基于分布式编码架构的低复杂度深度视频压缩

Xinjie Zhang、Jiawei Shao 和 Jun Zhang香港科技
大学电子与计算机工程学系中国香港{xinjie.zhang,
jiawei.shao}@connect.ust.hk, eejzhang@ust.hk

arXiv:2303.11599v2 [eess.IV] 2 Apr 2023

摘要—**基于预测编码的主流视频压缩方法依赖复杂的编码器来减少时间冗余，因此难以部署在资源受限设备上。自 20 世纪 70 年代以来，分布式信源编码理论指出，通过独立编码并利用边信息（SI）进行联合解码，可以高效压缩相关信源。这启发了**分布式编码**架构，其目标是降低编码复杂度。然而，传统分布式编码方法的性能与预测编码方法相比仍存在显著差距。受基于学习的压缩取得巨大成功的启发，我们提出首个端到端分布式深度视频压缩框架，以提升率失真性能。其关键要素是解码器中的一种有效边信息生成模块，能够有效利用帧间相关性，而无需在编码器端进行计算密集型运动估计与补偿。实验表明，我们的方法显著优于传统分布式视频编码和 H.264。同时，在压缩性能相当的情况下，该方法的 6 ∼ 7× 编码加速比超过 DVC [1]。代码已发布于https://github.com/Xinjie-Q/Distributed-DVC。**

索引词—**分布式编码、低编码复杂度、深度视频压缩**

I. 引言

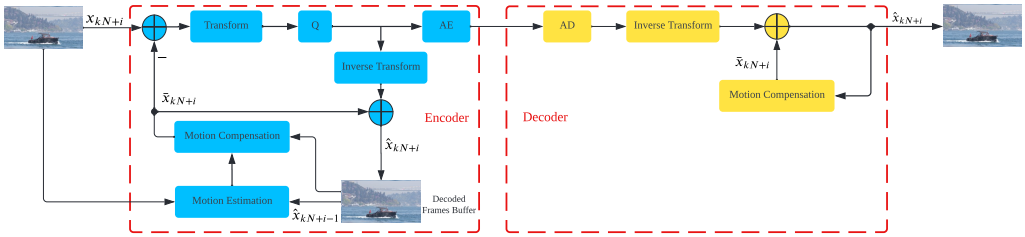
鉴于各类视频应用无处不在且广受欢迎，基于深度学习（DL）的视频压缩 [1]–[6] 因性能优于传统方法 [7], [8]而日益受到关注。大多数方法采用继承自标准视频编解码器的预测编码架构。如图 1(a) 所示，该架构在编码器端采用计算密集型的运动补偿预测环路，以显式减少帧间时间冗余。根据表 I，运动相关操作（即估计、补偿和压缩）在 DVC [1] 和 DCVC [5], 这两种具有代表性的深度学习视频编解码器中，分别约占计算复杂度的 90 % 和 65%。因此，这些基于深度学习的方法通常具有编码器负担较重、解码器相对简单的特点，适用于面向广播的应用（例如视频流 [9]和蓝光光盘 [10]），因为视频只需编码一次，却要解码多次。

近年来，视频监控 [11] 和多视角图像采集 [12], 等基于上行链路的视频应用迅速兴起。在这些应用中，视频编码器部署在计算资源和电源有限的设备上。此类场景需要低复杂度、低功耗的视频编码器，因此现有基于深度学习的视频编解码器并不适用。这要求对视频编码架构进行彻底变革。受 20 世纪 70 年代提出的分布式信源编码 Slepian-Wolf (SW) [13] 和 Wyner-Ziv (WZ) [14] 定理[1] 的启发，分布式视频编码[2]，（亦称 WZ视频编码 [15], ）已成为一种有前景的解决方案，可补充现有的视频压缩方法，以满足多媒体应用的需求。

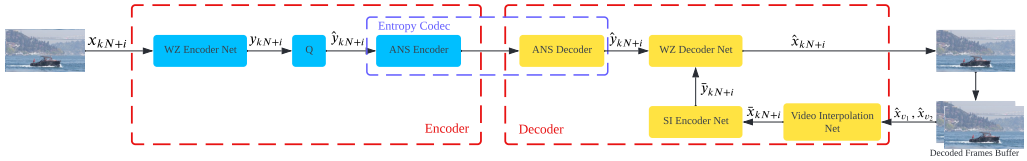
然而，直接应用 WZ 视频编码来构建高效且低复杂度的实用视频压缩系统并非易事。**首先**，现有 WZ视频编解码器的率失真（RD）性能仅与 H.264帧内编码 [16], [17] 相近。与主流标准视频编解码器相比仍有很大差距，这主要是由解码器处边信息（SI）的质量较差所致 [18]。目前尚不清楚如何生成高质量的边信息，以及如何在解码器处有效利用边信息，从而达到预测编码的性能。**其次**，传统 SW 编解码器中存在从解码器到编码器的反馈信道 [19]–[21]。解码器必须反复请求额外的校验位，直至成功完成解码过程，导致解码延迟较大。此外，还需要额外的帧缓冲区来存储编码码流，这会占用编码器大量内存，因此不适用于移动设备和摄像机。为确保视频应用的用户体验，开发资源友好型视频压缩方法至关重要。

本文设计了首个端到端分布式深度视频编码（分布式 DVC ）系统。该系统利用深度神经网络（ DNN ）在非线性变换和端到端优化方面的优势，提升编码增益，使其达到预测编码架构的水平；同时，通过从编码器端移除计算密集型运动操作，降低基于学习的视频编解码器所带来的高编码开销。本文的主要贡献总结如下。**首先**，我们提出的分布式 DVC 支持低复杂度编码器，这类编码器

[1]关于这两个定理的更多细节见附录A。[2]文献中常将分布式视频编码称为 DVC，但近年来 DVC 更多用于指代深度视频压缩，因此本文将分布式视频编码称为 WZ视频编码。关于 WZ视频编码的更多细节见附录 B。



(a) 标准视频编解码器和大多数基于学习的视频编解码器所采用的预测编码架构。



(b) 所提出的分布式深度视频编码架构。

图 1. 预测编码方法与所提出的分布式深度视频编码框架的比较。表 I 基于学习的视频编码器在 1080p 视频上的关键组件复杂度。测试采用 Intel Xeon Gold 6230R 处理器，基准频率为 2.10 GHz，并使用单个 CPU 核心。

延迟/浮点运算次数	运动			残差/WZ	总量
	估计	补偿	压缩	压缩	
DCVC [5]	13.02秒 1253.65G	12.74秒 732.83G	24.15秒 1049.95G	29.28秒 1557.32G	79.18秒 4593.79G
DVC [1]	13.53s 1253.65G	15.90秒 783.14G	12.60秒 635.04G	3.13秒 187.3G	45.16秒 2859.12 G
所提出的方法	0 0	0 0	0 0	6.68s 500.45G	6.68s 500.45G

非常适用于资源受限设备上的上行链路视频应用。尽管分布式信源编码理论早在数十年前就启发了这种架构，但仍缺乏能够实现接近预测编码架构的率失真性能的实用方法；我们的研究填补了这一空白。**其次**，我们提出了一种位于解码器端的边信息生成模块，用于显式捕获帧间时间相关性，从而重建信息更丰富的帧；同时引入编码器-解码器联合训练策略，以隐式地使潜在表示更加紧凑。此外，我们采用逐通道自回归熵模型 [22] ，在熵编码过程中为潜在表示提供准确的概率分布，从而有效减少统计冗余。同时，该模型取代了 SW 编解码器，并避免了传统 WZ 视频编码中反馈信道带来的问题。**最后**，大量实验结果表明，我们的框架相较于传统 WZ 视频编解码器 [17]在峰值信噪比方面最高可提升 10 dB，且性能优于 H.264。与 H.265 相比，在以多尺度结构相似性衡量时，我们的方法在大多数码率范围内都能获得更高的编码增益。同时，与 DVC [1]和 DCVC [5]，相比，其编码延迟分别降低约 85%和 90%。

II. 相关工作

基于学习的压缩。与标准编解码器相比，近期基于学习的图像和视频压缩方法在率失真性能方面取得了显著进展。大多数深度图像压缩方法采用自编码器式网络，并使用各种类型的熵模型以实现高压缩性能 [22]–[27]。此外，还提出了广义除法归一化（GDN） [28]、潜在残差预测和基于轮次的训练 [22] 等其他技术，以提高编码增益。我们认为这些现有工作是我们框架的重要组成部分。

