16% 的 MACs。我们的编解码器是无损压缩演进过程中的一个显著里程碑。代码见 https://github.com/microsoft/DCVC。



1. 导言

传统的标准编解码器依赖于基于残差编码的混合框架，已经发展了 30 多年，目前仍在不断改进。然而，压缩率的提高已经减弱，而复杂性却显著增加 [43]。这使得在传统框架内取得进一步进展变得越来越具有挑战性。最近，神经视频编解码器（NVC）受到了广泛关注，因为它有可能打破这一发展瓶颈。

图 1.与 H.265/HM、H.266/VTM、ECM 和之前的 SOTA NVC DCVC-DC [26] 的速率-失真曲线和 BD-Rate 比较。测试数据集为 HEVC E（600 帧），采用单一帧内设置（即周期内= -1）和 YUV420 色空间。在此设置下，DCVC-DC 的性能大幅下降，而我们的新编解码器 DCVC-FM 仍能显著超越 ECM。同时，我们的 DCVC-FM 的质量范围要比 DCVC-DC 大得多。

早期的 NVC 模型 DVC [32]仍沿用传统的编解码器，并使用基于残差编码的框架。后来，许多作品 [5, 19, 27, 28] 也是基于这种模式，并提出了更强子模块来提高性能。相比之下，新兴的条件编码 [22, 24] 比残差编码显示出更低的熵限，具有更大的潜力。条件可以自由地去掉和学习，而不是局限于像素域中的预测帧。同时，该条件可灵活地用于编码、解码和熵调节。最近的 DCVC-DC 模型 [26] 已经通过挖掘不同的背景作为条件，实现了比 H.266/VTM [12] 和正在开发中的 ECM（下一代传统标准的原型）更好的压缩比。

尽管 DCVC-DC 取得了进步，但我们发现它离实际应用还很远。第一个阻碍问题是缺乏对宽质量范围的支持。虽然 DCVC- DC 可以在单个模型中支持多个质量级别，但其质量范围相当有限，只有平均

在不同的数据集上，该值为 3.8 dB，这肯定无法满足实际产品的需要。为了解决这个问题，我们利用可学习量化标度器对当前帧的潜在特征进行调制。在训练阶段，我们不仅增加了 lambda 范围，还专门设计了统一的量化参数采样机制，让 NVC 体验速率和失真之间的各种权衡。这可以增强编码和量化过程之间的协调性。因此，我们获得了精细可控的量化缩放器，我们的 NVC 可以在较宽的质量范围（即约 11.4 dB）内无缝调节质量水平。此外，由于支持宽质量范围，本文还展示了在给定目标比特率的情况下，在单一模型中进行速率控制的能力。这有效证明了我们的 NVC 在现实世界的实用性。

第二个问题是如何使 NVC 有效应对较长的预测链。现有的大多数 NVC 模型都难以解决其中的时间误差累积问题。为了缓解这一问题，许多模型 [17, 18, 32-34] 都依赖于使用较小的周期内设置（如 10 或 12）来更频繁地插入高质量的帧内。然而，较小的周期内设置损害整个压缩效率。，文献 [25, 44]显示，在 H.265/HM 中，周期内 32 比周期内 12 平均节省 23.8%的比特率。因此，传统的标准委员会[11]将周期内严格定义为-1，即整个视频只对一个帧内进行编码。我们认为 NVC 也应遵循这一设定。这样在比较 NVC 和传统编解码器时也更公平。然而，我们发现之前的 SOTA DCVC-DC 在周期内 -1 的设置下质量下降较大，如图 1 所示。为解决这一问题，我们提出了两种应对措施。其一是增加视频帧数，以便在训练过程中更好地学习长距离时间相关性。另一种方法是，我们建议通过定期刷新时间特征来修改时间特征，这样可以大大缓解误差传播。

利用这些有效的特征调制技术，我们在 DCVC-DC 的基础上建立了一种新的编解码器 DCVC-FM。此外，我们的 DCVC-FM 还进行了其他改进，以实现多功能 NVC。现有的大多数 NVC 只适用于 RGB 色彩空间。事实上，传统编解码器和实际应用主要采用 YUV 色彩空间。，我们设计了一种无需任何微调训练即可同时支持 RGB 和 YUV 的 NVC。此外，通过改进实现方式，本文还展示了低精度推理，这可以显著减少运行时间和内存成本，而压缩比的下降可以忽略不计。实验表明，我们的 DCVC- FM 在周期内-1 设置下的性能比 VTM 高出 25.5%，与 ECM 相比也取得了非同小可的优势。与之前的 SOTA NVC DCVC-DC 相比，DCVC- FM 的性能提高了 25.5%、比特率节省了 29.7%，而 MAC（多层累加操作）减少了 16%。

总之，我们的贡献是

- 我们通过可学习量化标度器对潜在特征进行调制，并提出了一种均匀量化参数采样机制来帮助其学习。这使得我们的 DCVC-FM 能够在单一模型中支持广泛的质量范围，并展示了速率控制能力。
- 我们不仅利用较长的视频进行训练，还通过定期刷新机制对时间特征进行模块化，以提高质量。这些都有助于我们的 DCVC-FM 处理长预测链。
- 为了进一步提高实用性，我们使 DCVC-FM 能够在一个单一模型内同时支持 RGB 和 YUV 色彩空间。此外，我们还演示了低精度推理，而比特率的增加可以忽略不计。
- 我们的 DCVC-FM 性能优于所有传统编解码器，且不存在周期内 -1 设置。与之前的 SOTA NVC 相比，比特率降低了 29.7%，MAC 降低了 16%。我们的编解码器是无损压缩技术发展的一个重要里程碑。

2. 相关工作

2.1. 神经图像压缩

最近的神经图像编解码器（NIC）模型大多遵循 hy- perprior [9]，采用分层框架设计。一些研究[21, 31, 41, 52]使用变换器来加强自动编码器或熵模型。最近，人们还探索了扩散模型 [15, 39, 46, 50]，以提高生成能力。此外，还提出了轻量级模型 [47, 51]。现在，NIC 的功能已经相当强大，其标准化过程也已在考虑之中 [8]。

2.2. 神经视频压缩

NIC 的成功也推动了 NVC 的发展。早期的 DVC [32] 采用基于残差编码的传统框架，使用 NIC 分别对运动矢量和残差进行编码。许多 NVC [5, 14, 19, 27-29, 34, 36, 42] 也采用了这种模式，并设计了更强大的子模块来提高压缩率。例如，为处理复杂的运动区域，提出了尺度空间光流估计 [5]。利用多个参考帧来改进时间预测 [27]。[28] 提出了基于块的预测模式选择。与残差编码相比，条件编码 [16, 23-26, 30, 37, 40, 44] 显示出更大的潜力，因为它的时间上下文并不局限于像素域的预测帧，也不依赖次优减法来减少冗余。特征域的时间上下文可以灵活设计，其与当前帧的相关性也可以自动学习。在 [30] 中，时间条件熵模型是

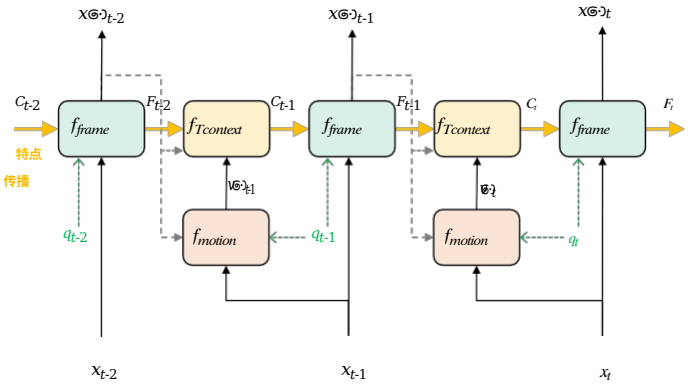


图 2.建立在 DCVC-DC 基础上的 DCVC-FM 框架。

设计。DCVC 系列 [24-26, 40, 44] 建议使用高维上下文来改进编码和解码，如
以及熵建模。DCVC-TCM [44] 引入了时间特征传播。DCVC-HEM [25] 设计了一个利用空间和时间背景的强大熵模型。最新的 DCVC-DC [26] 通过不断提高上下文多样性，已经超越了发展不足的 ECM。

然而，仍有几个关键问题阻碍着无损检测的实用性。首先是质量范围问题。尽管 DCVC-HEM 和 DCVC-DC 支持单一模式下的可变比特率，但其质量范围相当有限，无法满足各种质量要求。此外，包括先前的 SOTA DCVC-DC 在内的大多数现有 NVC 仍然使用较小的周期内设置（如 10、12 和 32），这与实际应用场景相去甚远。相比之下，我们提出的 DCVC-FM 通过特征调制解决了这两个关键问题。

3. 建议的方法

3.1. 概述

我们的 DCVC-FM 采用基于条件编码的工作框架，建立在 DCVC-DC [26] 的基础上。整体框架如图 2 所示。为编码每个输入帧 x_t（t 为帧索引），有三个主要函数：f_{motion}、$f_{Tcontext}$ 和 f_{frame}。首先，f_{motion} 利用光流网络估算 x_t 与上一重建帧 $\hat{x}_{t-1}$ 之间的运动矢量 v_t。v_t 需要编码、传输和解码为 $\hat{v}_t$。子项
通常，$f_{T\ (上下文)}$ 使用 $\hat{v}_t$来提取时间上下文 C_t
来自上一帧的传播特征 F_{t-1}。最后，以运动对齐的时间上下文 C_t 为条件，通过函数 f_{frame} 将 x_t 编码、传输和解码为 $\hat{x}_t$。同时，f_{frame} 生成下一帧所需的 F_t。与 DCVC-DC 相比，我们使支持质量范围从 3.8 dB 增加到 11.4 dB（第 3.2 节）。此外，为了实现

