深度语境视频压缩

李嘉豪、李斌、卢艳
微软亚洲研究院
{li.jiahao, libin, yanlu}@microsoft.com

摘要

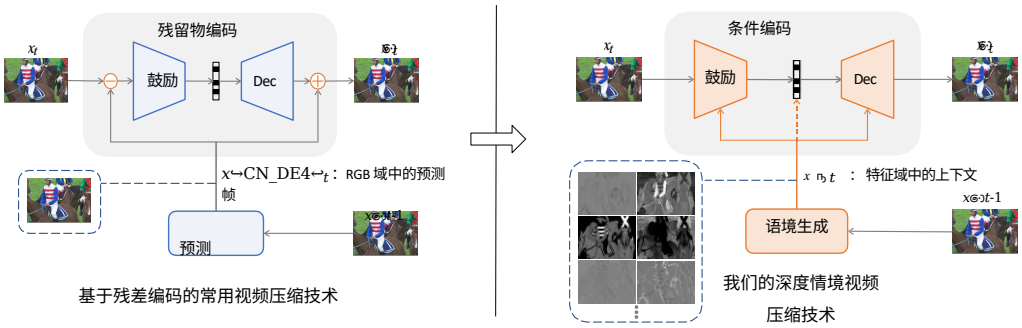
现有的神经视频压缩方法大多采用预测编码框架，首先生成预测帧，然后将其残差与当前帧进行编码。然而，就压缩率而言，预测编码只是一种次优解决方案，因为它使用简单的减法操作来消除各帧之间的冗余。在本文中，我们提出了一种深度上下文视频压缩框架，以实现从预测编码到条件编码的范式转变。特别是，我们试图回答以下问题：如何在深度视频压缩框架下定义、使用和学习条件。为了挖掘条件编码的潜力，我们建议使用特征域上下文作为条件。这使我们能够利用高维上下文为编码器和解码器提供丰富的信息，从而帮助重构高频内容，获得更高的视频质量。我们的框架还具有可扩展性，可以灵活设计条件。实验表明，我们的方法明显优于之前最先进的（SOTA）深度视频压缩方法。与使用 *veryslow* 预设值的 x265 相比，我们可以为 1080P 标准测试视频节省 26.0% 的比特率。代码见 https://github.com/DeepMC-DCVC/DCVC。

1 导言

从 1988 年开发的 H.261 [1] 到 2020 年刚刚发布的 H.266 [2]，所有传统视频编码标准都基于预测编码范式，即首先由手工制作的模块生成预测帧，然后对当前帧和预测帧之间的残差进行编码和解码。最近，许多基于深度学习（DL）的视频压缩方法[3-11]也采用预测编码框架对残差进行编码，其中所有手工制作的模块只是由神经网络代替。

考虑到帧与帧之间强烈的时间相关性，残差编码是一种简单而高效的视频压缩方式。然而，残差编码并不是在预测帧 $\tilde{x}_t$ 的情况下对当前帧 x_t 进行编码的最佳方法，因为它只使用手工减法运算来消除各帧之间的冗余。残差编码的熵大于或等于条件编码的熵[12]：$H(x_t - \tilde{x}_t) \geq H(x_t | \tilde{x}_t)$，其中 H 代表香农熵。从理论上讲，帧 x_t 中的一个像素与之前已解码帧中的所有像素以及 x_t 中已解码的像素相关。对于传统的视频编解码器，由于空间巨大，不可能使用手工制定的规则来明确地探索所有像素的相关性。因此，残差编码作为条件编码的一种极其简化的特殊情况而被广泛采用，它有一个很强的假设，即当前像素只与预测像素有相关性。DL 打开了在巨大空间中自动探索相关性的大门。考虑到 DL 在图像压缩方面的成功[13, 14]，它只是使用自动编码器来探索图像中的相关性，为什么不使用网络来构建基于条件编码的自动编码器来探索视频中的相关性，而不是将我们的视野局限在残差编码上呢？

第 35 届神经信息处理系统大会（NeurIPS 2021）。



x_t 是当前帧。$\hat{x}_t$ 和 $\hat{x}_{t-1}$ 是当前和之前的解码帧。橙色虚线表示上下文也用于熵建模。

当我们设计基于条件编码的解决方案时，自然会产生一系列问题：*什么是条件？如何使用条件？如何学习条件？* 从技术上讲，条件可以是任何有助于压缩当前帧的东西。预测帧可以用作条件，但没有必要将其限制为条件的唯一表示。因此，我们将条件定义为具有任意维度的可学习上下文特征。根据这一想法，我们提出了一种深度上下文视频压缩（DCVC）框架，以一种统一、简单而高效的方法利用条件。我们的 DCVC 框架示意图如图 1 所示。上下文信息被用作上下文编码器、上下文解码器以及熵模型输入的一部分。特别是，受益于上下文提供的时间先验信息，熵模型本身具有时间自适应能力，从而产生了更丰富、更准确的模型。至于如何学习条件，我们建议在特征域使用运动估计和运动补偿（MEMC）。MEMC 可以引导模型提取有用的上下文。实验结果证明了所提出的 DCVC 的有效性。对于 1080p 标准测试视频，我们的 DCVC 在使用 *veryslow* 预设值时比 x265 节省了 26.0% 的码率，比之前基于 SOTA DL 的模型 DVCPro [4] 节省了 16.4% 的码率。

事实上，条件编码的概念已在文献 [15, 16, 12, 17] 中出现。不过，这些作品只针对部分模块（如只针对熵模型或编码器）设计，或需要手工操作来筛选哪些内容应进行条件编码。相比之下，我们的框架是一个更全面的解决方案，它考虑了所有编码、解码和熵模型。此外，本文提出的 DCVC 是一种基于条件编码的可扩展框架，可以灵活设计条件。虽然本文提出使用特征域 MEMC 生成上下文特征并证明了其有效性，但我们认为这仍是一个值得进一步研究的开放性问题，以获得更高的压缩比。