现有的基于学习的视频压缩方法 [1]，[3]–[5] 主要采用预测编码架构。由于涉及大量运动相关操作，这些方法的编码复杂度较高。相比之下，我们的方法在率失真性能和编码复杂度之间实现了更好的权衡，适用于计算资源受限的场景，并可补充现有的基于学习的视频压缩方法。此外，与通过模型量化降低基于学习的图像编解码器复杂度的正交研究 [29]–[31]，不同，我们的工作旨在通过改变编码架构来降低编码复杂度。

分布式视频编码。过去几十年来，人们提出了各种人工设计工具来提升 WZ视频编码的压缩性能，包括采用不同的变换 [32]，、更准确的相关噪声估计 [33]以及改进边信息（SI）[34]，[35]。近期的研究引入了深度神经网络，以提升性能 [20]，[21]，[36]。然而，只有部分主要模块被深度神经网络取代，这阻碍了 WZ视频编解码器充分受益于端到端优化。相比之下，我们所提出的框架中的所有关键组件均由深度神经网络实现，并进行联合优化

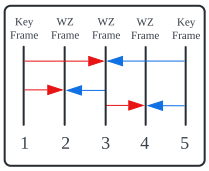


图 2. 分层帧插值顺序。

从而显著提升率失真性能。

III. 所提出的方法

符号说明。令 $\mathcal{X}=\{x_1,x_2,...\}$ 表示原始视频序列。WZ视频编码采用自适应或固定的帧分隔器，将视频序列划分为关键帧和WZ帧。为简单起见，我们假设图像组（GOP）的固定大小为 N。在这种情况下，x_{kN+1} 表示视频序列中的一个关键帧，其余帧 x_{kN+i} 为WZ帧，其中 $k=\{0,1,2,...\}$ 和 $i=\{2,3,...,N\}$。$\hat{}x_{kN+i}$ 表示重建WZ帧。为了利用时间相关性，解码器使用前两个解码帧 $\hat{}x_{v_1}$ 和 $\hat{}x_{v_2}$ 生成一个边信息帧 $\bar{x}_{kN+i}$，其中 v_1 和 v_2 表示参考帧的索引。可以使用变换编码来提高压缩效率。在这种情况下，原始帧 x_{kN+i} 和边信息帧$\bar{x}_{kN+i}$ 分别变换为 y_{kN+i} 和 $\bar{y}_{kN+i}$。$\hat{}y_{kN+i}$ 是 y_{kN+i}的量化版本。

A. 所提出的方法概述

图 1(b) 展示了我们框架的整体概况。下面将详细介绍各个步骤，并重点说明与经典 WZ视频编解码器 [17]的主要区别：

步骤 1. 变换与量化。我们以非线性 WZ 编码器网络取代 [17] 中的线性变换。输入的 WZ 帧 x_{kN+i} 被映射为表示 y_{kN+i}。然后对 y_{kN+i} 进行量化，得到 $\hat{}y_{kN+i}$。为实现端到端训练，我们采用 [22]中的量化方法。详情见第 III-B 节。

步骤 2. 熵编码。我们不再使用 SW 编解码器压缩量化 WZ 表示，而是在推理阶段采用非对称数制（ANS）编码器 [37]，将量化 WZ 表示 $\hat{}y_{kN+i}$ 编码为比特流，其中 $\hat{}y_{kN+i}$ 的概率分布由通道级自回归（ChAR）熵模型估计。更多详情见第 III-B 节。

步骤 3：边信息生成。采用 [38] 中基于光流的视频插值网络，按照图 2 所示的分层插值顺序，根据两个先前解码的帧 $\hat{}x_{v_1}$ 和 $\hat{}x_{v_2}$ 生成当前边信息帧 $\bar{x}_{kN+i}$。此外，设计了一个边信息编码器，将 $\bar{x}_{kN+i}$ 转换为边信息表示 $\bar{y}_{kN+i}$。更多信息见第 III-C 节。

步骤 4. 熵解码。解码器接收来自编码器的比特流，并执行 ANS 解码，以重建量化 WZ 表示 $\hat{}y_{kN+i}$。与 SW 解码相比，熵解码效率更高，且无需传统 WZ视频编码中的反馈信道。

步骤 5. 逆变换。我们将量化 WZ 表示 $\hat{}y_{kN+i}$ 与边信息表示 $\bar{y}_{kN+i}$拼接起来，并将其输入解码器网络，以重建 WZ 帧 $\hat{}x_{kN+i}$。这不同于在 [17] 中使用预定义量化表执行合并操作和重建过程。详情见第 III-B 节。

B. WZ 编码器和解码器网络

为便于压缩 WZ 帧 x_{kN+i}，我们借鉴深度图像压缩 [24], [25]，设计了一个基于 CNN 的 WZ 编码器—解码器网络。如图 3 所示，给定输入帧 x_{kN+i}，WZ 编码器生成表示 y_{kN+i}，随后将其量化为 $\hat{}y_{kN+i}$。WZ 解码器接收来自编码器的量化表示，并借助来自边信息生成模块的边信息表示 $\bar{y}_{kN+i}$，重建 WZ 信息 $\hat{}x_{kN+i}$。此外，我们采用 ChAR 熵模型 [22] 来估计 y_{kN+i} 的概率分布，其中超先验熵模型 [24] 生成超表示 $\hat{}z_{kN+i}$，以捕获 $\hat{}y_{kN+i}$ 各元素之间的空间冗余；ChAR 组件则利用 $\hat{}y_{kN+i}$ 各通道之间的相关性。这两部分共同生成条件高斯熵模型的均值和尺度参数。此外，由于量化操作不可微，我们采用 [22] 中的混合量化方法，以便通过随机梯度下降进行优化。具体而言，将添加均匀噪声的 y_{kN+i} 输入熵模型进行码率估计，而基于直通估计得到的取整表示 $\hat{}y_{kN+i}$ 则用作解码器的输入，以进行重建。

C. 边信息生成网络

如图 4 所示，我们的 SI生成网络由两部分组成：基于光流的视频插值网络和边信息编码器。我们采用 RIFE 网络[38] 进行视频插值，该网络可根据两帧输入进行任意时间步的帧插值。具体而言，给定前两帧解码帧 $\hat{}x_{v_1}$，$\hat{}x_{v_2}$ 及对应的时间步 t $(0\leq t\leq 1)$，中间光流估计网络会估计运动信息并生成一帧粗略的插值帧。随后，RefineNet 会减少伪影并细化高频区域，从而生成当前预测帧 $\bar{x}_{kN+i}$，使其尽可能接近当前帧 x_{kN+i}。接下来，我们使用与第 III-B 节所述 WZ 编码器相同网络的边信息编码器，生成边信息表示 $\bar{y}_{kN+i}$，然后将其与 WZ 表示 $\hat{}y_{kN+i}$ 拼接，以利用时间相关性生成质量更高的帧。网络的更多细节见附录 C。

D. 训练策略

损失函数。我们的目标是尽可能减少编码比特的数量，并降低原始 WZ 帧 x_{kN+i} 与重建WZ帧 $\hat{}x_{kN+i}$ 之间的失真。

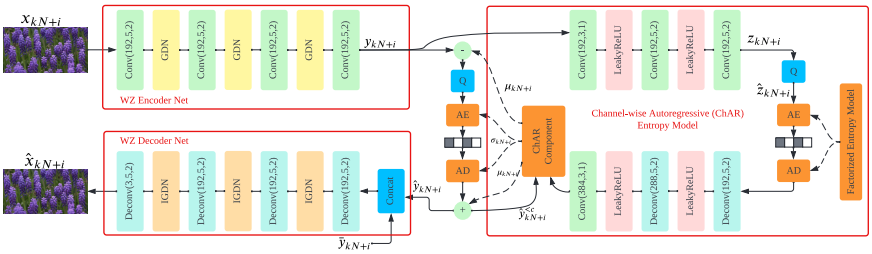


图 3. 我们所提出的 WZ 编码器-解码器网络。卷积参数的格式为（滤波器数量、卷积核大小、步幅）。AE 和 AD 分别表示 ANS 编码器和 ANS 解码器。



图 4. 边信息生成网络。

因此，我们使用以下由两个指标组成的损失函数进行训练：

$$\begin{aligned} L &= \lambda D + R \\ &= \lambda d(x_{kN+i}, \hat{x}_{kN+i}) + R(\hat{y}_{kN+i}) + R(\hat{z}_{kN+i}) \end{aligned} \tag{1}$$