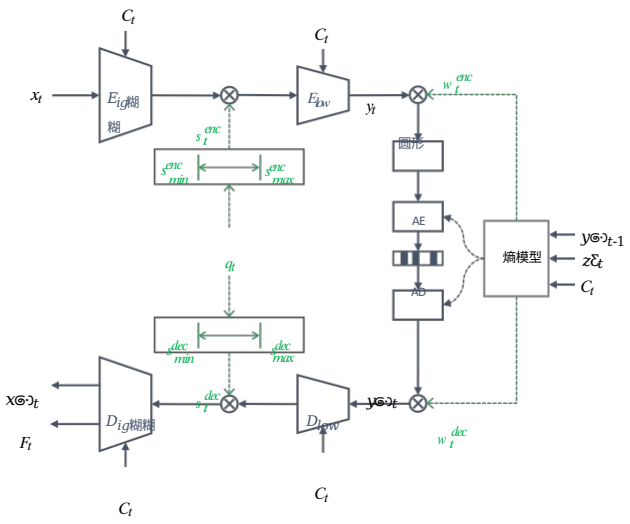


图 3.帧编码函数 f_{frame} 的框架。E_{high} 和 E_{low} 分别是高分辨率和低分辨率的编码器。D_{high} 和 D_{low} 是相应的解码器。AE 和 AD 是算术编码器和解码器。量化和反量化过程也以类似方式应用于 f_{motion}。

我们改进了特征传播机制（第 3.3 节），以有效地应对周期内 -1 的设置。第
3.4 展示了新功能的细节：RGB 和 YUV 色彩空间的单一模型，以及通过我们改进的实现方式进行的低精度推理。

3.2. 单一机型的广泛质量范围

基本的 量化和反量化过程可表述为

$$\hat{I} = QS \cdot \left\lfloor \frac{I}{QS} \right\rceil \quad (1)$$

I 和 QS 分别为输入值和量化步长。$\lfloor \cdot \rceil$是四舍五入运算。QS 控制输出 $\hat{I}$ 的重构质量。为了实现 NVC 的可变质量，我们还在编码和解码过程中加入了类似的机制来调节潜在特征。关键的挑战在于如何决定相应的 QS，并使其支持广泛的范围。

图 3 显示了我们设计的帧编码函数 f_{frame}。如图所示，在编码过程中，有两个值与量化步骤有关。
一个是 $s^{(enc)}$），另一个是 w^{enc}。它们都用于

s^{enc} 是全局量化标度，根据用户输入的量化参数 $q_{(t)}$生成，这与传统编解码器中 QP 的概念类似。q_t 是一个范围为 $[0, q\ num-1]$ 的整数标量，其中 $q\ num$ 是 q_t 值的可调数，设置为

64 在实现过程中。传统视频编解码器的量化步长会随着 QP 的线性增加而呈指数增长，与此类似，我们根据 q_t 和量化尺度范围 $[s^{(enc)}_{分钟}, s^{(enc)}_{最大}]$，通过下式对 s(enc) 进行插值：

$$s_t^{enc} = s^{enc}_{min} \cdot \left(\frac{s^{编码}_{max}}{s^{分钟}}()\right)^{\frac{q_t}{q\ num\ 1-}} \quad (2)$$

为了提高数值稳定性，我们将其改为

$$s_t^{加密} = e^{\ln s^{enc}_{分钟} + \frac{q_t}{q\ num\ 1-} \cdot (\ln s^{enc}_{最大} - \ln s^{(enc)}_{分钟})} \quad (3)$$

在我们的设计中 $s^{enc}_{分钟}$ 和 $s^{enc}_{最大}$ 是在培训期间学到的。

我们知道，编解码器的优化目标通常是 $Loss_{(RD)\ R} = + \lambda \cdot D$ [1]，其中 R 和 D 分别代表比特率和失真度。λ 用于控制 R 和 D 之间的权衡。目前，大多数现有的 NVC 都需要为每个质量级别训练一个单独的模型，并预先定义相应的恒定 λ 值。至于我们的模型，为了支持不同的质量水平，λ 在训练过程中也是可变的。我们预先定义的 λ 范围是 $[\lambda_{(最小)}, \lambda_{(最大)}]$。在我们的实现中，$[\lambda_{min}, \lambda_{max}]$被设置为[1, 768]。在每个训练步骤中，我们随机选择 q_t，并在此范围内对 λ 值进行内插。内插法与公式 3 类似，计算公式为

$$\lambda = e^{\ln \lambda_{min} + \frac{q_t}{q\ num\ 1-} \cdot (\ln \lambda_{max} - \ln \lambda_{min})} \quad (4)$$

对于每一步，我们从 [0, q *num-1*] 范围内均匀采样积分 q_t 值，然后通过公式 4 获得 λ 值，计算 $Loss_{RD}$。如果将公式 4 和公式 3，我们可以知道$[\lambda_{最小}, \lambda_{最大}]$将引导通过 $Loss_{RD}$ 的反向传播学习 $[s^{enc}_{最小值}, s^{enc}_{最大}]$。通过控制 λ 的取值范围，我们可以轻松调整量化标度的取值范围。通过对 q_t采用均匀采样机制，编解码器可以在训练过程中体验不同的 s^{enc}值，并探索 R 和 D 之间的各种权衡。这有助于编码和量化过程之间的协调，从而使我们的编解码器能够学习精细可控的量化缩放器 s^{enc}，以达到以下目的

t

调节潜在特征。

值得注意的是，与公式 1 不同，我们的编解码器没有在编码和解码过程中必须使用相同量化步长值的限制，因为我们的量化和反量化都是在拉帐篷特征域进行的。去掉这一限制，就能更灵活地平滑调整质量。因此，相应的 s^{dec} 可以在解码过程中单独学习。此外，正因为取消了这一限制，我们可以让 $s^{(enc)}_t$ 与潜在特征相乘来调制潜在特征，而不是在编

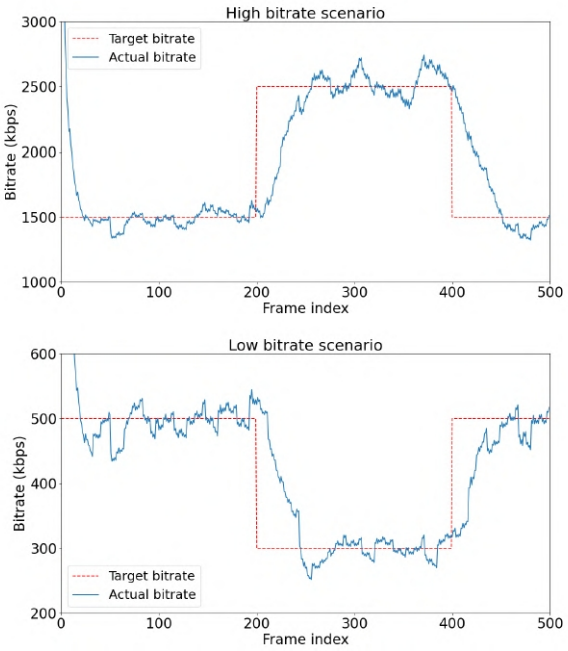


图 4：使用 BasketballDrive 视频信号的速率控制示例（1080p，50fps，500 帧使用 *BasketballDrive* 视频序列（1080p、50fps、500 帧）的速率控制示例。上图（下图）示例的目标比特率相对较高（较低）。

稳定培训过程，避免可能出现的问题

除以零的问题。

但是，所有空间位置的 $s^{(enc)}_t$ 都是相同的，这就意味着可能会忽略视频的不同空间特性

内容。因此，我们沿用 [20, 25] 的方法，使用熵模型来学习空间信道量化标度 $w^{(enc)}_t$。

不仅有助于实现每个位置的精确调制，还能适应每个帧的视频内容。这种内容自适应动态特征调制还能提高最终的压缩效率。

得益于这些先进的设计，我们的 DCVC-FM 终于可以支持大范围的质量调整。这种能力也是速率控制的先决条件，而速率控制是实用编解码器的核心功能。图 4 显示了两个具有不同目标比特率的速率控制示例。

情况。因为我们可以在编解码器中调整每个帧的 q *值*、

码过程中除数。这有助于

[1] 有时 λ 应用于 R。

如图 4 所示，可以支持波动的目标比特率。实际比特率接近目标比特率。值得注意的是，我们只是展示了使用 DCVC-FM 进行速率控制的可行性，而不是专注于为 NVC 设计新的速率控制算法。目前，我们只是简单地根据缓冲区的饱和度调整 q_t（更多细节补充材料）。未来，我们可以在 DCVC-FM 的基础上提出更先进的速率控制算法。