我们的主要贡献有四个方面：

- 我们设计了一种基于条件编码的深度上下文视频压缩框架。条件的定义、使用和学习方式都是创新性的。与之前基于残差编码的方法相比，我们的方法可以实现更高的压缩率。
- 我们提出了一种简单而高效的方法，利用上下文帮助编码、解码和熵建模。在熵建模方面，我们设计了一种利用空间-时间相关性的模型，以获得更高的压缩率，或者只利用时间相关性来获得更快的速度。
- 我们将条件定义为特征域中的上下文。维度更高的上下文可以提供更丰富的信息，帮助重建高频内容。
- 我们的框架具有可扩展性。通过更好地定义、使用和学习条件，在提高压缩率方面存在巨大潜力。

2 相关作品

深度图像压缩 最近有许多关于深度图像压缩的研究。例如，压缩自动编码器 [18] 可以获得与 JPEG 2000 相当的效果。随后，许多工作通过更先进的熵模型和网络结构提高了性能。例如，Ballé 等人提出了因式[19]和超先验[13]熵模型。基于超先验的方法赶上了 H.265 内部编码。联合使用超先验和自动回归上下文的熵模型优于 H.265 内部编码。采用高斯混合模型[20]的方法与 H.266 内部编码不相上下。在网络结构方面，早期提出了一些基于 RNN（循环神经网络）的方法[21-23]，但最近的方法大多基于 CNN（卷积神经网络）。

深度视频压缩 深度视频压缩的现有工作可分为两类，即非延迟约束和延迟约束。对于第一类，参考帧位置没有限制，这意味着参考帧可以来自未来。例如，Wu 等人[10] 提出将预测帧与先前帧和未来帧进行插值，然后对帧残差进行编码。Djelouah 等人[8] 也沿用了这种编码结构，并引入了光流估计网络，以获得更好的预测帧。Yang 等人[6] 为这种编码结构设计了一个循环增强模块。此外，[24, 25] 还提出了三维自动编码器来对图片组进行编码。这是通过增加输入维度对深度图像压缩的自然扩展。值得注意的是，这种编码方式会带来更大的延迟，GPU 内存成本也会显著增加。对于延迟受限的方法，参考帧只能来自前几帧。例如，Lu 等人[3] 设计了 DVC 模型，用网络取代了传统混合视频编解码器中的所有模块。随后，[4] 又提出了改进的 DVCPro 模型，它采用了[14] 中更先进的熵模型和更深的网络。按照与 DVC 相似的框架，Agustsson 等人设计了一种更先进的尺度空间光流估计。Hu 等人[26]考虑了编码运动矢量（MV）时的速率失真优化问题。在文献[7]中，单参考帧被扩展到多参考帧。最近，Yang 等人[6] 提出了一种基于 RNN 的运动矢量/残差编码器和解码器。在 [11] 中，残差是通过学习参数自适应缩放的。

我们的研究属于延迟受限方法，因为它可以应用于更多场景，如实时通信。与上述工作不同的是，我们设计了一个基于条件编码的框架，而不是沿用常用的残差编码。其他视频任务表明，利用时间信息作为条件是有帮助的 [27, 28]。在视频压缩方面，最近的文献 [15]、[16] 和 [12, 17] 对条件编码进行了一些研究。在 [15] 中，条件编码只针对熵建模。然而，由于缺少 MEMC，压缩比并不高，[15] 中的方法在 PSNR 方面无法超越 DVC。相比之下，我们为编码、解码和熵建模设计的条件编码能明显优于 DVCPro。在 [16] 中，只有编码器采用了条件编码。但解码器仍采用残差编码。由于使用了潜在状态，[16] 中的框架很难训练[7]。相比之下，我们使用显式 MEMC 来指导上下文学习，更容易训练。在 [12, 17] 中，视频内容需要明确分为跳过模式和非跳过模式，其中只有非跳过模式的内容才使用条件编码。相比之下，我们的方法不需要手工操作来分解视频。此外，DCVC 中的条件是特征域中的上下文，容量更大。总之，与文献[15]、[16]和[12, 17]相比，DCVC 中条件的定义、使用和学习方式都具有创新性。

3 建议的方法

本节将详细介绍拟议的 DCVC。我们首先描述 DCVC 的整个框架。然后，我们介绍压缩潜码的熵模型，接着介绍学习上下文的方法。最后，我们将提供有关训练的详细信息。

3.1 DCVC 框架

在传统视频编解码器中，帧间编码采用残差编码，其公式为

$$\hat{x}_t = f_{dec}\left[f_{enc}(x_t - \tilde{x}_t)\right] + \tilde{x}_t, \text{ 其中 } \tilde{x}_t = f_{predict}(\hat{x}_{t-1}). \quad (1)$$

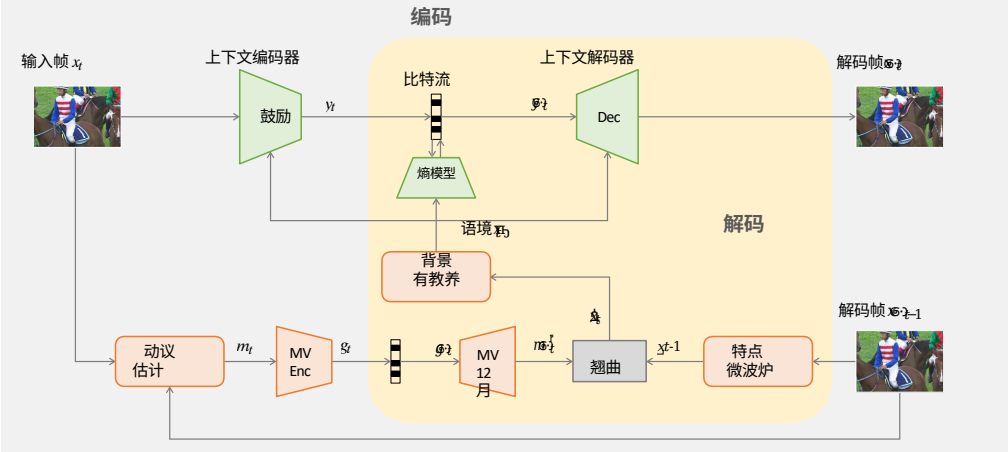


图 2：我们的 DCVC 框架。

x_t 是当前帧。$\hat{x}_t$ 和 $\hat{x}_{t-1}$ 是当前和上一个解码帧。$f_{enc}(\cdot)$ 和 $f_{dec}(\cdot)$ 是残差编码器和解码器。$f_{predict}(\cdot)$ 表示生成预测帧 $\tilde{x}_t$ 的函数。在传统的视频编解码器中，$f_{predict}(\cdot)$ 是以 MEMC 的方式实现的，它使用手工制作的编码工具来搜索最佳 MV，然后对预测帧进行插值。对于大多数现有的基于 DL 的视频编解码器 [3-9]，$f_{predict}(\cdot)$ 是完全由神经网络组成的 MEMC。