其中，$d(x_{kN+i}, \hat{x}_{kN+i})$ 表示 x_{kN+i} 与 $\hat{x}_{kN+i}$之间的失真，对于不同任务，可以采用均方误差或多尺度结构相似性。$R(\hat{y}_{kN+i})$ 和 $R(\hat{z}_{kN+i})$ 分别表示对量化后的 WZ 表示 $\hat{y}_{kN+i}$ 及其对应的超先验表示 $\hat{z}_{kN+i}$进行编码所需的比特数。 λ 是拉格朗日乘子，用于控制比特成本 R 与失真 D之间的权衡。

两步训练。 训练期间，对于均方误差优化模型，我们采用两步训练流程。首先，使用 [38] 中的预训练 RIFE 网络，仅训练 WZ 编码器-解码器和边信息编码器。之后，联合微调整个模型，包括 RIFE 网络。对于多尺度结构相似性优化模型，则基于均方误差优化模型对所有模块进行微调。

IV. 实验

A. 实验设置

训练数据。 我们使用 Vimeo- 90k 数据集的训练部分 [39] 训练所提出的视频压缩框架，并将视频随机裁剪为 256×256图像块。训练期间，我们将批量大小设为 16。

测试设置。 我们在两个内容多样的 1080p 测试数据集 UVG [40] 和 MCL-JCV [41]上评估所提出方法的压缩性能。默认情况下，GOP 大小设为 8。参与比较的基准包括传统视频编解码器（WZ视频编码 [17], H.264[7] 和 H.265 [8]），以及近期提出的基于学习的方法（DVC [1], DVC-Lite [2] 和 DCVC [5]）。对于 H.264 和 H.265，我们报告顺序 P 帧顺序和层级 B 帧顺序下使用*veryslow* 模式的结果。关于 H.264 和 H.265 设置的详情，请参阅附录 D。

评估指标。 由于大多数基准测试的多尺度结构相似性值都高于 0.9，我们遵循 [24] 将该数值转换为分贝以提高可读性，其中多尺度结构相似性（dB）按 $10\log_{10}(1/(1-\text{多尺度结构相似性}))$ 计算。报告 Bjøntegaard Delta 比特率（BDBR） [42]，以表示平均在重建质量相同的情况下节省的比特率。此外，我们还计算 BD-PSNR 和 BD-MSSSIM，以表示在比特率相同的情况下重建质量的平均增益。

实现细节。 在我们的实验中，关键帧压缩使用 CompressAI 提供的预训练模型*mbt-2018* [25] [43]。对于 WZ 帧压缩，我们针对不同的 λ 值训练了五个模型（MSE：0.0018、0.0035、0.0067、0.0130、0.0250；多尺度结构相似性：2.40、4.58、8.73、16.64、31.73）。训练期间，使用 Adam 优化器 [44]，初始学习率设为 0.001。当评估损失趋于稳定时，学习率便减半。第一阶段和第二阶段的耐心值分别设为 10 轮和 5 轮。至于总训练轮数，两个阶段分别设为 100 轮和 50 轮。整个系统基于 PyTorch 实现，并在 NVIDIA RTX A5000 GPU 上训练。

B. 实验结果

图 5 展示了不同方法的率失真曲线。显然，与传统 WZ视频编解码器相比，我们的方法性能大幅领先；以峰值信噪比（多尺度结构相似性）衡量时，平均增益为 10 dB（6.94 dB），这表明端到端优化设计能够有效提升分布式编码的性能。与 H.264（P 帧）和 H.264（B 帧）相比，在 UVG（MCL-JCV）数据集上，按峰值信噪比衡量，我们的方法分别节省了 17.10%（16.79%）和 12.91%（10.85%）的比特率。与 H.265 相比，除极低码率区间外，我们的方法在多尺度结构相似性指标下均取得了更好的性能。表 II 报告了在 UVG 和 MCL-JCV 数据集上，与 H.264（P 帧）相比时，不同视频压缩方法的 BDBR 和 BD-PSNR（BD-MSSSIM）结果。我们看到，我们的方法取得了可与 DVC 相媲美的编码增益，并优于 DVC-Lite，这表明我们的分布式 DVC 是一种切实可行的尝试，有望达到预测编码的性能。每段视频的更多结果见附录 E。

我们还比较了四种基于学习的视频编解码器在 1080p 视频上的复杂度。如表 III 所示，在计算能力各异的 CPU 平台上（即 N_c=1、2、4、8），我们的方法相较于 DVC-Lite 实现了 3∼3.6× 倍的编码加速比，并节省了更多比特。与 DVC 相比，我们的方法实现了 6∼7× 倍的编码加速比，且解码延迟相近；在 UVG 和 MCL-JCV 数据集上的峰值信噪比分别仅下降 0.033 dB 和 0.049 dB。当编码器部署在高性能图形处理器上时，我们的方法仍可节省 60% 的编码时间。尽管性能

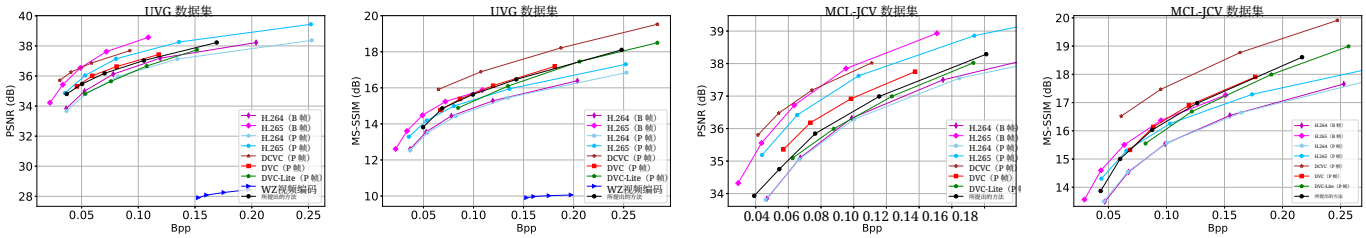


图 5. 峰值信噪比和多尺度结构相似性下的率失真性能。由于 WZ视频编解码器的串行实现效率低下，我们仅报告其在 UVG 数据集上的结果。

表 II 基于学习的视频编解码器与 H.264（P 帧）相比的 BDBR 和 BD-PSNR（BD-MSSSIM）结果。

方法	UVG				MCL-JCV			
	PSNR		多尺度结构相似性		PSNR		多尺度结构相似性	
	BDBR	峰值信噪比	BDBR	BD-MSSSIM	BDBR	峰值信噪比	BDBR	BD-MSSSIM
DCVC [5]	-52.96%	1.98dB	-59.07%	2.12dB	-52.18%	2.18dB	-61.29%	2.29dB
DVC [1]	-22.88%	0.68dB	-31.23%	0.75dB	-23.46%	0.91dB	-31.97%	0.83dB
DVC-Lite [2]	3.47%	-0.14dB	-12.05%	0.13dB	-4.18%	0.55dB	-22.42%	0.59dB
所提出的方法	-17.10%	0.52dB	-28.82%	0.64dB	-16.79%	0.90dB	-33.10%	1.00dB

表 III

学习基础的复杂度 D 视频编解码器在 1080p 视频上的表现。N_c 表示用于对视频进行编码的 CPU 核心数

方法	编码	编码延迟				解码	解码延迟				
	浮点运算次数	N_c=1	N_c=2	N_c=4	N_c=8	GPU	浮点运算次数	N_c=1	N_c=2	N_c=4	N_c=8	GPU
DCVC [5]	4593.79G	79.18秒	42.70秒	26.71秒	18.61秒	7.26秒	3030.68G	63.38秒	45.50秒	34.30秒	27.81秒	31.80秒
DVC [1]	2859.12G	45.16秒	24.40秒	15.51秒	9.38秒	0.44秒	1317.55G	23.78秒	15.16秒	10.07秒	6.45秒	0.23秒
DVC-Lite [2]	1062.14G	20.69s	12.28s	7.82s	4.86s	0.31秒	932.37G	18.31秒	10.75秒	6.82秒	4.25秒	0.21秒
所提出的方法	500.45G	6.68秒	3.98秒	2.29秒	1.34秒	0.18秒	2542.98G	22.43秒	14.09秒	9.36秒	6.49秒	0.19秒