3.3. 长预测链

时间质量下降是所有视频编解码器都会遇到的基本问题，但对于无损压缩技术来说尤为严重。预

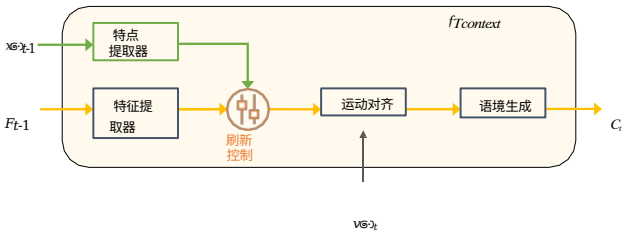


图 5.通过周期性刷新进行时间特征调制。

传统的 NVC 已经开始解决这一问题。例如，DCVC-DC 遵循传统的编解码器，引入了广泛使用的分层质量结构，以实现更高的质量。

从时间上提高质量。这有助于缓解误差传播，但还不够。如图 6 所示，当只使用单一帧内（即周期内 =-1）时，DCVC-DC 的质量会严重下降。为解决这一问题，我们提出了两种对策。值得注意的是，我们的 NVC 是基于条件编码的框架工作，其主要优势是使用时间特征作为上下文和条件。特别是，时间特征可以在许多帧中传播。因此，关键在于如何设计更有效的时间特征传播机制。我们的第一项改进是在训练过程中增加视频帧数。虽然这只是一个简单的修改，但却很有帮助。较长的视频可以在较长的时间距离内识别相似的模式，从而更好地探索时间相关性。

特征传播也是一把双刃剑，因为传播的特征可能会受到累积错误的污染，或包含一些不相关的信息。为此，我们建议对传播的特征进行调制，并强制 NVC 定期刷新（在实现过程中将刷新周期设置为 32）。如图 5 所示，对于输入帧 x_t，我们不会让 $f_{T\ (上下文)}$ 提取临时特征，而是让 NVC 提取临时特征。

从传播的特征 F $_{t-1}$ 如果我们因此，我们不需要对 x^(t)进行特征刷新。相反，我们使用单独的特征提取模块从 $\hat{x}_{t-1}$ 中提取时间背景。由于 $\hat{x}_{t-1}$ 只有 3 个二维的像素信息，比 F_{t-1} 包含的信息少得多。因此，为了便于对 x_t 进行编码，将迫使这个独立模块尽可能从 $\hat{x}_{t-1}$ 中提取相关的时间上下文。新提取的时间上下文将传播到未来的帧中。这种基于刷新的调制机制有效地缓解了错误传播问题。如图 6 所示，与 DCVC-DC 相比，我们的 DCVC-FM 能以较低的比特率成本保持各帧的质量。

3.4. 实施情况

RGB 和 YUV 色彩空间的单一模型。尽管 DCVC-DC 支持 RGB 和 YUV 两种色彩空间的单一网络结构，但仍分别训

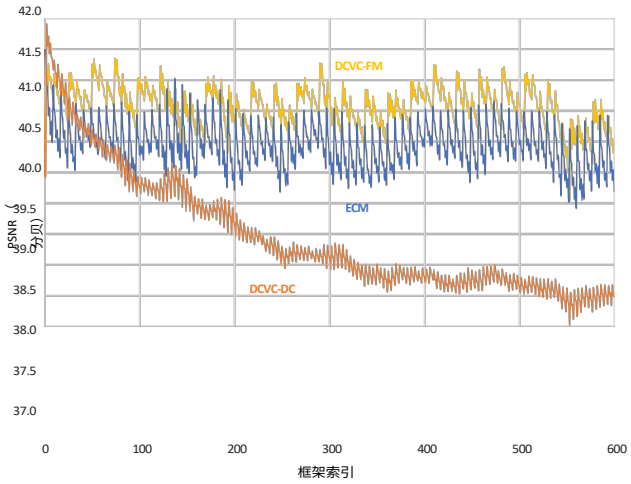


图 6.各帧质量比较。测试视频是来自 HEVC E 数据集（视频会议场景）的 *KristenAndSara*。DCVC-DC、ECM 和提议的 DCVC-FM 的平均 bpp（每像素比特）结果分别为 0.0037、0.0029 和 0.0026。周期内 =-1。

根据 NVC 的多功能性，我们直接为 RGB 和 YUV 训练一个模型，而无需额外的微调。为了支持这一点，我们的训练损耗涵盖了两种色彩空间，具体如下 $Loss_{(RD)\ R}$= + λ-(k-$D_{Y\ UV}$ +(1− k)-D_{RGB}).$D_{YUV)}$和 D_{RGB}分别是 YUV 和 RGB 的失真度。k 是用于加权的超参数，在实现中设置为 0.8。为了对 RGB 和 YUV 使用相同的输入接口，如果输入的是 YUV420 内容，UV 内容将向上采样。相应地，在获得重建帧后，UV 内容将被降采样。

低精度推断。大多数 NVC 只报告基于 32 位浮点实现的结果。我们希望使用 16 位来加速 NVC。然而，为了支持 16 位，我们需要改进*网格采样*函数的实现，该函数被广泛用于运动配准。我们的 DCVC-FM 是用 PyTorch 实现的。在 PyTorch [2]中，*网格样本*中的 "网格 "参数表示显示帧中的绝对位置。使用 16 位

练不同的模型权重。为了改进

精度，只有 10 位符号不足以表示
[3].为了解决这个问题，我们使用 16 位精度来表示相对偏移量，并重新实现*网格采样*。这样就能以 16 位精度进行 NVC 推理，大大节省了内存和复杂度，而压缩比的变化可以忽略不计。

为了减少计算量，我们在 DCVC-DC 的基础上又做了两项改进。一个是减少运动估计模块高分辨率特征的卷积核大小。另一项改进是，我们使用了更多的深度可分离卷积，这可以降低计算成本，同时避免过度拟合[13]。第 4.3 节验证了这两项改进在比特率增加较少的情况下带来了非同一般的 MAC 降低。

[2] 目前达到 PyTorch-2.1 版本。
[3])https://en.wikipedia.org/wiki/Half-precision 浮点格式

表 1.RGB 色彩空间下的 BD-Rate (%) 对比。96 帧，周期内=32。

	UVG	MCL-JCV	HEVC B	HEVC C	HEVC D	HEVC E	平均
VTM-17.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HM-16.25	38.4	45.6	40.3	37.9	32.4	41.0	39.3
ECM-5.0	-10.6	-13.2	-11.5	-12.6	-11.2	-9.8	-11.5
CANF-VC	73.0	70.8	64.3	76.2	63.1	120.5	78.0
DCVC	166.1	121.5	123.0	143.0	98.2	272.9	154.1
DCVC-TCM	44.1	51.0	40.2	66.3	37.0	82.7	53.6
DCVC-HEM	1.1	8.6	5.1	22.2	2.4	20.5	10.0
DCVC-DC	-19.1	-11.3	-12.0	-10.3	-26.1	-18.0	-16.1
DCVC-FM	-17.0	-5.6	-14.3	-23.7	-36.7	-24.5	-20.3

注：有些数字与 [26] 中的数字略有不同，因为我们使用了更宽的质量范围来计算 BD-Rate。

表 2.YUV420 色彩空间下的 BD-Rate (%) 对比。96 帧，周期内=32。

	UVG	MCL-JCV	HEVC B	HEVC C	HEVC D	HEVC E	平均
VTM-17.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HM-16.25	38.0	45.9	39.3	34.6	29.0	37.5	37.4
ECM-5.0	-11.5	-15.0	-12.7	-13.7	-12.2	-11.0	-12.7
DCVC-DC	-17.8	-12.0	-10.8	-12.4	-28.5	-20.4	-17.0
DCVC-FM	-21.6	-11.4	-16.3	-25.8	-39.3	-30.1	-24.1

4. 实验结果

4.1. 实验设置

数据集。与大多数现有的 NVC 一样，我们使用 Vimeo-90k 数据集 [49] 进行训练。除了 Vimeo-90k 中现成的 7 帧视频外，我们还获取了原始 Vimeo 视频[4]，生成了额外的 6658 个视频，每个视频由 32 帧组成。首先使用 7 帧视频进行训练，然后使用 32 帧视频进行微调，从而进一步提高质量。测试时，我们使用常见的 HEVC B~ E [11]、UVG [38] 和 MCL-JCV [48] 数据集。

测试条件。所有测试均在低延迟编码设置下进行。我们采用 BD-Rate 指标[10]来衡量压缩率的变化，正值表示比特率提高，负值表示降低。视频质量采用 PSNR 进行评估。我们的基准包括传统编解码器 H.265/HM [2]、H.266/VTM [3] 和 ECM [1]。

在 RGB 色彩空间方面，为了便于与现有方法进行比较，我们沿用了 [26] 中的测试条件，对每段视频的 96 个帧进行测试，测试期间为 32。我们还将 DCVC-FM 与之前的 SOTA NVC 模型进行了比较，包括 CANF-VC [16]、DCVC [24]、DCVC- TCM [44]、DCVC-HEM [25] 和 DCVC-DC [26]。

对于 YUV420 色彩空间，我们最初按照 [26] 的方法测试了 96 帧，周期内 32 帧。如前所述、

而传统的标准委员会[11]严格规定周期内为-1。因此，我们主张在不设置周期内-1 的情况下测试 NVC。与 [26] 仅测试 96 帧不同，我们还测试了周期内为 -1 的所有帧。这对现有的 NVC 来说是一个相当大的挑战，但却是与传统编解码器进行比较时最公平的设置。我们注意到，所有编码工具和传统编解码器的参考结构都使用最佳设置来代表其最佳压缩比。我们测试了 ECM-5.0 和最新的 ECM-11.0。对于 NVC，大多数现有模型仅针对 RGB 色彩空间进行了优化，而 DCVC-DC [26] 是唯一发布了 YUV420 色彩空间模型的 NVC。BD-Rate 的计算基于三个色彩分量的加权 PSNR，权重为 (6, 1, 1) / 8，与标准委员会 [45] 一致。

4.2. 与以往 SOTA 方法的比较

RGB 色彩空间。表 1 显示了在 96 帧、周期内 32 帧的 RGB 色彩空间下的性能比较。从表中可以看出，我们的 DCVC-FM 压缩比最佳。DCVC-FM 比 VTM 平均节省了 20.3% 的比特率，比 ECM 也有显著优势。此外，我们的 DCVC-FM 也优于 DCVC-DC，因为 DCVC-DC 比 VTM 节省了 16.1%的比特率。在这种下，DCVC-DC 仍然优于 VTM 并不是一件小事。

表 3.YUV420 色彩空间中的 BD-Rate (%) 对比。96 帧，周期内= -1.