在本文中，我们没有采用常用的残差编码，而是尝试设计一种基于条件编码的框架，以获得更高的压缩比。事实上，一种直接的条件编码方式是直接将预测帧 $\tilde{x}_t$ 作为条件：

$$\hat{x}_t = f_{dec}\left(f_{enc}(x_t \mid \tilde{x}_t), \tilde{x}_t\right), \quad \text{其中} \ \tilde{x}_t = f_{predict}(\hat{x}_{t-1}). \tag{2}$$

然而，该条件在像素域中仍受到限制，通道尺寸较小。这将限制模型的容量。既然使用了条件编码，为什么不让模型自己学习条件呢？因此，本文提出了一种上下文视频压缩框架，即利用网络生成上下文，而不是预测帧。我们的框架可表述为

$$\hat{x}_t = f_{dec}\left(f_{enc}(x_t \mid \bar{x}_t), \bar{x}_t\right), \quad with \ \bar{x}_t = f_{context}(\hat{x}_{t-1}). \tag{3}$$

$f_{context}(\cdot)$ 表示生成上下文 $\bar{x}_t$ 的函数。$f_{enc}(\cdot)$ 和 $f_{dec}(\cdot)$ 是上下文编码器和解码器，与残差编码器和解码器不同。我们的 DCVC 框架如图 2 所示。

为了给编码 x_t 提供更丰富、更相关的条件，上下文处于维度更高的特征域中。此外，由于上下文的容量较大，其中的不同信道可以自由提取不同种类的信息。下面我们以图 3 为例进行分析。图中，右上部分显示了上下文中的四个通道示例。通过观察这四个信道，我们可以发现不同的信道有不同的侧重点。例如，与 x_t 中的高频可视化相比，第三通道似乎更强调高频内容。相比之下，第二和第四通道则像是在提取颜色信息，其中第二通道侧重于绿色，而第四通道则强调红色。得益于这些不同的上下文特征，我们的 DCVC 可以获得更好的重建质量，尤其是对于具有大量高频的复杂纹理。图 3 右下方的图像显示了 DCVC 与基于残差编码的框架相比所减少的重建误差。通过比较可以看出，DCVC 可以在背景和前景的高频区域实现非同小可的误差降低，而这些区域对于许多视频编解码器来说都是难以压缩的。

如图 2 所示，当前帧的编码和解码都以上下文 $\bar{x}_t$ 为条件。通过上下文编码器，x_t 被编码为潜码 y_t。然后，通过舍入运算，y_t 被量化为 $\hat{y}_t$。通过上下文解码器，最终得到重建帧 $\hat{x}_t$。在我们的设计中，我们利用网络自动学习 x_t 和 $\bar{x}_t$ 之间的相关性，然后

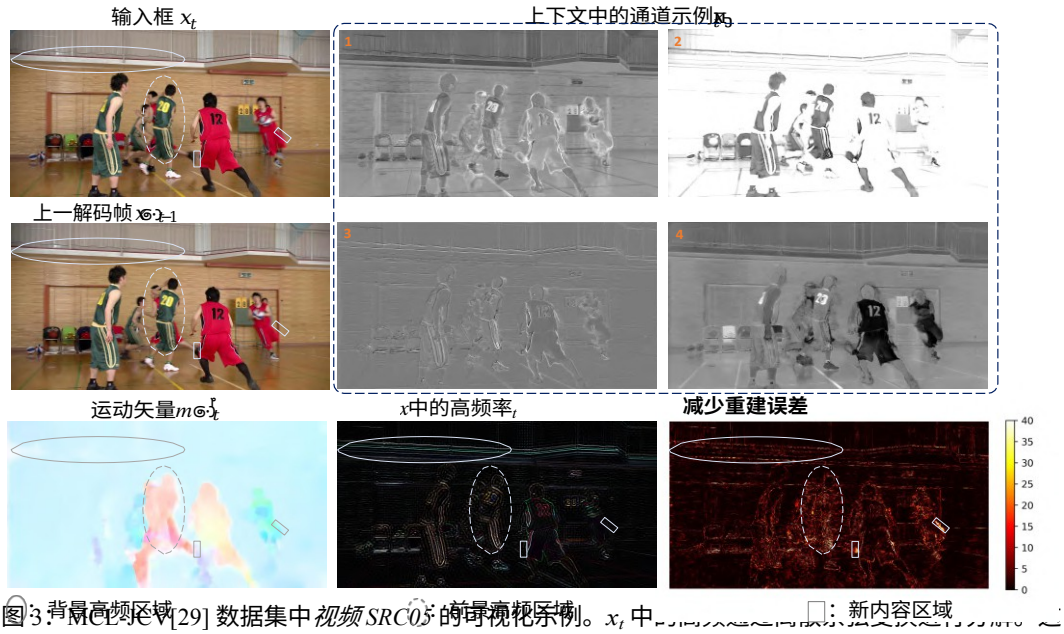


图 3：MCL-JCV [29] 数据集中*视频 SRC05* 的可视化示例。x_t 中[illegible]表明，DCVC 改善了小运动背景和大运动前景中高频内容的重建。此外，DCVC 擅长对运动引起的新内容区域进行编码，与基于残差编码的框架 DVCPro[4] 相比，其重构误差可显著降低。DCVC 的 BPP（每像素比特数）（0.0306）小于 DVCPro（0.0359）。