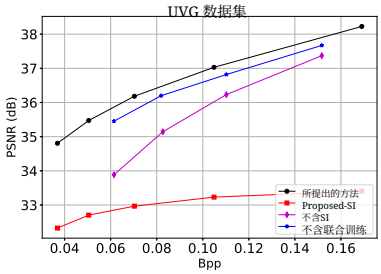


图 6. 消融实验。*Proposed-SI* 表示边信息帧的质量。不含*SI* 表示移除解码器处的SI生成网络。不含联合训练表示固定预训练编码器（包括熵模型），仅训练解码器部分。

与所提出的方法和 DCVC 之间存在差距，但我们的方法能够将编码延迟降低约 90%。这些结果表明，基于分布式编码架构的视频编解码器有望满足需要低复杂度编码器的上行链路应用需求。

C. 消融实验

边信息。 如第 III-C 节所述，我们提出了一种 SI 生成网络，以利用时间相关性。为验证其有效性，我们在 UVG 数据集上进行了一系列消融实验。如图 6 所示，我们展示了不同每像素比特数下边信息帧的质量。移除 SI 生成网络后，在相同的每像素比特数下，不含*SI* 的峰值信噪比下降约 1.17 dB，这表明边信息帧有助于我们的 WZ 解码器重建质量更高的帧。在表 IV 中，我们进一步分析了 SI 生成网络中的各个组件。固定 RIFE 网络会使比特率增加 8.05%，峰值信噪比降低 0.18 dB，这是因为预训练 RIFE 网络的目标只是更准确地估计中间帧，而非优化率失真性能。此外，如果不生成中间帧，而是按照层次顺序将两个解码帧拼接起来作为边信息，则所需比特率会增加 25.68%，峰值信噪比会降低 0.54 dB，这凸显了视频插值的优势。此外，在像素空间中利用边信息会使比特率最多增加 48.86%，峰值信噪比增益减少 0.94 dB，这证明了在特征空间中处理边信息的必要性。

编码器-解码器联合训练策略。 本文对编码器和解码器进行联合优化，以隐式减少帧间的部分时间冗余，从而使潜在表示 $\hat{}y_{kN+i}$ 更加紧凑。仅使用固定的预训练编码器和熵模型训练解码器网络时，图 6 所示的不含联合训练 方法会消耗更多比特，这表明我们的方法的潜在表示中低幅值元素更多，因此编码所需的比特更少。

V. 结论与讨论

本文提出了一种端到端分布式深度视频压缩框架，以提升分布式视频编码的压缩性能。所提出的方法兼具传统分布式视频编码低编码复杂度的优势，以及基于学习的压缩强大的非线性表示能力。实验结果表明，与传统分布式视频编码方法和 H.264 相比，我们的框架能够实现更高的编码效率。此外，与 DVC 相比，我们提出的方法在保持有竞争力的编码性能的同时，显著降低了编码器的

表 IV 边信息组件消融实验。第一行表示我们的最终方案，并被选为锚点。

RIFE 网络	SI 编码器	联合微调	比特率增加	PSNR 减少
✓	✓	✓	0%	0dB
✓	✓		8.05%	0.18dB
	✓	✓	25.68%	0.54dB
✓		✓	48.86%	0.94dB

复杂度。总体而言，我们的框架有望推动采用低复杂度编码器的深度视频压缩系统的发展。

尽管基于预测编码架构的深度视频压缩仍存在性能差距，但我们相信，通过利用深度学习的最新进展进一步提高编码效率，可以缩小这一差距。例如，可以采用尺度流 [3]和可变形补偿 [4]来提高边信息质量。此外，我们的框架在由不同摄像头传感器采集多个视频源 [12], 的应用场景中具有巨大潜力，在这些场景中，只有解码器能够利用视图间冗余。因此，值得投入更多研究力量，探索我们框架的潜力。

致谢

本研究获香港研究资助局通用研究基金（项目编号：16209622）资助。

参考文献

[1] 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》，发表于*CVPR*，2019年。[2] G. Lu、X. Zhang、W. Ouyang、L. Chen、Z. Gao 和 D. Xu，《一种用于视频压缩的端到端学习框架》，*TPAMI*，第43卷，第10期，第3292–3308页，2020年。[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》，发表于*CVPR*，2020年。[4] 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》，发表于*CVPR*，2021年。[5] J. Li、B. Li 和 Y. Lu，《深度上下文视频压缩》，发表于*NeurIPS*，2021年。[6] J. Li、B. Li 和 Y. Lu，《用于神经视频压缩的混合时空熵建模》，发表于*ACM*多媒体会议，2022年。[7] T. Wiegand、G. J. Sullivan、G. Bjontegaard 和 A. Luthra，《H.264/AVC视频编码标准概述》，*IEEE*电路与视频技术汇刊，第13卷，第7期，第560–576页，2003年。[8] G. J. Sullivan、J.-R. Ohm、W.-J. Han 和 T. Wiegand，《高效视频编码（HEVC）标准概述》，*IEEE*电路与视频技术汇刊，第22卷，第12期，第1649–1668页，2012年。[9] C.-L. Fan、W.-C. Lo、Y.-T. Pai 和 C.-H. Hsu，《360度视频流综述：采集、传输与显示》，*ACM*计算调查，第52卷，第4期，第1–36页，2019年。[10] N. Miyagawa，《蓝光光盘TM可录制／可重写媒体技术概述》，电子影像学，第7卷，第4期，第409–424页，2014年。[11] O. Elharrouss、N. Almaadeed 和 S. Al-Maadeed，《视频监控系统综述》，视觉通信与图像表示期刊，第77卷，第103116页，2021年。[12] X. Zhang、J. Shao 和 J. Zhang，《Ldmic：基于学习的分布式多视图图像编码》，发表于*ICLR*，2023年。[13] D. Slepian 和 J. Wolf，《相关信息源的无噪声编码》，*TIT*，第19卷，第4期，第471–480页，1973年。[14] A. Wyner 和 J. Ziv，《解码器端具有边信息的信源编码率失真函数》，*TIT*，第22卷，10，1976 1，1第 – 页。[15] B. Girod 、 A. M. Aaron 、 S. Rane 和 D. Rebollo-Monedero，《分布式视频编码》，*IEEE*会刊，第93卷，第1期，第71–83页，2005年。[16] V. K. Kodavalla 和 P. K. Mohan，“反馈免分布式视频编码中的色度分量编码”，载于 IMSAA，2011年。[17] V. K. Kodavalla 和 P. K. Mohan，“分布式视频编码中的色度分量编码方法”，载于 ICDCS，2012年。[18] F. Dufaux、W. Gao、S. Tubaro 和 A. Vetro，“分布式视频编码：趋势与展望”，JIVP，第 2009 卷，第 1–13 页，2010年。[19] A. Aaron、R. Zhang 和 B. Girod，“运动视频的 Wyner-Ziv 编码”，载于 ASILOMAR，2002年。[20] B. Dash、S. Rup、A. Mohapatra、B. Majhi 和 M. Swamy，“利用多层感知机网络集成生成解码器驱动的边信息，用于分布式视频编码”，Multimed. Tools. Appl.，第 77 卷，第 12 期，第 15221–15250 页，2018年。[21] B. Dash、S. Rup、A. Mohapatra、B. Majhi 和 M. Swamy，“利用基于切比雪夫多项式的 FLANN 技术进行解码器端 Wyner–Ziv 帧估计，用于分布式视频编码”，Multidim. Syst. Sign. P.，第 30 卷，第 3 期，第 1031–1061 页，2019年。[22] D. Minnen 和 S. Singh，“学习式图像压缩的通道级自回归熵模型”，载于 ICIP，2020年。[23] J. Ball´e、V. Laparra 和 E. P. Simoncelli，“端到端优化图像压缩”，载于 ICLR，2017年。[24] J. Ball´e、D. Minnen、S. Singh、S. J. Hwang 和 N. Johnston，“具有尺度超先验的变分图像压缩”，载于 ICLR，2018年。[25] D. Minnen、J. Ball´e 和 G. D. Toderici，“学习式图像压缩的联合自回归与分层先验”，载于 NeurIPS，2018年。[26] Z. Cheng、H. Sun、M. Takeuchi 和 J. Katto，“采用离散化高斯混合似然和注意力模块的学习式图像压缩”，载于 CVPR，2020年。[27] D. He、Y. Zheng、B. Sun、Y. Wang 和 H. Qin，“用于高效学习式图像压缩的棋盘格上下文模型”，载于 CVPR，2021年。[28] J. Ball´e、V. Laparra 和 E. P. Simoncelli，“使用广义归一化变换进行图像密度建模”，载于 ICLR，2016年。[29] W. Hong、T. Chen、M. Lu、S. Pu 和 Z. Ma，“通过定点推理实现高效神经图像解码”，IEEE电路与视频技术汇刊，第 31 卷，第 9 期，第 3618–3630 页，2020年。[30] H. Sun、L. Yu 和 J. Katto，“采用量化权重与激活值的端到端学习式图像压缩”，arXiv，2021年。[31] C. Jia、X. Hang、S. Wang、Y. Wu、S. Ma 和 W. Gao，“FPX-NIC：一种 FPGA 加速的 4K 超高清神经视频编码系统”，IEEE电路与视频技术汇刊，第 32 卷，第 9 期，第 6385–6399 页，2022 年。[32] A. Aaron 、 S. D. Rane 、 E. Setton 和 B. Girod，“用于视频的变换域 Wyner-Ziv 编解码器”，载于 VCIP，2004年。[33] C. Brites 和 F. Pereira，“用于高效像素域和变换域 Wyner–Ziv 视频编码的相关噪声建模”，IEEE电路与视频技术汇刊，第 18 卷，第 9 期，第 1177–1190 页，2008年。[34] X. Artigas、J. Ascenso、M. Dalai、S. Klomp、D. Kubasov 和 M. Ouaret，“DISCOVER 编解码器：架构、技术与评估”，载于 PCS，2007年。[35] P. Ren、P. Shi、C. Luo 和 Q. Liu，“一种利用光流算法生成 DVC 边信息的新方案”，载于 ICMT，2011年。[36] P. R. Bhagath、J. Mukherjee 和 S. Mukopadhayay，“利用解码器端深度卷积神经网络实现的无反馈信道分布式视频编码低复杂度编码器”，载于 ICVGIP，2016年。[37] J. Duda，“非对称数制系统：结合霍夫曼编码速度与算术编码压缩率的熵编码”，arXiv，2013年。[38] Z. Huang、T. Zhang、W. Heng、B. Shi 和 S. Zhou，“RIFE：用于视频帧插值的实时中间光流估计”，载于 ECCV，2022年。[39] T. Xue、B. Chen、J. Wu、D. Wei 和 W. T. Freeman，“采用面向任务的光流进行视频增强”，IJCV，第 127 卷，1125，2019年 8，1106 期，第 – 页。[40] A. Mercat、M. Viitanen 和 J. Vanne，“UVG 数据集：用于视频编解码分析与开发的 50/120 fps 4K 序列”，载于 MMSys，2020年。[41] H. Wang、W. Gan、S. Hu、J. Y. Lin、L. Jin、L. Song、P. Wang、I. Katsavounidis、A. Aaron 和 C.-C. J. Kuo，“MCL-JCV：基于恰可察觉差异的 H.264/AVC 视频质量评估数据集”，载于 ICIP，2016年。[42] G. Bjøntegaard，“RD 曲线之间平均峰值信噪比差异的计算”，ITU-T VCEG-M33，2001年。[43] J. B´egaint、F. Racap´e、S. Feltman 和 A. Pushparaja，“CompressAI：用于端到端压缩研究的 PyTorch 库和评估平台”，arXiv，2020年。[44] D. P. Kingma 和 J. Ba，“Adam：一种随机优化方法”，载于 ICLR，2015年。