	UVG	MCL-JCV	HEVC B	HEVC C	HEVC D	HEVC E	平均
VTM-17.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HM-16.25	39.5	48.3	41.5	40.3	32.6	41.5	40.6
ECM-5.0	-13.3	-16.4	-14.6	-15.6	-14.1	-12.6	-14.4
DCVC-DC	-13.6	-7.9	-7.8	-5.6	-27.6	-10.3	-12.2
DCVC-FM	-25.4	-11.6	-17.1	-24.4	-41.5	-31.6	-25.3

表 4.**YUV420 色彩空间中的 BD-Rate (%) 对比。所有帧的周期内= -1.**

	UVG	MCL-JCV	HEVC B	HEVC C	HEVC D	HEVC E	平均
VTM-17.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HM-16.25	40.1	48.6	47.6	41.0	34.5	42.8	42.4
ECM-5.0	-14.9	-17.0	-17.3	-16.6	-16.1	-14.1	-16.0
ECM-11.0	-20.0	-22.1	-22.2	-21.2	-20.4	-17.2	-20.5
DCVC-DC	5.7	-5.0	12.2	-4.2	-16.5	84.4	12.8
DCVC-FM	**-20.7**	**-10.3**	**-18.2**	**-32.2**	**-41.2**	**-30.2**	**-25.5**

与 DCVC-DC 相比，我们的编解码器质量范围更广，在单一模型中同时支持 RGB 和 YUV。

YUV420 色彩空间。表 2 显示了 YUV420 色彩空间在 96 帧（周期内 32 帧）下的对比情况。如表所示，我们的 DCVC-FM 可以超越所有传统编解码器和 DCVC-DC。与 VTM 相比，我们的比特率平均节省了 24.1%，而 DCVC-DC 则节省了 17.0%。表 1 和表 2 显示了 DCVC-DC 在 RGB 和 YUV420 色彩空间方面的持续改进。

不过，我们更侧重于使用周期内 -1 设置进行评估。表 3 显示了在 96 帧、周期内 -1 的情况下进行的比较。从表中可以看出，我们的 DCVC-FM 可以保持低的比特率、
相比之下，当周期内 32 变为-1 时，DCVC-DC 比 VTM 节省的比特率为 24.1%，而 DCVC-DC 比 VTM 节省的比特率为 25.3%。相比之下，当周期内从 32 变为 -1 时，DCVC-DC 比 VTM 节省的比特率从 17.0% 降为 12.2%。

但只测试 96 帧还不足以评估 NVC 在处理长预测链时的行为。因此，我们还测试了每个测试视频的所有帧。表 4 显示了相应的比较结果。从表中可以看出，我们的 DCVC-FM 性能仍然比 VTM 高出 25.5%。相比之下，在测试所有帧时，DCVC-DC 的性能下降幅度较大，即比 VTM 提高了 12.8%。特别是在 HEVC E 数据集上，DCVC-DC 的比特率下降了 84.8%。相比之下，我们的编解码器能很好地处理这一数据集。此外，如果使用 DCVC-DC 作为表 4 中的锚点，我们的 DCVC-FM 可以在所有数据集上平均节省 29.7% 的比特率。据我们所知，我们的 DCVC-FM 是第一个能达到如此高比特率的 NVC。

期内-1 设置下的压缩率。

图 7 显示了速率-失真曲线。从这些曲线可以看出，我们的 DCVC-FM 比 DCVC-DC 实现了更宽的质量范围。例如，在 MCL-JCV 数据集上，DCVC-DC 的质量范围为 3.7 dB（[38.66, 42.32]），无法满足更高的信噪比要求。相比之下，DCVC-FM 为 11.4 dB（[31.40, 42.80]），范围更广。表 5 显示了更多的比较结果，我们可以看到每个测试数据集的质量范围都有明显的扩大。拟议模型的质量范围与使用 QP 25~ 49 进行 VTM 编码的质量范围相似。

4.3. 消融研究

表 6 显示了对每种改进的消融研究。表中的基准模型 M_a 为 DCVC-DC。我们首先测试了减少运动估计模块和使用更多深度卷积的结构优化。如表 8 所示，这种优化（$M_{(b)}$）比特率增加了 0.8%，但比 DCVC-DC（$M_{(a)}$）节省了 16% 的 MAC。表 6 还显示，如果进一步支持更宽的质量范围（M_c），则比特率会降低 3.4%。如果同时支持 RGB 和 YUV 色彩空间的单一模型（M_d），则比特率会增加到 4.8%，因为这些功能并不是免费的。使用较长的视频进行训练可以大幅提高性能，M_e 可以节省 15.5% 的比特率，这说明了利用较长的时间相关性的好处。在 M_e 的基础上，刷新时间特征（M_f）可有效解决质量下降问题，从而使比特率节省率提高到 29.7%。

表 8 还显示了运行时间比较。我们的 MAC

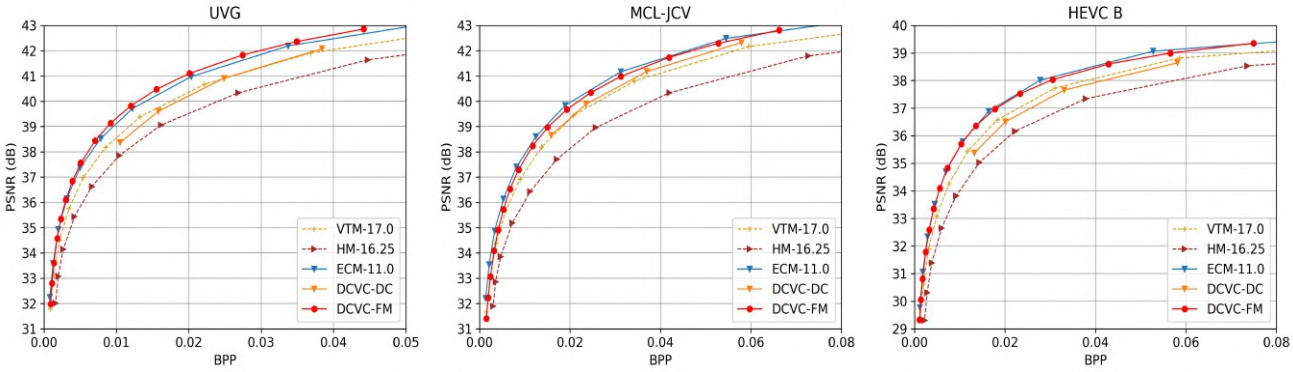


图 7：UVG、MCL-JCV 和 HEVC B 数据集的速率和失真曲线UVG、MCL-JCV 和 HEVC B 数据集的速率和失真曲线。比较采用 YUV420 色彩空间。所有帧的周期内 =-1。更多数据集的曲线见补充材料。

表 5.YUV420 色彩空间的质量范围（PSNR，dB）比较。所有帧的周期内= -1.

	UVG	MCL-JCV	HEVC B	HEVC C	HEVC D	HEVC E	平均
DCVC-DC	3.7	3.7	3.3	4.2	4.6	3.6	3.8
DCVC-FM	10.9	11.4	10.0	12.4	12.5	11.3	11.4

表 6.使用 BD-Rate 进行的消融研究（%）。

	M_a	M_b	M_c	M_d	M_e	M_f
结构优化		✓	✓	✓	✓	✓
更广泛的质量范围支持			✓	✓	✓	✓
单一型号用于 RGB&YUV				✓	✓	✓
培训与更长的视频					✓	✓
特点刷新						✓
BD 比率(%)	0.0	0.8	3.4	4.8	-15.5	-29.7

表 7.使用 16 位浮点（fp）推断的 BD 速率（%）。

fp32	未优化的 fp16	带优化的 fp16
0.0	87.3	0.9

表 8.复杂性比较。

	MAC	编码时间	解码
DCVC-DC w/ fp32	2642G	1005ms	765ms
DCVC-FM w/ fp32	2225G	1040ms	775ms
DCVC-FM w/ fp16	2225G	530ms	475ms

注：测试在 NVIDIA 2080TI 上进行，使用 1080p 作为输入。

但使用 32 位浮点推理的实际运行时间要稍长一些。这是因为我们使用了更多的深度卷积层，而深度卷积层的计算密度比普通卷积层低，但未来可以进一步加速[35]。当我们的*网格采样*操作化实现启用 16 位浮点推理时，运行时间显著缩短。特别是，如果使用 16 位推理，我们的 NVC 可以节省一半的内存使用量。表 7 还显示，如果不使用我们的优化实现，比特率将大幅提高 87.3%。

5. 结论和限制

总之，本文提出了特征调制技术，并解决了限制实际应用的两大难题。NVC 的逻辑性。通过均匀量化参数采样机制来帮助量化标度器的学习，我们可以对潜在特征进行精细调制，实现宽质量范围的支持。同时，我们还展示了速率控制能力。我们还通过周期性刷新机制对潜在特征进行调制，解决了预测链过长的问题。此外，我们的 DCVC-FM 现在还支持 RGB 和 YUV 色彩空间，并允许低精度推理。我们的 DCVC-FM 标志着无损检测技术的发展迈出了重要一步。