而不是在残差编码中使用固定的减法操作。从另一个角度看，我们的方法还具有自适应使用上下文的能力。由于视频中的运动，物体边界区域经常会出现新的内容。这些新内容很可能无法在之前解码的帧中找到很好的参照。在这种情况下，基于 DL 的帧残差编码视频编解码器仍不得不对残差进行编码。对于新内容来说，残差可能非常大，通过减法操作进行的间编码可能比内编码更糟。相比之下，我们的条件编码框架能够自适应地利用条件。对于新内容，模型可以自适应地倾向于学习内部编码。如图 3 中的重建误差减少所示，新内容的重建误差可以显著减少。

此外，本文不仅建议使用上下文 $\bar{x}_t$ 来生成潜码，还建议利用它来建立熵模型。下一小节将介绍更多细节。

3.2 熵模型

根据文献 [30]，估计概率分布与实际潜码分布之间的交叉熵是实际比特率的严格下限，即

$$R(\hat{y}_t) \geq \mathrm{E}_{\hat{y}_t \sim q_{\hat{y}}} [-\log p_{\hat{y}_t}(\hat{y}_t)], \tag{4}$$

$p_{\hat{y}_t}(\hat{y}_t)$ 和 $q_{\hat{y}_t}(\hat{y}_t)$ 分别是量化潜码 $\hat{y}_t$ 的估计概率质量函数和真实概率质量函数。实际上，算术编码几乎能以交叉熵的比特率对潜码进行编码。实际比特率 $R(\hat{y}_t)$ 与交叉熵比特率之间的差距可以忽略不计。因此，我们的目标是设计一个能准确估计潜码概率分布 $p_{\hat{y}_t}(\hat{y}_t)$ 的熵模型。我们的熵模型框架如图 4 所示。首先，我们使用超先验模型 [13] 来学习层次先验，并使用自动回归网络 [14] 来学习空间先验。这两种先验（层次先验和空间先验）是深度图像压缩中常用的先验。然而，对于视频来说，潜码还具有时间相关性。因此，我们建议使用上下文 $\bar{x}_t$ 来生成时间先验。如图 4 所示，我们设计了一个时间先验编码器来探索时间相关性。先验融合网络将学习

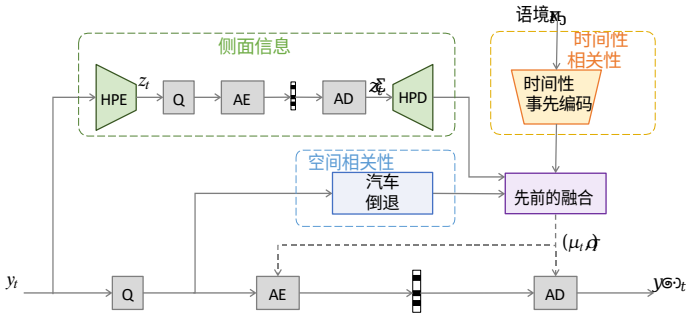


图 4：我们用于编码量化潜码 $\hat{y}_t$ 的熵模型。HPE 和 HPD 是超先验编码器和解码器。Q 表示量化。AE 和 AD 是算术编码器和解码器。

然后估计潜码分布的均值和尺度。本文沿用 xisting work [31]，假设 $p_{\hat{y}_t}(\hat{y}_t)$ 遵循拉普拉斯分布。$p_{\hat{y}_t}(\hat{y}_t)$ 的计算公式为：

$$p(\hat{y}_t \mid \bar{x}_t, \hat{z}_t) = \prod_i L(\mu_{t,i}, \sigma_{t,i}^2) * U(-\frac{1}{2}, \frac{1}{2}) (\hat{y}_{t,i}) \tag{5}$$

$$\text{与 } \mu_{t,i}, \sigma_{t,i} = f_{pf}(f_{hpd}(\hat{z}_t), f_{ar}(\hat{y}_{t,<i}), f_{tpe}(\bar{x}_t)) \text{。}$$

$f_{hpd}(\cdot)$ 是超先验解码器网络。$f_{ar}(\cdot)$ 是自动回归网络。$f_{tpe}(\cdot)$ 是专门设计的时间先验编码器网络。$f_{pf}(\cdot)$ 表示先验融合网络。在我们的熵模型中，$f_{ar}(\hat{y}_{t,<i})$ 和 $f_{tpe}(\bar{x}_t)$ 分别提供空间先验和时间先验。$f_{hpd}(\hat{z}_t)$ 提供无法从空间和时间相关性中学到的补充侧信息。

The entropy model formulated in Eq. 5 utilizes spatial prior for higher compression ratio. However, the operations about spatial prior are non-parallel and then lead to slow encoding/decoding speed. By contrast, all operations about the proposed temporal prior are parallel. Thus, we also provide a solution which removes spatial prior but relies on temporal prior for acceleration, namely $\mu_{t,i}, \sigma_{t,i} = f_{pf}(f_{hpd}(\hat{z}_t), f_{tpe}(\bar{x}_t))$.

3.3 情境学习

对于如何学习上下文，一种替代方案是直接使用由多个卷积层组成的普通网络，其中输入是上一个解码帧 $\hat{x}_{t-1}$，输出是 $\bar{x}_t$。然而，普通网络很难在没有监督的情况下提取有用信息。视频通常包含各种内容，存在大量复杂的运动。对于当前帧中的某个位置，参考帧中的定位位置可能相关性较低。在这种情况下，上下文 $\bar{x}_t$ 中的拼合位置与 x_t 中的位置相关性也较低，相关性较低的上下文无法促进对 x_t 的压缩。因此，我们也采用了 MEMC 的思想。但与通常在像素域应用 MEMC 不同，我们建议在特征域执行 MEMC。上下文生成函数 $f_{context}$ 的设计如下

$$f_{context}(\hat{x}_{t-1}) = f_{cr}(warp(f_{fe}(\hat{x}_{t-1}), \hat{m}_t)) \tag{6}$$