附录

A. 基础

如图7 (a)所示，分布式编码是指对两个（或更多）在统计上相关、但在物理上相互分离的信源分别进行编码并联合解码。联合解码过程旨在利用不同信源之间的统计相关性，实现高效压缩。1973年，Slepian 和 Wolf 在基本的双信源情形下研究了分布式无损编码问题。随后，Wyner 和 Ziv 将 Slepian-Wolf（SW）定理扩展到有损情形，即 Wyner-Ziv（WZ）定理；该定理给出了给定失真条件下可达到的比特率下界，如图7 (b)所示。这种架构已扩展至视频编码领域，称为分布式视频编码 [15]，其目标是对每一帧独立编码，并利用先前解码帧生成的边信息进行联合解码。SW 定理和 WZ 定理的正式表述如下。

定理*1*（*Slepian-Wolf*）：考虑两个统计独立的独立同分布源 A 和 B，无失真压缩 A 和 B 的可达速率区域为：

$$R_A \geq H(A|B), R_B \geq H(B|A), R_A + R_B \geq H(A) + H(B),$$

其中，R_A 和 R_B 分别表示传输 A 和B的速率。

定理*2*（*Wyner-Ziv*）：假设源 A 和 B 在统计上相关。给定失真级别 D，在解码器处有边信息B 时，编码 A 所需的最小速率 R_{WZ}（D）大于或等于 $R_{A|B}$（D），其中 B 在编码器和解码器两端均可用。将解码器的输出记为 A，我们有以下等价关系：

$$R_{WZ}(D) \geq R_{A|B}(D) = \min_{E[d(A,\hat{A})] \leq D} I(A; \hat{A}|B),$$

其中，当 A 和 B 联合高斯时，R_{WZ}（D）$= R_{A|B}$（D）而 A 与A 之间的失真 d（A，$\hat{}A$）以均方误差衡量。

SW 定理指出，对于两个相关的数据源，采用独立编码器和联合解码器进行无损压缩，其渐近压缩率可达到采用联合编码器和解码器进行最优压缩时的压缩率。WZ 定理将这一思想扩展至有损压缩，并证明即使编码器不具备边信息，也不会造成率失真性能损失。SW 和 WZ 定理表明，可以对两个在统计上相互依赖的信号进行分布式压缩（独立编码、联合解码），同时达到预测编码方法（联合编码和解码）的率失真性能。有关分布式编码及其在视频编码中的应用，详情请参阅 [15]和 [18]。

B. 经典*WZ*视频编码简介

图 8 展示了经典的 WZ 视频编码架构[17]。WZ 视频压缩的编码和解码过程简要概述如下：

步骤 1. 变换与量化。对输入的 WZ 帧 x_{kN+i} 应用离散余弦变换（DCT），将其变换为 y_{kN+i}。随后，根据预定义的量化级$\mathcal{Q} = \{0, 2^m | m = 1, ..., 7\}$，对 y_{kN+i} 进行均匀量化，得到 $\hat{}y_{kN+i}$。值 0 表示某些变换频带未被编码，在解码器处将由边信息（SI）中对应的频带替代；其余频带则被划分为多个位平面，并由下一个模块处理。

步骤 2. Slepian-Wolf 编码。使用 LDPC 累加编码器分别对 $\hat{}y_{kN+i}$ 的位平面进行编码，生成相应的校验信息，并将其存储在缓冲区中；随后根据解码器通过反馈信道发出的请求，分块发送这些信息。

步骤 3：边信息生成。基于前两个已解码帧 $\hat{}x_{v_1}$，$\hat{}x_{v_2}$ 以及图 2 所示的分层帧插值顺序，采用运动补偿帧插值算法生成一个 SI 帧$\bar{x}_{kN+i}$，并将其变换为 $\bar{y}_{kN+i}$。$\hat{}y_{kN+i}$ 与 $\bar{y}_{kN+i}$ 之间的相关噪声采用拉普拉斯分布建模，作为虚拟信道模型。随后，利用 SI 表示 $\bar{y}_{kN+i}$、相关噪声和先前解码的位平面，估计每个位平面的软信息（即条件比特概率 P_{cond}）。