然而，尽管 DCVC-FM 可以实现低精度浮点推理，但其速度仍与实时性相去甚远。此外，浮点推理对于 NVC 中的熵编码还存在跨平台问题。今后，我们将对这些问题进行研究，并构建功能更强大的无损检测器，以广泛应用于实际产品中。

参考资料

[1] ECM. https://vcgit.hhi.fraunhofer.de/ecm/ECM.6, 11

[2] HM. https://vcgit.hhi.fraunhofer.de/jvet/HM/.6, 11

[3] VTM. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/.6, 11

[4] 原始 vimeo 链接。 https://github.com/anchen1011/toflow/blob/master/data/original_vimeo_links.txt.6

[5] Eirikur Agustsson、David Minnen、Nick Johnston、Johannes Balle、Sung Jin Hwang 和 George Toderici。用于端到端优化视频压缩的规模空间流。In *Pro- ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8503-8512, 2020.1, 2

[6] E.Alshina, J. Ascenso, T. Ebrahimi, F. Pereira, and T. Richter.[AHG 11] JPEG AI CfP 现状简介。*JVET-AA0047*, 2022.11

[7] 主播 - JPEG-AI MMSP 挑战赛。锚 - JPEG-AI MMSP 挑战赛。https://jpegai.github.io/7- anchors/。11

[8] Joa˜o Ascenso、Elena Alshina 和 Touradj Ebrahimi。JPEG AI 标准：提供高效的人机视觉数据消费。*IEEE Multimedia*，30（1）：100- 111，2023.2

[9] 约翰内斯-巴勒、戴维-明宁、绍拉布-辛格、黄成金和尼克-约翰斯顿。使用尺度超优先级的变量图像压缩。*第六届国际学习表示会议，ICLR*，2018。2

[10] Gisle Bjontegaard.计算 RD 曲线之间的平均 PSNR 差异 .*VCEG-M33*, 2001.6

[11] Frank Bossen 等人. 通用测试条件和软件参考配置。见 *JCTVC-L1100*，2013 年。2, 6

[12] Benjamin Bross、Ye-Kui Wang、Yan Ye、Shan Liu、Jianle Chen、Gary J Sullivan 和 Jens-Rainer Ohm。通用视频编码（VVC）标准及其应用概述。*IEEE 视频电路与系统论文 Technology*, 31(10):3736-3764, 2021.1

[13] Franc¸ois Chollet.Xception：深度可分离卷积深度学习。In *Proceedings of the IEEE con- ference on computer vision and pattern recognition*, pages 1251-1258, 2017.5

[14] Abdelaziz Djelouah、Joaquim Campos、Simone Schaub- Meyer 和 Christopher Schroers。用于视频编码的神经帧间通信。2019 年*IEEE/CVF计算机视觉国际会议（ICCV）论文集*。2

[15] Noor Fathima Goose, Jens Petersen, Auke Wiggers, Tianlin Xu, and Guillaume Sautiere.基于扩散的解码器的神经图像压缩。*ArXiv 预印本 arXiv:2301.05489*, 2023.2

[16] Yung-Han Ho、Chih-Peng Chang、Peng-Yu Chen、Alessan- dro Gnutti 和 Wen-Hsiao Peng。Canf-vc：用于视频压缩的条件增强归一化流。*欧洲计算机视觉会议*，2022 年。2, 6

[17] Zhihao Hu, Zhenghao Chen, Dong Xu, Guo Lu, Wanli Ouyang, and Shuhang Gu.通过分辨率自适应流编码改进深度视频压缩。*欧洲计算机视觉大会》*，第 193-209 页。Springer, 2020.2

[18] Zhihao Hu, Guo Lu, and Dong Xu.FVC：实现特征空间深度视频压缩的新框架。In *Pro- ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1502-1511, 2021.2

[19] Zhihao Hu, Guo Lu, Jinyang Guo, Shan Liu, Wei Jiang, and Dong Xu.超优先模式预测的粗到细深度视频编码。*IEEE/CVF 计算机视觉与模式识别会议论文集》*，第 5921-5930 页，2022 年。1, 2

[20] Cong Huang、Jiahao Li、Bin Li、Dong Liu 和 Yan Lu。基于神经压缩的视频还原特征学习。在 *IEEE/CVF 计算机视觉与模式识别会议（CVPR）论文集上*，第 5872-5881 页，2022。4

[21] A Burakhan Koyuncu、Han Gao 和 Eckehard Steinbach。Contextformer：用于学习图像压缩中上下文建模的空间通道注意变换器。*arXiv preprint arXiv:2203.02452*, 2022.2

[22] The´o Ladune, Pierrick Philippe, Wassim Hamidouche, Lu Zhang, and Olivier De´forges.基于学习的视频编码光流和模式选择。*第 22 届 IEEE 多媒体信号处理国家间研讨会*，2020 年。1

[23] The´o Ladune、Pierrick Philippe、Wassim Hamidouche、Lu Zhang 和 Olivier De´forges.用于灵活学习视频压缩的条件编码。*神经压缩：从信息论到应用--ICLR研讨会*，2021。2

[24] Jiahao Li、Bin Li 和 Yan Lu.深度上下文视频通信。*神经信息处理系统进展》（Advances in Neural Information Processing Sys- tems）*，34，2021。1, 3, 6

[25] Jiahao Li, Bin Li, and Yan Lu.用于神经视频压缩的混合时空建模。In *Proceed-ings of the 30th ACM International Conference on Multime- dia*, pages 1503-1511, 2022.2, 3, 4, 6

[26] Jiahao Li、Bin Li 和 Yan Lu.不同语境下的神经视频压缩。In *IEEE/CVF Conference on Com- puter Vision and Pattern Recognition, CVPR 2023, Vancou- ver, Canada, June 18-22, 2023, 2023*.1, 2, 3, 6, 11, 12

[27] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu.M-LVC：用于学习视频压缩的多帧预测。2020年*IEEE/CVF计算机视觉与模式识别会议论文集（puter Vision and Pattern Recognition）*。1, 2

[28] Bowen Liu、Yu Chen、Rakesh Chowdary Machineni、Shiyu Liu 和 Hun-Seok Kim。Mmvc：基于块预测模式选择和密度自适应熵编码的学习型多模式视频压缩。*IEEE/CVF 计算机视觉与模式识别会议论文集》*，第 18487-18496 页，2023 年。1, 2

[29] 刘豪杰、卢明、马湛、王帆、谢志煌、曹勋和王尧。使用多尺度运动补偿和时空上下文的神经视频编码

模型。*IEEE 视频电路与系统 技术论文集*，2020 年。2

[30] Jerry Liu、Shenlong Wang、Wei-Chiu Ma、Meet Shah、Rui Hu 、Pranaab Dhawan 和 Raquel Urtasun。高效视频压缩的条件编码。*欧洲计算机视觉会议*，第 453-468 页。Springer, 2020.2

[31] Jinming Liu、Heming Sun 和 Jiro Katto。使用混合变压器-神经网络架构的学习图像压缩。In *Proceedings of the IEEE/CVF Conference on Computer Vi- sion and Pattern Recognition*, pages 14388-14397, 2023.2

[32] Guo Lu、Wanli Ouyang、Dong Xu、Xiaoyun Zhang、Chunlei Cai 和 Zhiyong Gao。DVC：端到端深度视频通信框架。In *Proceedings of the IEEE/CVF Con- ference on Computer Vision and Pattern Recognition*, pages 11006-11015, 2019.1, 2

[33] Guo Lu 、Chunlei Cai 、Xiaoyun Zhang 、Li Chen 、Wanli Ouyang、Dong Xu 和 Zhiyong Gao。内容自适应和误差传播感知深度视频压缩。*欧洲计算机视觉会议*》，第 456-472 页。Springer, 2020.

[34] Guo Lu、Xiaoyun Zhang、Wanli Ouyang、Li Chen、Zhiyong Gao 和 Dong Xu。用于视频压缩的端到端学习框架。*IEEE patterns analysis and machine intelligence*, 2020.2

[35] Gangzhao Lu, Weizhe Zhang, and Zheng Wang.优化 GPU 上的深度可分离卷积操作。*IEEE 并行与分布式系统论文集*，33（1）：70- 87，2021。8

[36] Wufei Ma、Jiahao Li、Bin Li 和 Yan Lu.不确定性感知的集合深度视频压缩。*IEEE Transac- tions on Multimedia*, 2024.2

[37] Fabian Mentzer 、George Toderici 、David Minnen 、Sung-Jin Hwang、Sergi Caelles、Mario Lucic 和 Eirikur Agusts- son。Vct：视频压缩转换器。*ArXiv 预印本 arXiv:2206.07307*, 2022.2

[38] Alexandre Mercat、Marko Viitanen 和 Jarno Vanne。UVG 数据集：用于视频编解码器分析和开发的 50/120fps 4k 序列。*第 11 届 ACM 多媒体 系统会议论文集*》，第 297-302 页，2020 年。6

[39] Zhihong Pan, Xin Zhou, and Hao Tian.通过从差异模型学习文本嵌入实现极端生成式图像压缩》（Extreme generative image compression by learning text embedding from diffu- sion models. *arXiv preprint arXiv:2211.07793*, 2022.2