我们首先设计一个特征提取网络 $\check{x}_{t-1} = f_{fe}(\hat{x}_{t-1})$ 将参考帧从像素域转换到特征域。同时，我们使用光流估计网络 [32] 来学习参考帧 $\hat{x}_{t-1}$ 与当前帧 x_t 之间的 MV。然后对 MV m_t 进行编码和解码。解码后的 $\hat{m}_t$ 通过 warping 操作 $\ddot{x}_t = warp(\check{x}_{t-1}, \hat{m}_t)$ 引导网络提取上下文。$\ddot{x}_t$ 是一种相对粗糙的上下文，因为翘曲操作可能会引入一些空间不连续性。因此，我们设计了一个上下文细化网络 $f_{cr}(\cdot)$，以获得最终上下文 $\bar{x}_t = f_{cr}(\ddot{x}_t)$。在函数 $f_{context}(\cdot)$中，MV $\hat{m}_t$ 不仅可以指导网络从何处提取上下文，而且与没有 MEMC 的解决方案相比，还能使网络从更大的参考区域学习上下文。

表 1：BD-比特率比较

方法	MCL-JCV	UVG	HEVC B级	HEVC C级	HEVC D级	HEVC E级
DCVC (拟议)	**-23.9%**	**-25.3%**	**-26.0%**	**-5.8%**	**-17.5%**	**-11.9%**
DVCPro [4]	-4.1%	-7.9%	-9.0%	7.2%	-6.9%	17.2%
x265 (*veryslow*)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
DVC [3]	13.3%	17.2%	7.9%	15.1%	7.2%	21.1%
x264 (*veryslow*)	32.7%	30.3%	35.0%	19.9%	15.5%	50.0%

锚点是 x265（*veryslow*）。负数表示节省比特率，正数表示提高比特率。

3.4 培训

视频压缩的目标是以最小的比特率获得最佳的重构质量。因此，训练损耗包括两个指标：

$$L = \lambda \cdot D + R \tag{7}$$

对于不同的目标，D 可以是 MSE（均方误差）或 MS-SSIM（多尺度结构相似性）。在训练过程中，R 是作为潜码的真实概率和估计概率之间的交叉熵来计算的。

关于学习率，开始时设定为 1e-4，微调阶段设定为 1e-5。为了将 DCVC 与其他方法进行比较，我们按照文献 [5]，训练了 4 个不同 λ s 的模型 {MSE: 256, 512, 1024, 2048; MS-SSIM: 8, 16, 32, 64}。

4 实验结果

4.1 实验设置

训练数据 我们使用 Vimeo-90k septuplet 数据集 [33] 中的训练部分作为训练数据。在训练过程中，我们会将视频随机裁剪成 256x256 的小块。

测试数据 测试数据包括编解码标准社区常用测试条件[34]中的 HEVC Class B（1080P）、C（480P）、D（240P）、E（720P）。此外，还测试了来自 MCL-JCV[29] 和 UVG[35] 数据集的 1080P 视频。

测试设置 GOP（图片组）大小与 [4] 相同，即 HEVC 视频为 10，非 HEVC 视频为 12。由于本文只关注帧间编码，对于帧内编码，我们直接使用 CompressAI [36] 提供的现有深度图像压缩模型。我们使用 *cheng2020-anchor* [20] 作为 MSE 目标，使用 *hyperprior* [13] 作为 MS-SSIM 目标。

根据文献[37, 38, 31]中的性能比较，DVCPro[4]是近期文献[8, 6, 9, 5, 26]中基于 SOTA DL 的编解码器之一。因此，我们在本文中对 DVCPro 进行了比较。此外，我们还测试了其前身 DVC[3]。值得注意的是，为了公平比较，DVC 和 DVCPro 使用与 DCVC 相同的帧内编码进行了重新测试。在传统编解码器方面，测试了 x264 和 x265 编码器[39]。除了两个选项外，这两种编码器的设置与 [4] 相同。其一，我们使用 *veryslow* 预设值，而不是 *veryfast* 预设值。与*非常快*预设相比，*非常慢*预设能达到更高的压缩率。另一个是我们使用恒定量化参数设置，而不是恒定速率因子设置，以避免速率控制的影响。

4.2 性能比较

压缩比 图 5 和图 6 显示了这些方法的速率-失真曲线，其中图 5 中的失真度是用 PSNR 测量的，图 6 中的失真度是用 MS-SSIM 测量的。从图中可以看出，在所有比特率范围内，我们的 DCVC 模型都优

于 DVCPro。表 1 给出了相应的 BD-Bitrate [40] PSNR 结果。与使用 *veryslow* 预设值的 x265 相比，DVCPro 在 MCL-JCV、UVG、HEVC Class B 和 D 上分别节省了 4.1%、7.9%、9.0% 和 6.9% 的比特率。然而，对于 HEVC Class C 和 E，DVCPro 的表现更差，比特率分别增加了 7.2% 和 17.2%。相比之下，我们的 DCVC 在以下所有方面的表现都优于 x265