步骤 4：Slepian-Wolf 解码。给定软信息，通过反馈信道从编码器缓冲区请求连续的数据块校验位，逐个位平面进行解码，直到达到较低的比特错误概率。

步骤 5：重建和逆变换。首先，将 SI 的高频频带与已解码频带组合，以重建变换频带；随后进行反量化和逆变换，得到重建WZ帧 $\hat{}x_{kN+i}$。

C. 网络架构

ChAR 组件。如图9所示，每个 ChAR 块由三个模块组成，包括均值变换模块、尺度变换模块和潜在残差预测模块。需要注意的是，将量化切片 $Q(y_j) = Round(y_j - \mu_j) + \mu_j$ 与当前切片的均值 μ_j 及先前解码切片 $\hat{}y_{<j}$ 拼接，以得到预测残差 r_j。为减小量化误差，我们将r_j 和 $Q(y_j)$相加，以得到当前解码切片 $\hat{}y_j$。**RIFE 网络**。如图10所示，RIFE 网络由中间光流估计网络（IFNet）和 RefineNet 组成。给定两个先前解码帧 $\hat{}x_{v_1}$，$\hat{}x_{v_2}$及对应的时间步 t ($0 \leq t \leq 1$)，IFNet 会估计运动信息并生成粗略的插值帧。随后，使用 RefineNet 优化高频区域并减少伪影，从而生成当前预测帧 $\bar{x}_{kN+i}$，使其尽可能接近当前帧 x_{kN+i}。IFNet 采用多个在不同分辨率下堆叠的 IFBlock，以通过以下公式获得粗略的插值帧：

$$\begin{aligned} \tilde{I}_t &= M \odot \tilde{I}_{t\leftarrow 0} + (1 - M) \odot \tilde{I}_{t\leftarrow 1} \\ \tilde{I}_{t\leftarrow 0} &= \overleftarrow{\mathcal{W}}(I_0, F_{t\to 0}), \quad \tilde{I}_{t\leftarrow 1} = \overleftarrow{\mathcal{W}}(I_1, F_{t\to 1}) \end{aligned} \tag{2}$$

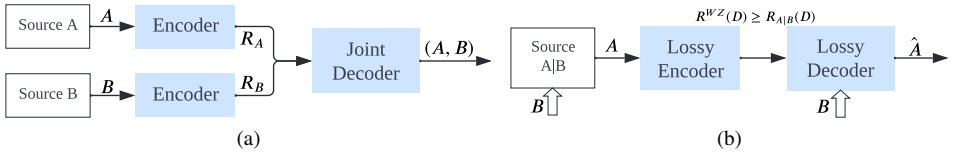


图7. (a)：对两个相关信源 A 和 B进行分布式压缩。解码器联合解压 A 和 B，以利用它们之间的相互依赖。(b)：利用统计相关的边信息 B对信源 A 进行有损压缩。

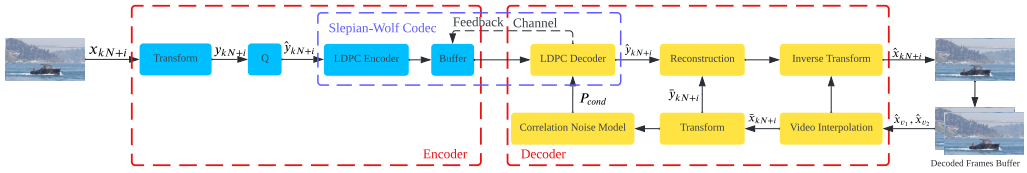


图8. 传统的 WZ视频编码架构。

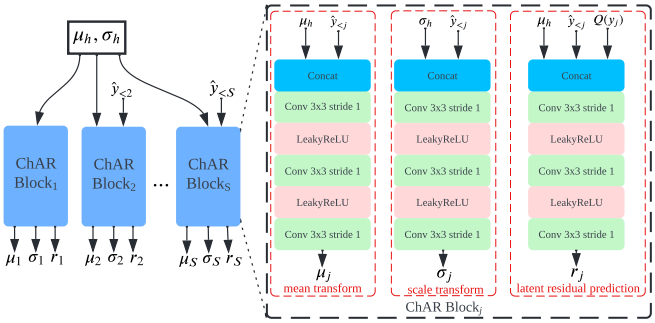


图9. ChAR 组件的网络结构。 μ_h 和 σ_h 分别表示超先验熵模型估计的均值和方差。由于在获得 μ_j 和 σ_j 后才解码 $\hat{y}_j$，因此会依次执行 S ChAR块。我们在所提出的模型中将 S 设为 8。

其中， M 表示融合图（$0 \le M \le 1$），操作$\odot$ 是逐元素乘法器， $\overleftarrow{W}$ 表示图像反向扭曲。对于每个 IFBlock，首先根据来自（$i-1$）th IFBlock 的估计光流 F^{i-1}，将两个输入帧 I_0 和 I_1 扭曲到当前帧 $I_{t\leftarrow 0}$ 和 $I_{t\leftarrow 1}$。然后，我们沿通道维度拼接输入帧 I_0， I_1、扭曲后的帧 $\tilde{I}^{i-1}_{t\leftarrow 0}$， $\tilde{I}^{i-1}_{t\leftarrow 1}$、当前时间步 t、先前光流 F^{i-1}和融合图 M^{i-1}，并将它们输入一系列双线性运算和卷积运算中，以近似光流残差和融合图。得到最终光流 F^2 和融合图 M^2后，我们使用公式 公式2来获得$\tilde{}$插值帧 I_t。

为细化高频区域并减少 I_t的伪影$\tilde{}$，采用 RefineNet 生成重建残差 $\Delta(-1 \le \Delta \le 1)$。具体而言，上下文提取器首先分别从输入帧 I_0 和 I_1中提取多尺度上下文特征 C_0 和 C_1。基于中间光流 $F^2_{t\to 0}$ 和 $F^2_{t\to 1}$，将这些特征变形为对齐特征 $C_{t\leftarrow 0}$ 和 $C_{t\leftarrow 1}$。同时，将输入帧 I_0， I_1、变形后的帧 $\tilde{I}^2_{t\leftarrow 0}$， $\tilde{I}^2_{t\leftarrow 1}$、估计光流 $F^2_{t\to 0}$， $F^2_{t\to 1}$ 以及融合图 M^2 拼接后，输入 U-Net细化网络的编码器，并在 $C_{t\leftarrow 0}$和 $C_{t\leftarrow 1}$的帮助下生成$\bar{}$ $\tilde{}$经过细化的重建帧 $I_t = I_t + \Delta$。

D. 实验细节

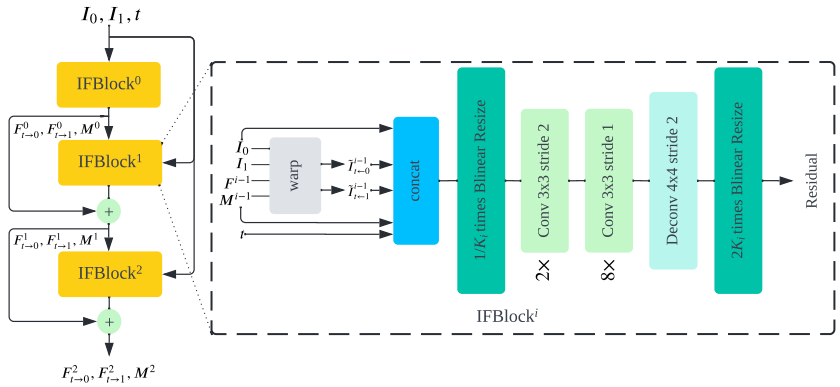
关键帧压缩。 在基于学习的视频编解码器中，使用 CompressAI [43] 提供的预训练模型 *mbt-2018* [25] （质量索引为 4）压缩关键帧（即每个 GOP 的第一帧）。为进行公平比较，使用其开源代码重新测试采用上述帧内编码方法的 DVC [3] [1] 和 DCVC [4] [5]。对于传统视频编解码器，H.264 和 H.265 使用其默认的帧内编码方法，而 WZ视频编码 [17] 采用 H.265 帧内编码方法压缩关键帧。

H.264 和 H.265 设置。我们使用以下命令，通过顺序 P 模式和分层 B 模式实现 H.264 和 H.265 编码方案。具体而言，给定分辨率为 $W\times H$ 的序列 *Video.yuv* ，使用 x264 和 x265 编解码器生成压缩视频的命令行如下：