[40] Linfeng Qi, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu.用于神经视频压缩的移动信息传播。*IEEE/CVF 计算机视觉与模式识别会议论文集*》，第 6111-6120 页，2023 年。2, 3

[41] 钱一辰、林明、孙秀玉、谭志宇和金蓉。Entroformer：基于变换器的学习图像压缩熵模型。*ArXiv 预印本 arXiv:2202.05492*, 2022.2

[42] Oren Rippel、Alexander G Anderson、Kedar Tatwawadi、San- jay Nair、Craig Lytle 和 Lubomir Bourdev。ELF-VC：有效学习的灵活速率视频编码。*IEEE/CVF 计算机视觉国际会议（ICCV）论文集*》，第 14479-14488 页，2021 年。2

[43] Vadim Seregin、Jie Chen、Roman Chernyak、Fabrice Lean- nec 和 Kai Zhang。JVET AHG 报告：ECM 软件开发（AHG6）。*JVET-AF0006*, 2023。1

[44] Xihua Sheng、Jiahao 、Bin Li、Li Li、Dong Liu 和 Yan Lu.用于学习视频压缩的时间上下文挖掘 *电气和电子工程师学会多媒体期刊*，2022 年。2, 3, 6

[45] Gary J Sullivan 和 Jens-Rainer Ohm.视频编码联合协作组（JCT-VC）第四次会议报告，韩国大邱，2011 年 1 月 20-28 日。*文件 JCTVC-D500，韩国大邱*，2011 年。6

[46] Lucas Theis, Tim Salimans, Matthew D Hoffman, and Fabian Mentzer. 高斯扩散的有损压缩。*arXiv preprint arXiv:2206.08889*, 2022.2

[47] Guo-Hua Wang, Jiahao Li, Bin Li, and Yan Lu.EVC：带掩码的实时神经图像压缩。*国际学习代表会议（International Conference on Learning Representa- tions）*，2023 年。2

[48] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavouni- dis, Anne Aaron, and C-C Jay Kuo.MCL-JCV：基于 JND H.264/AVC 视频质量评估数据集。In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1509-1513.IEEE, 2016.6

[49] 薛天帆、陈百安、吴佳俊、魏东来、William T Freeman。面向任务流的视频增强*国际计算机视觉杂志》（IJCV）*，127 (8):1106-1125, 2019.6

[50] Ruihan Yang and Stephan Mandt.使用条件扩散模型的有损图像压缩。*ArXiv 预印本 arXiv:2209.06950*, 2022.2

[51] 杨一博和斯蒂芬-曼特使用浅层解码器的计算高效神经图像压缩。*计算机视觉国际会议论文集*》，第 530-540 页，2023 年。2

[52] 邹仁杰、宋春峰、张兆祥。细节决定成败：基于窗口的图像压缩注意力。*IEEE/CVF 计算机视觉与模式识别会议论文集*》，第 17492- 17501 页，2022 年。2

附录

本文件为我们提出的神经视频编解码器（NVC）（即DCVC-FM）提供了补充材料。

A. 测试设置

为了进行全面的比较分析，我们在 YUV420 和 RGB 两种颜色空间中对 NVC 和传统编解码器进行了比较。

YUV420 色彩空间。大多数传统编解码器和实际应用主要采用 YUV420 色彩空间作为输入和输出，并在此色彩空间中进行优化。因此，在 YUV420 色彩空间中对比 NVC 和传统编解码器对于评估 NVC 的发展进程相当重要。对于传统编解码器，测试了 HM [2]、VTM [3] 和 ECM [1]，其中 HM、VTM 和 ECM 分别是 H.265、H.266 和正在开发的下一代传统编解码器的参考软件。三种传统*编解码器*分别使用了*编码器低延时 main10.cfg*、*编码器低延时 vtm.cfg* 和*编码器低延时 ecm.cfg* 配置文件。每个视频的参数如下

- -c *{配置文件名}*
  --输入文件=*{输入视频名称}*.
  --输入位深度=8
  --输出位深度=8
  --输出位深度C=8
  --*FrameRate*=*{帧频}*
  --DecodingRefreshType=2解码刷新类型
  --*FramesToBeEncoded*=*{*帧*数}*.
  --*SourceWidth*={width}（源宽度
  --*SourceHeight*={height}源高度
  --内期=*{内期}*。
  --*QP*={qp}。
  --级别=6.2
  --*BitstreamFile*=*{*比特流*文件名}*.

RGB 色彩空间。由于所有测试集的原始格式都是 YUV420 色彩空间。因此，要测试 RGB 视频，我们需要将它们从 YUV420 转换到 RGB 色彩空间。我们遵循 JPEG AI [6, 7] 和 [26]，使用 BT.709 将原始 YUV420 视频转换为 RGB 视频。这是因为与常用的 BT.601 相比，在视觉质量相似的情况下，使用 BT.709 可以获得更高的压缩比。文献[26]指出，当传统编解码器测试 RGB 视频时，使用 10 位 YUV444 作为内部色彩空间会比直接使用 RGB 获得更好的压缩比，尽管最终的失真度是以 RGB 测量的。因此，我们也采用了这一设置。对于 HM、VTM 和 ECM，分别使用了*编码器低延时主 rext.cfg*、*编码器低延时 vtm.cfg* 和*编码器低延时 ecm.cfg* 配置文件。每个视频的参数如下

表 9.使用不同特征刷新周期的 BD-Rate (%)。

刷新周期	0	8	16	32	64	96
BD 比率	0	-7.2	-17.1	-20.0	-17.9	-14.8

- -c *{配置文件名}*

  --*输入文件*={输入*文件名}*.

  --输入位深度=10

  --输出位深度=10

  --输出位深度C=10

  --输入色度格式=444

  --*FrameRate*={*帧频}*

  --DecodingRefreshType=2解码刷新类型

  --*FramesToBeEncoded*={帧*数}*.

  --*SourceWidth*={width}（源宽度

  --*SourceHeight*={height}源高度

  --内期={*内期}*。

  --*QP*={qp}。

  --级别=6.2

  --*BitstreamFile*={比特流*文件名}*.

值得注意的是，对于 YUV420 和 RGB 色彩空间，传统编解码器的所有编码工具和参考结构都使用其最佳设置来表示其最佳压缩比。

B. 时间特征调制

在本文中，我们专门设计了特征刷新机制来调节时间特征。这将提高特征传播的有效性。默认刷新周期设置为 32。这里我们测试了不同的刷新周期设置，相应的比较结果如表 9 所示，其中刷新周期为 0 表示禁用特征刷新。从表中我们可以看出，随着刷新周期的增大，双态节省最初会增加。当刷新周期大于 32（即 64 和 96）时，性能开始下降。这一现象说明，刷新周期太小或太大都不合适。如果刷新周期过小，增加的比特率成本就会过高，因为具有刷新特征的帧不仅质量更好，而且比特率成本也会更高。相反，过大的周期尺寸会导致传播的特征被累积的错误所污染，或包含更多不相关的信息。因此，目前刷新周期 32 是一个很好的权衡选择。不过，需要注意的是，不同视频的最佳刷新周期大小可能会有所不同。采用内容自适应方法来确定刷新周期大小可能会产生更好的效果。我们将在未来对此进行研究。

C. 费率控制

我们采用了一种简单的速率控制算法，通过调整模型中的量化参数 q_t 值来证明其可行性。由于比特率在帧间波动，我们只在偶数帧调整量化参数 q_t。速率控制算法算法 1。需要注意的是，该算法只是一种简单的实现方法，用于演示速率控制的可行性。在我们的编解码器基础上，还可以开发出更先进的速率控制算法，我们将在未来对其进行研究。

D. 重新实施的*网格示例* –

重新实现的*网格采样*函数可在我们的发布代码中找到。我们还比较了随机生成的特征和运动向量的误差率。如果直接将 16 位张量值输入默认的*网格采样*函数，误差率高达 16.149% 。相比之下，如果使用我们重新实现并改进的*网格采样*函数，误差率仅为 0.026%。

E. 速率-失真曲线

在本文中，我们展示了所有数据集的速率-失真（RD）曲线，这些数据集测试了所有帧，并使用了pe-riod-1 设置。图 8 显示了 HEVC B、C 和 D 数据集的结果，图 9 显示了 HEVC E、UVG 和 MCL-JCV 数据集的结果。在这两幅图中，我们还分别比较了质量相对较低和较高的区域。从这些比较中我们可以看出，先前的 SOTA NVC DCVC-DC [26] 的质量范围相当有限。相比之下，我们的 DCVC-FM 的质量范围要宽泛得多，在许多情况下，它的 RD 性能比以前所有的编解码器都要好。

但如图 9 所示，在 MCL-JCV 数据集上，我们的 DCVC-FM 无法超越 ECM。MCL-JCV 包括屏幕内容视频。我们发现我们的编解码器在屏幕内容方面表现不佳。这是因为我们的训练数据集 Vimeo 是一个自然内容数据集。此外，我们的编解码器在噪点较多的视频中表现也较差。目前，神经编解码器很难对源视频中的随机噪声进行编码，因为噪声数据的概率很难准确预测。今后，我们将针对这些视频改进编解码器。

F. 单一模型中的平滑质量调整

为方便起见在比较 DCVC-FM 和其他编解码器时，我们测试了 16 个 RD 点。实际上，我们的 NVC 可以在一个模型中支持 64 种不同的质量水平。我们测试了所有这些 RD 点，如图 10 所示。从这些图中可以看出，我们的 DCVC-FM 在单一模型中可以实现非常平滑的质量调整，没有出现任何失真。