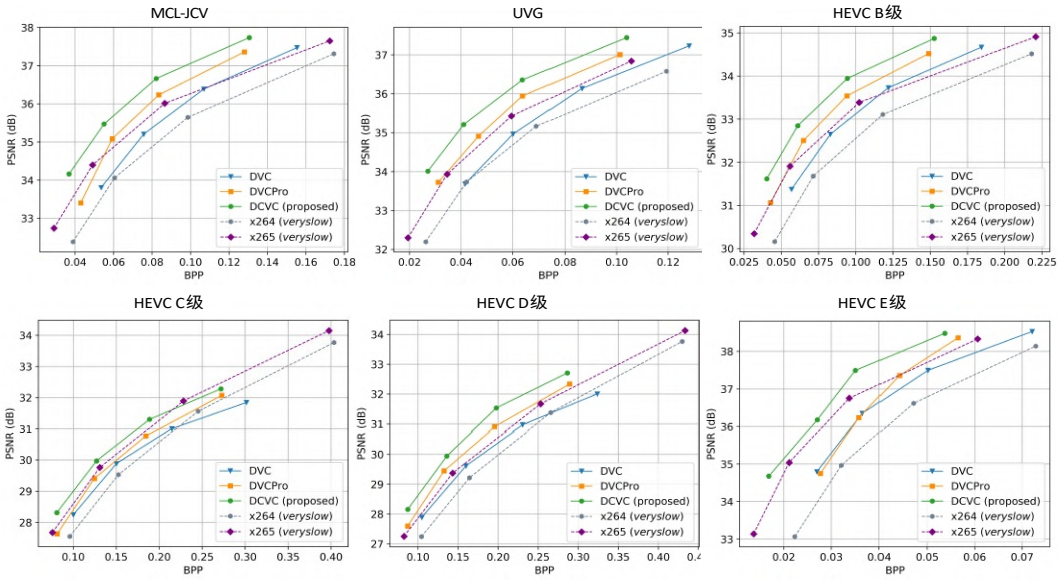


图 5：PSNR 和比特率比较。横轴为代表比特率成本的每像素比特数（BPP），纵轴为代表重建质量的 PSNR。

数据集。对于三个 1080p 数据集（MCL-JCV、UVG 和 HEVC Class B），比特率节省率分别为 23.9%、25.3% 和 26.0%。对于低分辨率的 HEVC C 类和 D 类视频，改进幅度也足够大，分别为 5.8% 和 17.5%。对于运动幅度相对较小的 HEVC E 类视频，比特率节省了 11.9%。通过这些比较，我们可以发现，对于不同分辨率和不同内容特征的视频，我们的 DCVC 可以明显优于 DVCPro 和 x265。

此外，我们还可以发现，DCVC 在高分辨率视频中的改进幅度更大。这是因为高分辨率视频包含更多高频纹理。对于这类视频，维度更高的特征域上下文更有帮助，能够承载更丰富的上下文信息来重构高频内容。

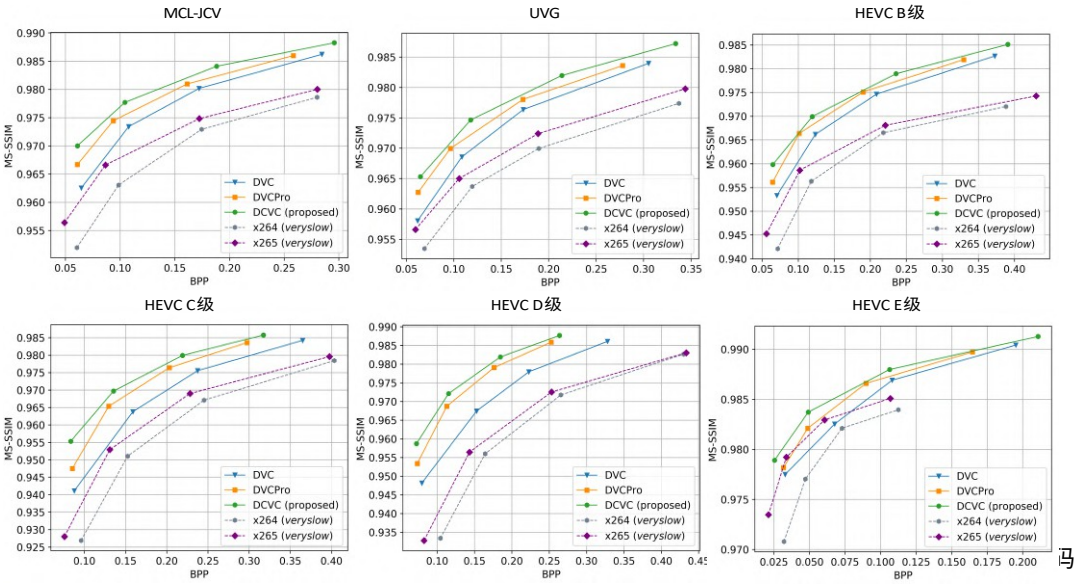
复杂性 DCVC 的 MAC（乘法累加运算）为 2268G，DVCPro 为 2014G，增加了约 13%。不过，在 P40 GPU 上，每 1080P 帧的实际推理时间 DCVC 为 857 毫秒，DVCPro 为 849 毫秒，只增加了约 1%，这主要归功于 GPU 的并行能力。

4.3 消融研究

条件编码和时间先验 在我们的 DCVC 中，我们建议使用基于连接的条件编码来取代基于减法的残差编码。同时，我们还为熵模型设计了时间先验。为了验证这些想法的有效性，我们进行了表 2 所示的消减研究，其中基线为我们的最终解决方案（即时间先验+串联上下文特征）。从表中我们可以发现，串联 RGB 预测和串联上下文特征都提高了压缩率。这验证了条件编码与残差编码相比的优势。此外，我们还可以发现，串联上下文特征的改进幅度远大于串联 RGB 预测。这说明了上下文在特征域中的优势。从表 2 中，我们还发现所提出的时序先验进一步提高了性能，其在条件编码（无论串联 RGB 预测还是串联上下文特征）下的改进幅度都大于在残差编码下的改进幅度。这些结果证明了我们的想法的优势。

熵模型 在 DCVC 中，除了超先验模型外，用于压缩量化潜码 $\hat{y}_t$ *的*熵模型同时利用了空间和时间先验，以获得更高的压缩比。然而，空间先验的缺点是编码/解码速度慢，因为它会带来空间依赖性和

非并行性。相比之下，拟议的时间先验的所有操作都是并行的。因此，我们的 DCVC 也支持移除空间先验，但依靠时间先验来加速。



和时间先验的消融研究

时间	先验聚合上下文	特征聚合 RGB	预测比特率提高
C	C		0.0%
C		C	5.4%
	C		4.6%
		C	8.7%
C			11.2%
			12.9%

得益于丰富的时间背景，没有空间先验的模型只增加了少量比特率。表 3 比较了空间先验和时间先验对性能的影响。从表中我们可以发现，如果禁用这两个先验，性能会大幅下降。如果启用这两个先验中的任何一个，性能都会有很大提高。如果同时启用这两个先验，性能还会进一步提高。不过，考虑到复杂度和压缩率之间的权衡，只使用超先验和时序先验的解决方案更好。这些结果表明了我们基于时间先验的熵模型的优势。