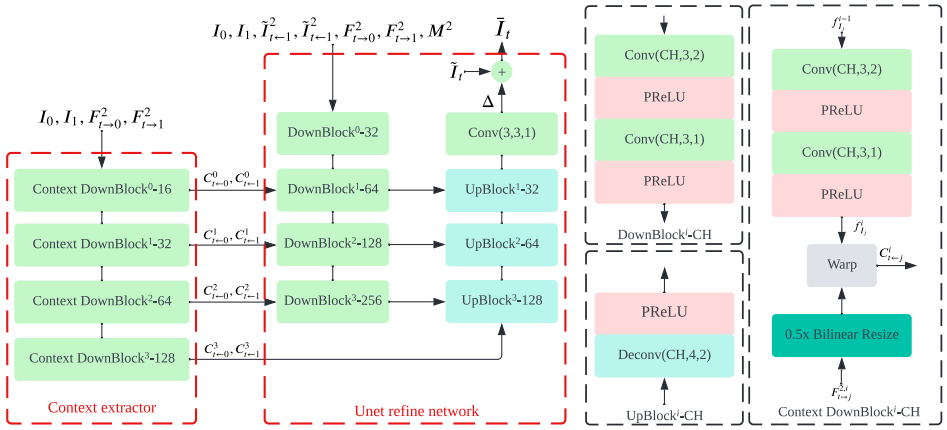
- H.264(P)：*ffmpeg -pix fmt yuv420p -s:v W×H -i Video.yuv -vframes* N_e *-c:v libx264 -preset veryslow -x264- params “ crf=CRF:keyint=GOP:bframes=0:scenecut=0” output.mkv*
- H.264(B)：*ffmpeg -pix fmt yuv420p -s:v W×H -i Video.yuv -vframes* N_e *-c:v libx264 -preset veryslow -x264-params “crf= CRF:keyint=GOP:scenecut=0:b- adapt=0:bframes=BF:b-pyra mid=1” output.mkv*
- H.265(P)：*ffmpeg-pixfmtyuv420p -s:vW×H-iVideo.yuv -vframes* N_e *-c:v libx265 -preset veryslow -x265-params “crf= CRF:keyint=GOP:bframes=0” output.mkv*
- H.265(B)：*ffmpeg -pixfmt yuv420p -s:vW×H -iVideo.yuv -vframes* N_e *-c:v libx265 -preset veryslow -x265-params “crf= CRF:keyint=GOP:b-adapt=0:bframes=BF:b- pyramid=1” output.mkv*

其中， N_e、CRF、BF分别表示编码帧的数量、量化参数和 B 帧的数量。对于分层 B 模式，我们将 BF 设为 GOP-1。

[3]https://github.com/ZhihaoHu/PyTorchVideoCompression/tree/master/DVC [4] https://github.com/DeepMC-DCVC/DCVC



(a) IFNet 的网络结构。每个 IFBlock 都有一个分辨率参数，即 $(K_0,\ K_1,\ K_2) = (4,\ 2,\ 1)$。



(b) RefineNet 的网络结构，包含一个上下文提取器和一个 U-Net细化网络。

图 10. RIFE 的网络结构。

E. 视频级别分析

在图 11 中，我们将 H.264（P 帧）设为锚点，计算 UVG 和 MCL-JCV 数据集中每个视频的 BDBR。随后，我们绘制每个压缩视频相对于 H.264 的文件大小，例如，BDBR 为 30% 表示相对大小为 70%（即，100%-30%）。

观察发现，在大多数视频上，我们所提出的方法生成的编码文件比 H.264 和 DVC-Lite 更小。与 DVC 相比，按峰值信噪比衡量，使用所提出的方法压缩的视频中约有一半（16/37）的文件大小更小或相同。然而，对于一部分运动信息变化快速的视频（例如 ReadySetGo、视频 03、12 和 13），如果编码器不利用时间信息，所提出的模型就难以取得更优的峰值信噪比性能。此外，由于训练数据集中缺少动画视频，我们的方法无法很好地泛化到 MCL-JCV 数据集中的卡通视频（即视频 18、20、24、25）。

此外，与峰值信噪比相比，我们的方法在多尺度结构相似性方面具有更高的压缩效率，这部分归因于解码器在特征空间而非像素空间中利用边信息。因此，网络倾向于关注结构信息，而非像素信息。尽管在某些视频上，所提出的分布式 DVC 框架仍是实现接近预测编码性能的一种可行尝试。

F. 特征可视化

为了更好地理解边信息表示与 WZ 表示之间的差异，我们在图 12 和图 13 中展示了 WZ 和边信息特征图的可视化结果。从可视化结果可以看出，与边信息表示$\bar{y}_{kN+i}$相比，$\hat{y}_{kN+i}$中用于表示低频（高频）信息的通道更多（更少），例如第 128（30）和第 178（170）个通道。此外，$\bar{y}_{kN+i}$中只有少数通道（例如第 48 和第 123 个通道）突出显示低频内容，而边信息表示的大多数通道（例如第 151 和第 174 个通道）似乎更关注鸟类边缘和轮廓的高频细节，与 x_{kN+i}中的高频信息形成对比。这与传统 WZ 视频编码中的重建操作相符：边信息帧的高频离散余弦变换系数直接用作重建WZ帧的高频系数。这些结果表明，未来值得探索如何充分利用边信息特征中的高频信息，并移除 WZ 特征中不重要的高频特征图，以提升压缩性能。