算法 1 速率控制算法

输入：

cbs：当前缓冲区大小（第一帧设置为 0） *tbs*：目标缓冲区大小（第一帧设置为 0） *q*：当前 q 值（第一帧设置为 32） *cfs*：当前帧大小
AFS：平均帧尺寸
fidx： 当前帧索引

输出

cbs： 更新下一帧的缓冲区大小
tbs： 更新的下一帧目标缓冲区大小
q：下一帧的更新 q 值

CBS
+=*CFS*
CBS
−=*afs*
if *fidx* mod 2== **1 then return**
cbs, *tbs*, *q*
如果结束
buff diff cbs= − *tbs*
tbs cbs= ∗ 0.95
if *buff diff*> 0 **then**
if *cbs*> 10∗ *cfs* **then**
q −=12
else if *cbs*> 5∗ *cfs* **then**
q −=6
else if *cbs*> 2∗ *cfs* **then**
q −=2
else if *buff diff*> 0.5∗ *cfs* & *cbs* >− *cfs* **then**
q −=1
如果结束
else if *buff diff*< **0 then**
if *cbs* <− 10∗ *cfs* **then**
q +=12
else if *cbs* <− 5∗ *cfs* **then**
q +=6
else if *cbs* <− 2∗ *cfs* **then**
q +=2
else if *buff diff* <− 0.5∗ *cfs* & *cbs*< *cfs* **then**
q +=1
如
果结束
如果结
束
q= *clip*(0, 63, *q*)
返回 *CBS*、*TBS*、*Q*

这也是实现精确速率控制的前提条件。这也是实现精确速率控制的前提条件。

G. 视觉对比

在本节中，我们将通过直观对比来说明 DCVC-FM 的卓越性能。图 11 展示了四个示例。这些示例表明

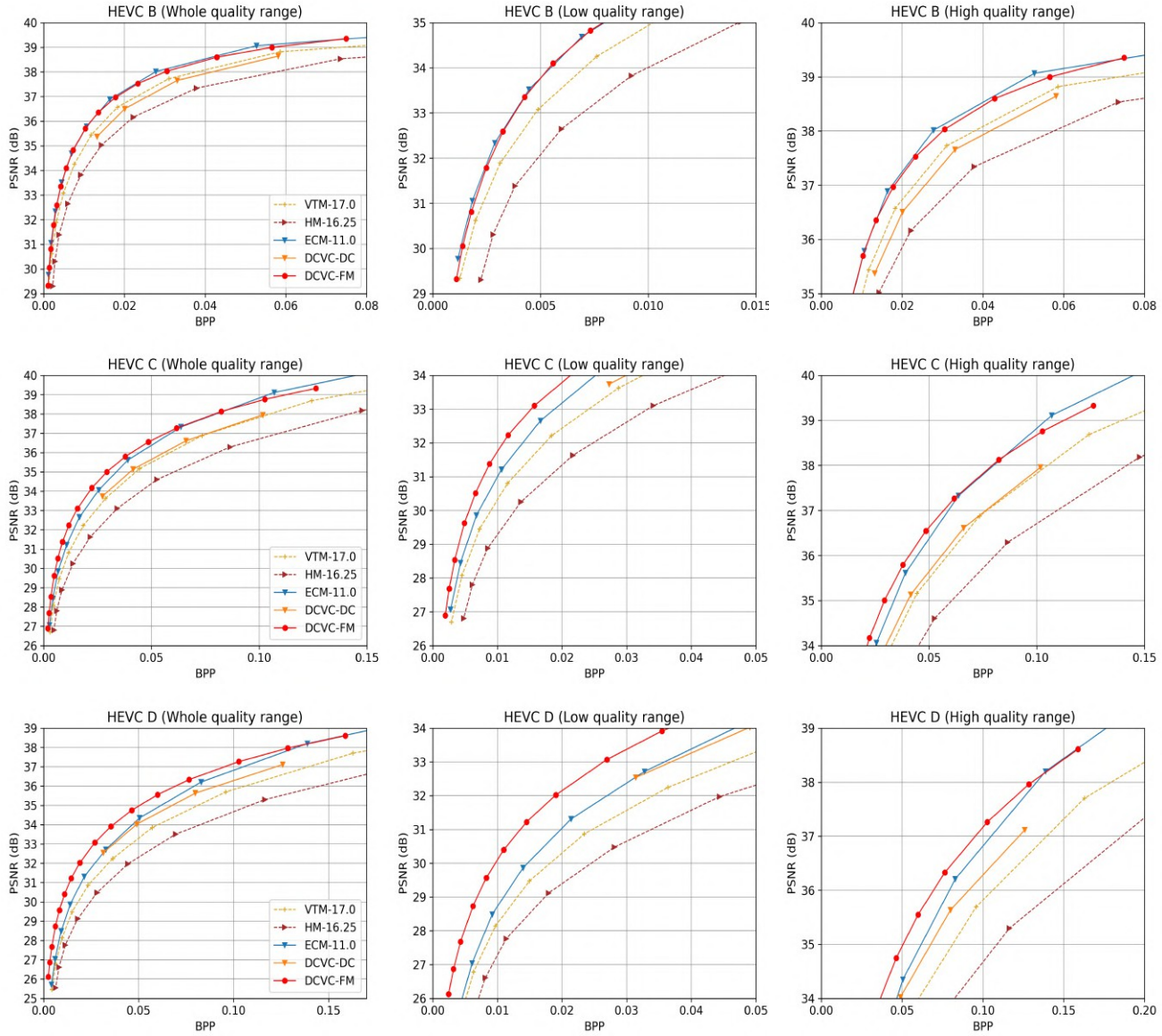


图 8：HEVC B、HEVC C 和 HEVC D 数据集的速率和失真曲线HEVC B、HEVC C 和 HEVC D 数据集的速率和失真曲线。每行显示一个数据集。从左到右分别为总体质量范围、相对较低质量范围和相对较高质量范围。比较采用 YUV420 色彩空间。所有帧内周期 =-1。

与传统编解码器 ECM 和以前的 SOTA NVC DCVC-DC 相比，我们的 DCVC-FM 能够以更高的清晰度重构纹理，而不会产生额外的比特率成本。

H. 使用 B 帧配置的传统编解码器

目前，我们的神经编解码器侧重于低延迟场景，因此传统编解码器使用低延迟-B（LDB）设置，以便在正文中进行公平比较。实际上，我们已经测试了 96 帧下 ECM-5.0 的分层-B（HieB）设置（随机访问配置，周期内= -1，低延迟要求被打破），如图所示见表 10。LDB 和 HieB 之间的压缩率差距约为 26%，与 JCTVC-K0279 报告数字一致（HEVC 平均为 21%）。设计一种在 HieB 环境中超越最佳传统编解码器的神经编解码器将是未来的工作。

表 10.BD-Rate (%) 对比（YUV420，96 帧，周期内= -1）。

	UVG	MCL-JCV	HEVC B	HEVC C	HEVC D	HEVC E	平均
VTM-17.0 (LDB)	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ECM-5.0 (LDB)	-13.3	-16.4	-14.6	-15.6	-14.1	-12.6	-14.4
ECM-5.0 (HieB)	-40.3	-40.5	-42.3	-39.1	-39.3	-39.4	-40.2
DCVC-FM （低延时）	-25.4	-11.6	-17.1	-24.4	-41.5	-31.6	-25.3