4.4 帧内编码的影响以及与更多基线的比较

为了建立最佳的基于 DL 的视频压缩框架，我们使用基于 SOTA DL 的图像压缩作为帧内编码。值得注意的是，为了进行公平比较，我们使用与 DCVC 相同的帧内编码对 DVC 和 DVCPro 进行了重新测试，其结果优于 SOTA DL 的报告。

表 3：熵模型消融研究

熵模型	提高比特率
超先验 + 空间先验 + 时间先验	0.0%
超先验 + 时间先验	3.8%
超先验 + 空间先验	4.6%

超先验	60.9%

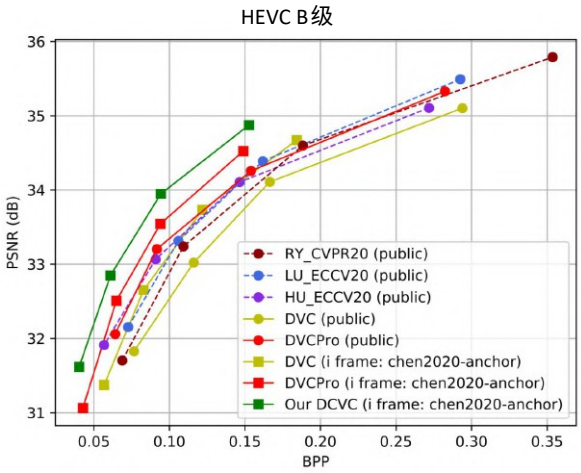


图 7：与公开结果的性能比较。后缀为 *（public）的方法结果由 [38, 31] 提供。后缀为 *（i 帧：*cheng2020*-anchor）的方法使用 *cheng2020-anchor* 作为帧内编码。

图 7 显示了我们重新测试的 DVC/DVCPro 与 [38] 提供的公开结果之间的性能对比。图 7 显示了我们重新测试的 DVC/DVCPro 与 [38, 31] 提供的公开结果之间的性能对比。此外，图 7 还显示了[38, 31]提供的 RY_CVPR20 [6]、LU_ECCV20 [5] 和 HU_ECCV20 [26]等最新作品的结果。从图中的公开结果可以看出，DVCPro 是近期研究中的一种 SOTA 方法。LU_ECCV20 和 HU_ECCV20 的结果与 DVCPro 非常接近。当 DVC 和 DVCPro 的帧内编码使用 CompressAI 提供的基于 SOTA DL 的图像压缩模型 *cheng2020-anchor 时*[36]，它们的性能有了很大的提高。如图 7 所示，当使用相同的帧内编码时，我们提出的 DCVC 方法的性能明显优于 DVCPro。

5 讨论

本文致力于设计一种基于条件编码的深度视频压缩框架，它比常用的基于残差编码的框架具有更低的熵限。基于残差编码的框架假定帧间预测总是最有效的，这是不充分的，尤其是在编码新内容时。相比之下，我们的条件编码可以在学习时间相关性和学习空间相关性之间进行调整。此外，在 DCVC 中，条件被定义为特征域上下文。维度更高的上下文可以提供更丰富的信息来帮助条件编码，尤其是对于高频内容。未来，高分辨率视频将更加流行。高分辨率视频包含更多高频内容，这意味着我们的 DCVC 的优势将更加明显。

在设计基于条件编码的框架时，核心问题是 *什么是条件？如何使用条件？如何学习条件？* 本文提出的 DCVC 解决方案回答了这些问题，并展示了其有效性。然而，这些核心问题仍有待解决。我们的 DCVC 框架具有可扩展性，值得进一步研究。通过更好地定义、使用和学习条件，设计更高效的解决方案大有可为。

在本文中，我们在训练过程中不对上下文中的通道添加监督。各通道之间可能存在冗余，这不利于充分利用高维度的上下文。今后，我们将研究如何消除各通道之间的冗余，以最大限度地利用上下文。在上下文生成方面，本文只使用了单参考帧。传统的编解码器表明，使用更多的参考帧可以显著提高性能。因此，如何在多个参考帧的情况下设计基于条件编码的框架是非常有前景的。此外，我们目前还没有考虑重建质量的时间稳定性，这可以通过后处理或额外的训练监督（如时间稳定性损失）来进

一步改善。

参考资料

[1] B.Girod, E. G. Steinbach, and N. Faerber, "Comparison of the H. 263 and H. 261 video compression standards," in *Standards and Common Interfaces for Video Information Systems：A Critical Review*》，1995 年。

[2] B.Bross、J. Chen、J.-R.Ohm、G. J. Sullivan 和 Y.-K.Wang, "Developments in international video coding standardization after AVC, with an overview of Versatile Video Coding (VVC)," *Proceedings of the IEEE*, 2021.

[3] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: an end-to-end deep video compression framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*、pp.11006-11015, 2019.

[4] G. Lu, X. Zhang, W. Ouyang, L. Chen, Z. Gao, and D. Xu, "An end-to-end learning framework for video compression," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[5] G. Lu, C. Cai, X. Zhang, L. Chen, W. Ouyang, D. Xu, and Z. Gao, "Content adaptive and error propagation aware deep video compression," in *European Conference on Computer Vision*, pp.

[6] R.Yang, F. Mentzer, L. V. Gool, and R. Timofte, "Learning for video compression with hierarchical quality and recurrent enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[7] J.Lin、D. Liu、H. Li 和 F. Wu，"M-LVC：用于学习视频压缩的多帧预测"，《视频压缩》，2011 年。*IEEE/CVF 计算机视觉与模式识别会议论文集*，2020 年。

[8] A.Djelouah、J. Campos、S. Schaub-Meyer 和 C. Schroers，"用于视频编码*的*神经帧间压缩"，《*IEEE/CVF 计算机视觉国际会议（ICCV）论文集*》，2019 年 10 月。

[9] E.Agustsson, D. Minnen, N. Johnston, J. Balle, S. J. Hwang, and G. Toderici, "Scale-space flow for end-to-end optimized video compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.