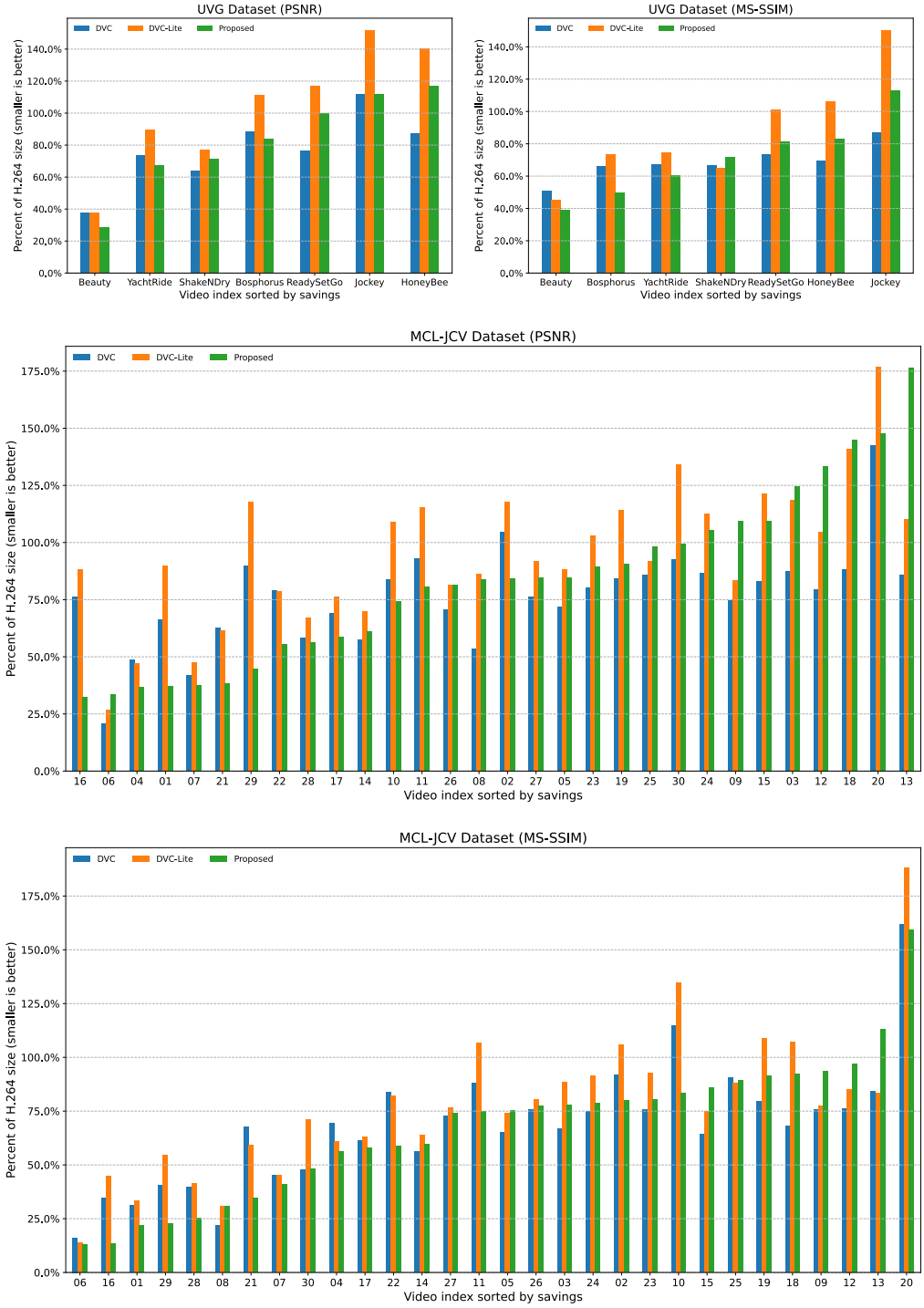


图 11. UVG 和 MCL-JCV 数据集上各视频的码率节省。数值表示在相同重建水平下，根据 BDBR 衡量时相对于 H.264(P) 的相对大小。

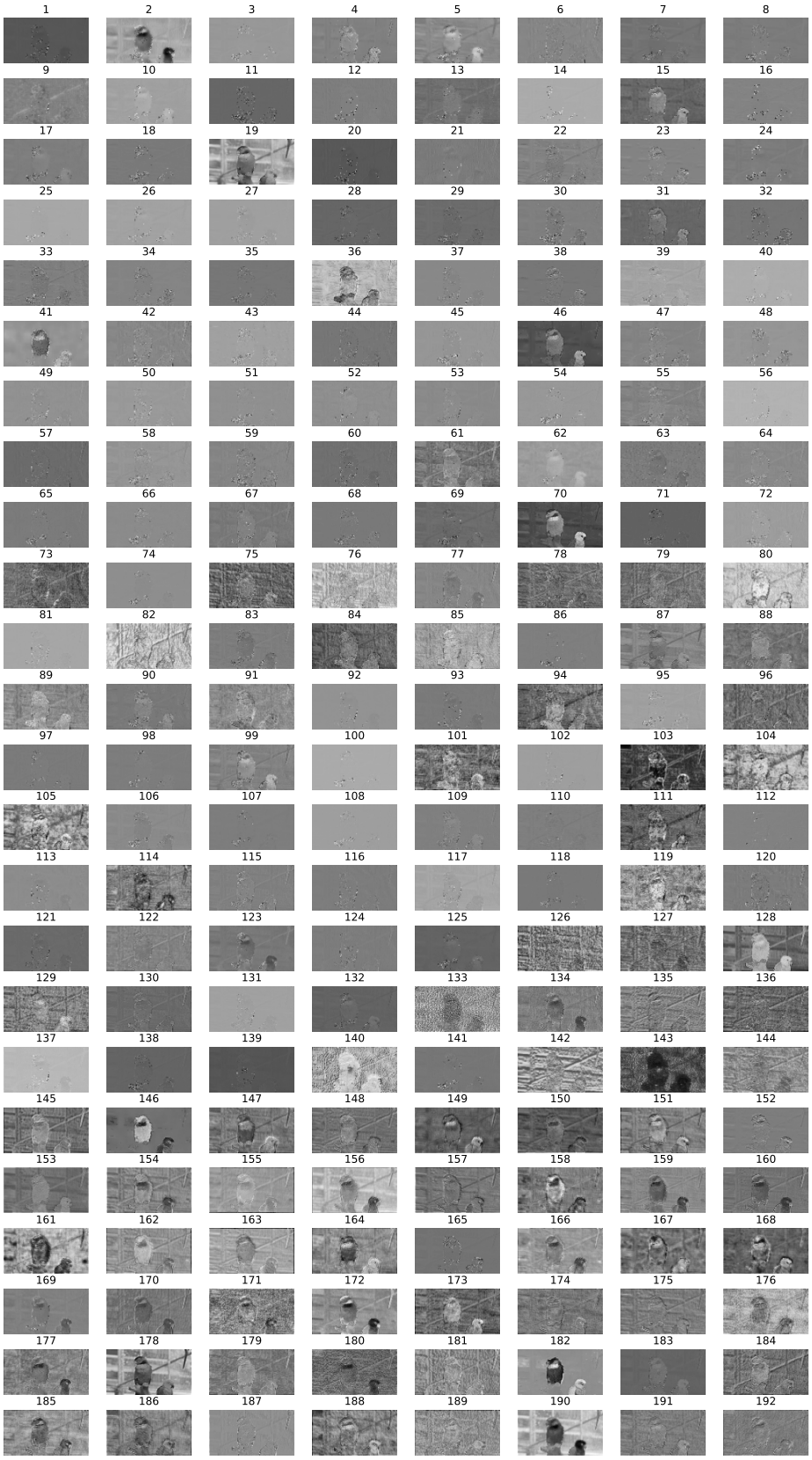


图 12. MCL-JCV 数据集中*videoSRC30* 序列的 Wyner-Ziv表示可视化。

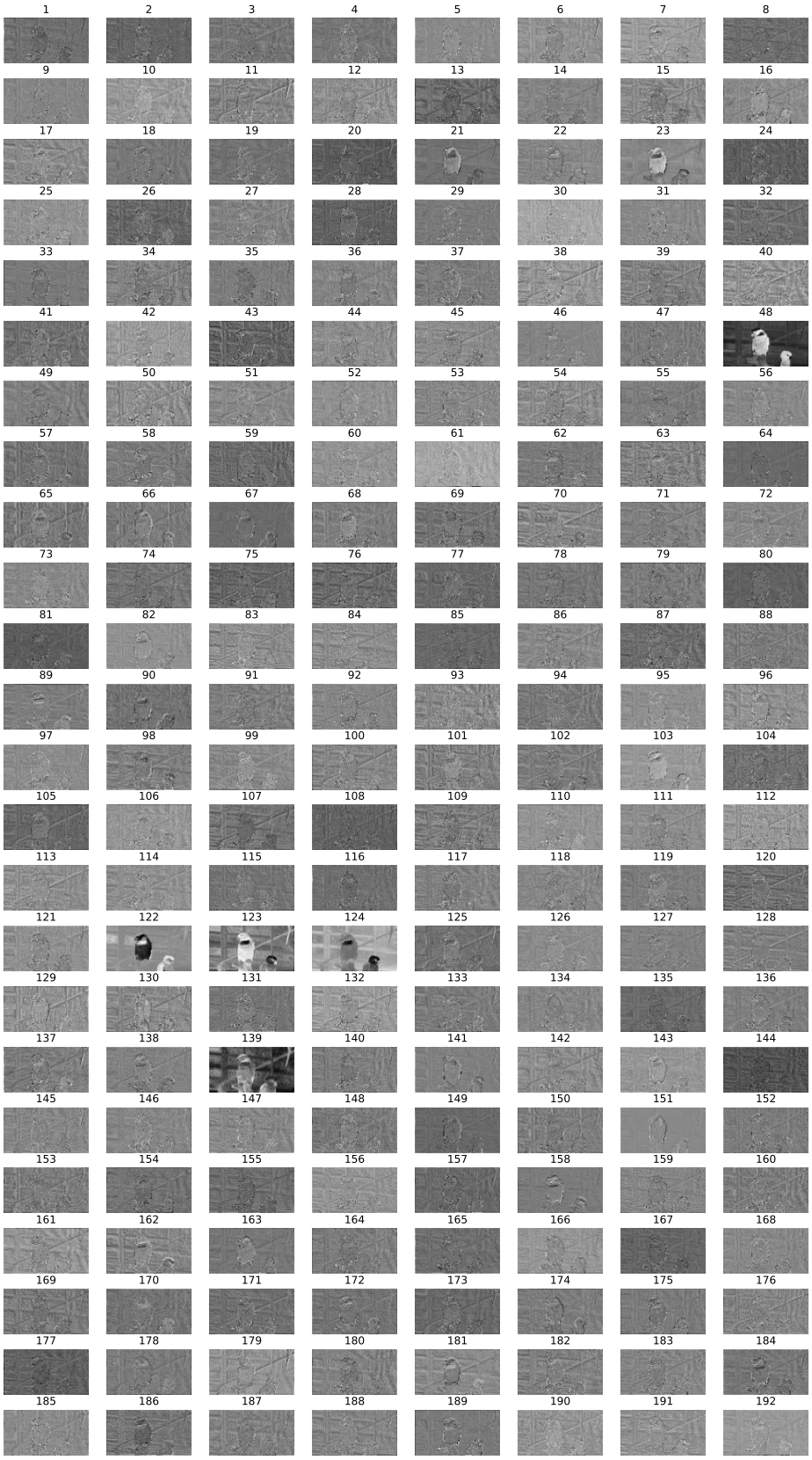


图13. MCL-JCV 数据集中*videoSRC30* 序列的侧信息表示可视化

t.