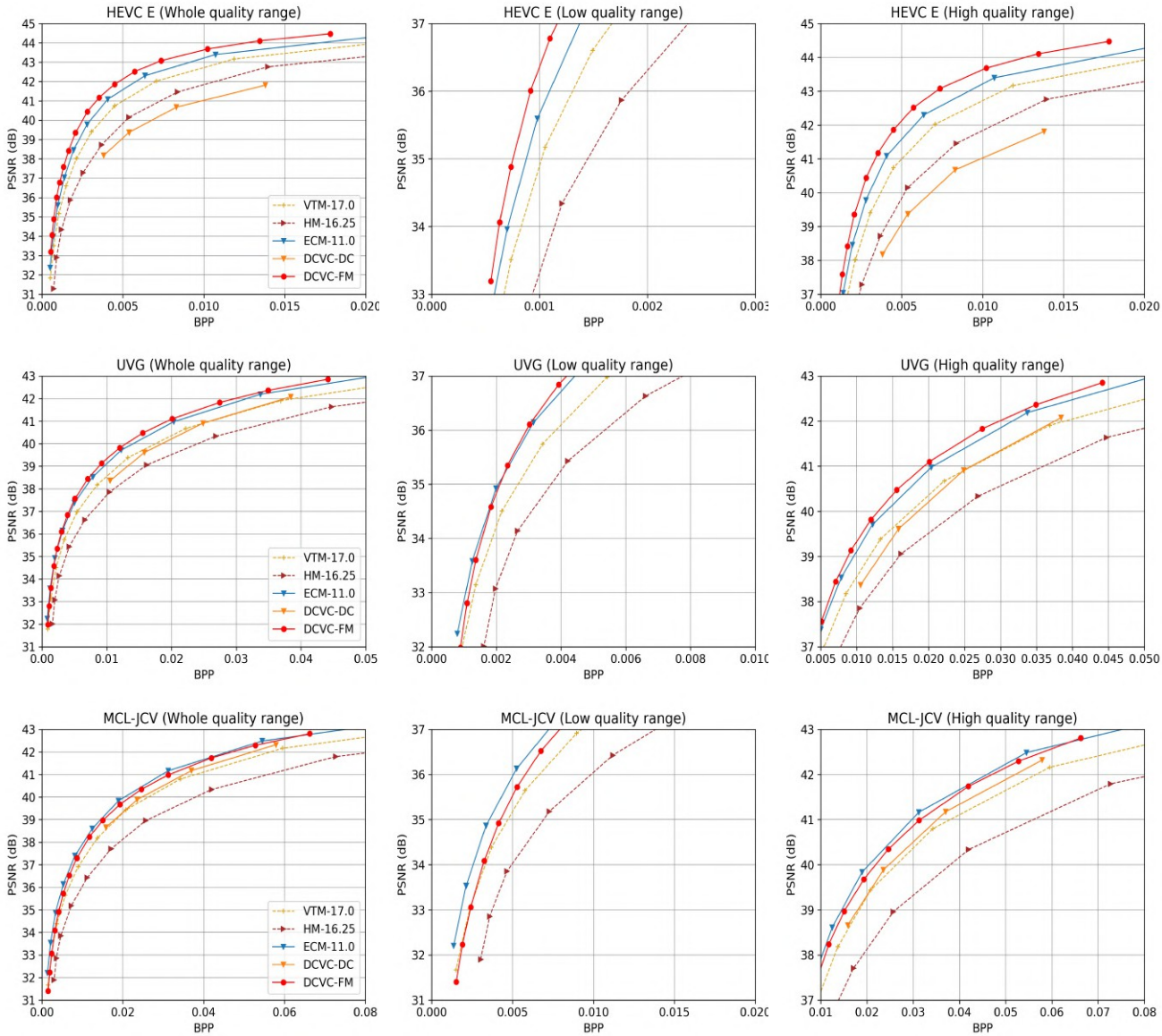


图 9.HEVC E、UVG 和 MCL-JCV 数据集的速率和失真曲线。每行显示一个数据集。从左到右分别为总体质量范围、相对较低质量范围和相对较高质量范围。比较采用 YUV420 色彩空间。所有帧的周期内 =-1。

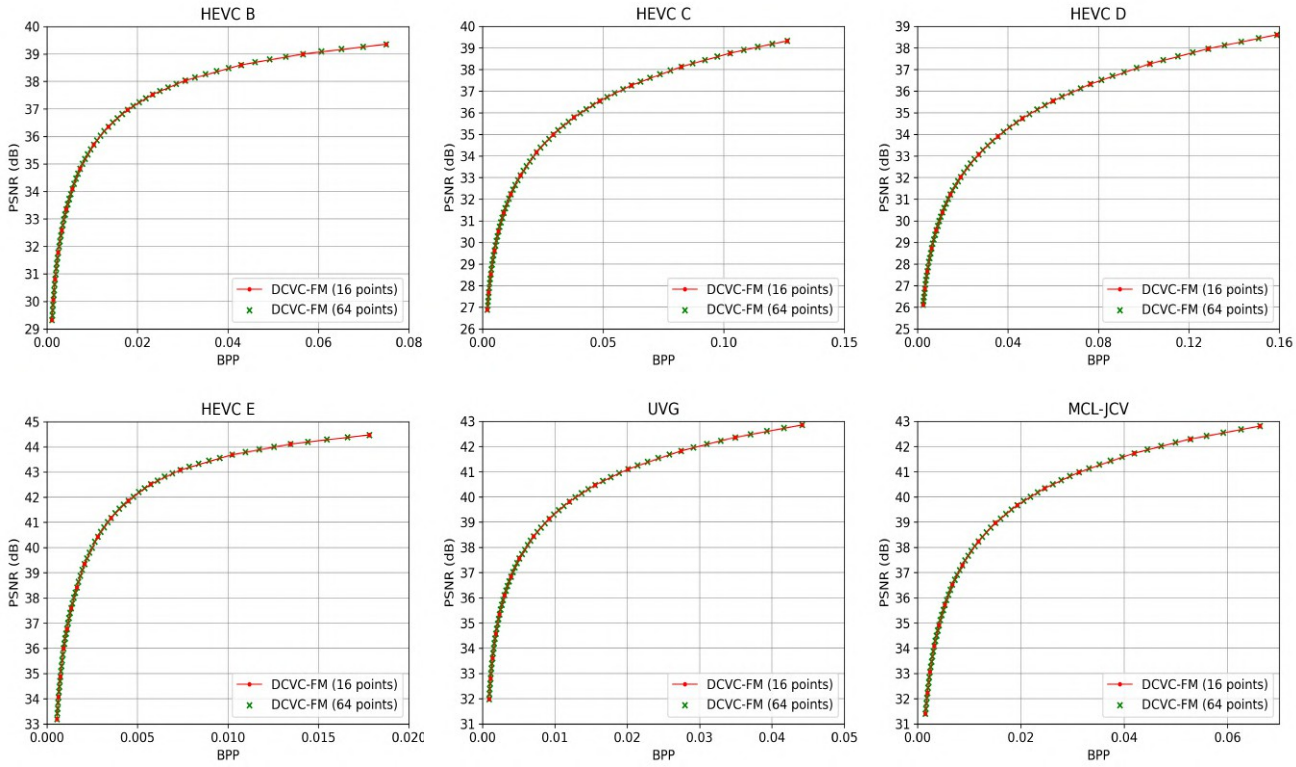


图 10.单一模型中的平滑质量调整我们的 DCVC-FM 支持 64 种不同的质量等级。

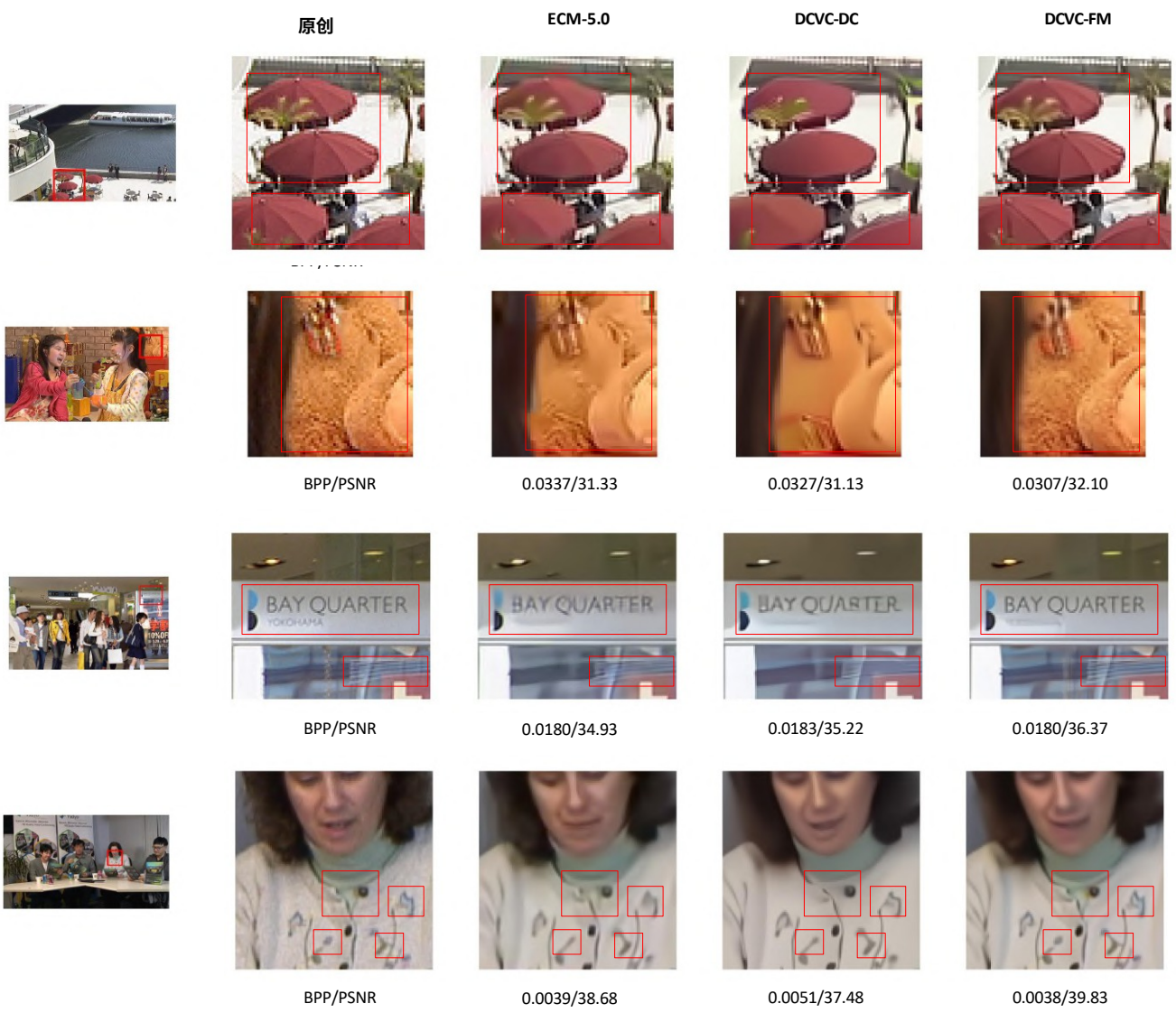

原创
ECM-5.0
DCVC-DC
DCVC-FM
BPP/PSNR
0.0337/31.33
0.0327/31.13
0.0307/32.10
BAY QUARTER
YOKOHAMA
BPP/PSNR
0.0180/34.93
0.0183/35.22
0.0180/36.37
BPP/PSNR
0.0039/38.68
0.0051/37.48
0.0038/39.83

图 11.视觉对比。