[10] C.-Y. Wu、N.Wu, N. Singhal, and P. Krähenbühl, "Video compression through image interpolation," in *ECCV*, 2018.

[11] R.Yang, Y. Yang, J. Marino, and S. Mandt, "Hierarchical autoregressive modeling for neural video compression," *9th International Conference on Learning Representations, ICLR*, 2021.

[12] T.Ladune, P. Philippe, W. Hamidouche, L. Zhang, and O. Déforges, "Optical flow and mode selection for learning-based video coding," in *22nd IEEE International Workshop on Multimedia Signal Processing*, 2020.

[13] J.Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *6th International Conference on Learning Representations, ICLR*, 2018.

[14] D.Minnen, J. Ballé, and G. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *arXiv preprint arXiv:1809.02736*, 2018.

[15] J.Liu, S. Wang, W.-C. Ma, M. Shah, R. Hu, P. Dhawan, and R. Urtasun, "Conditional entropy coding for efficient video compression," arXiv preprint arXiv.Ma, M. Shah, R. Hu, P. Dhawan, and R. Urtasun, "Conditional entropy coding for efficient video compression," *arXiv preprint arXiv:2008.09180*, 2020.

[16] O.O.Rippel、S.Nair、C.Lew、S.Branson、A.G.Anderson 和 L. Bourdev，"学习视频压缩"，《视频压缩》，第 2 卷第 2 期。*IEEE/CVF 计算机视觉国际会议论文集*》，第 3454-3463 页，2019 年。

[17] T.Ladune, P. Philippe, W. Hamidouche, L. Zhang, and O. Déforges, "Conditional coding for flexible learned video compression," in *Neural Compression：神经压缩：从信息论到应用--ICLR 研讨会*，2021 年。

[18] L.Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoen-coders," *arXiv preprint arXiv:1703.00395*, 2017.

[19] J.Ballé、V. Laparra 和 E. P. Simoncelli，"端到端优化图像压缩"，*arXiv 预印本 arXiv:1611.01704*, 2017。

[20] Z.Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF Conference on Computer Vision and*

Pattern Recognition, pp.

[21] G. Toderici、S. M. O'Malley、S. J. Hwang、D. Vincent、D. Minnen、S. Baluja、M. Covell 和 R. Sukthankar，《使用递归神经网络的可变速率图像压缩》，*arXiv 预印本 arXiv:1511.06085*, 2015。

[22] G. Toderici、D. Vincent、N. Johnston、S. Jin Hwang、D. Minnen、J. Shor 和 M. Covell，"利用递归神经网络*进行*全分辨率图像压缩"，《*电气和电子工程师学会计算机视觉与模式识别会议论文集*》，第 5306-5314 页，2017 年。

[23] N.Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S. J. Hwang, J. Shor, and G. Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.

[24] J.Pessoa, H. Aidos, P. Tomás, and M. A. Figueiredo, "End-to-end learning of video compression using spatio-temporal autoencoders," in *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, pp.

[25] A.Habibian, T. v. Rozendaal, J. M. Tomczak, and T. S. Cohen, "Video compression with rate-distortion autoencoders," in *Proceedings* of *the IEEE/CVF International Conference on Computer Vision*, pp.

[26] Z.Hu, Z. Chen, D. Xu, G. Lu, W. Ouyang, and S. Gu, "Improving deep video compression by resolution-adaptive flow coding," in *European Conference on Computer Vision*, pp.

[27] W.Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H.Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in Proceedings IEEE/CVF Conference Computer Vision and Pattern Recognition, pp.深度感知视频帧插值"，《*IEEE/CVF 计算机视觉与模式识别会议论文集*》，第 3703-3712 页、
2019.

[28] S.Niklaus 和 F. Liu，"用于视频帧插值的 Softmax splatting"，《*IEEE/CVF 计算机视觉与模式识别会议论文集*》，第 5437-5446 页，2020 年。

[29] H.Wang, W. Gan, S. Hu, J. Y. Lin, L. Jin, L. Song, P. Wang, I. Katsavounidis, A. Aaron, and C.-C. J. Kuo.MCL-JCV: a JND-based H. 264/AVC video quality assessment dataset," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp.

[30] C.E. Shannon，"通信的数学理论"，《*ACM SIGMOBILE 移动计算与通信评论*》，第 5 卷，第 1 期，第 3-55 页，2001 年。

[31] "PyTorchVideoCompression." https://github.com/ZhihaoHu/PyTorchVideoCompression.在线；2021 年 4 月 12 日访问。

[32] A.Ranjan 和 M. J. Black，"使用空间金字塔网络*进行*光流估计"，《*IEEE 计算机视觉与模式识别会议论文集*》，第 4161-4170 页，2017 年。

[33] T.Xue、B. Chen、J. Wu、D. Wei 和 W. T. Freeman，"面向任务流的视频增强"。
国际计算机视觉杂志》（IJCV），第 127 卷，第 8 期，第 1106-1125 页，2019 年。

[34] F.Bossen *等人*，《通用测试条件和软件参考配置》，*JCTVC-L1100*，第 12 卷，2013 年。

[35] "超视频群组测试序列"。http://ultravideo.cs.tut.fi。在线；2021 年 4 月 12 日访问。

[36] J.Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, "CompressAI: a PyTorch library and evaluation platform for end-to-end compression research," *arXiv preprint arXiv:2011.03029*, 2020.

[37] D.Xu, G. Lu, R. Yang, and R. Timofte, "Learned image and video compression with deep neural networks," in *2020 IEEE International Conference on Visual Communications and Image Processing, VCIP 2020, Macau, China, December* 1-4*, 2020,* pp.

[38] D.Xu, G. Lu, R. Yang, and R. Timofte, "Tutorial：利用深度神经网络学习图像和视频压缩。" https://drive.google.com/file/d/162omgk0CmHPBj4J7vWsNr8N9SPn5j97F/ view.在线；2021 年 4 月 12 日访问。

[39] "Ffmpeg。" https://www.ffmpeg.org/。在线；2021 年 4 月 12 日访问。

[40] G. Bjontegaard，"计算 RD 曲线之间的平均 PSNR 差异"，*VCEG-M33*，2001 年。