arXiv:2303.00520v1 [cs.CV] 28 Feb 2023

用于压缩视频质量增强的有效信息引导网络

Xuan Sun、Ziyue Zhang、Guannan Chen 和 Dan Zhu 京
东方科技集团股份有限公司 sunxuan@boe.com.cn

2023年3月2日

摘要

近年来，深度学习方法在压缩视频质量增强任务中展现出了显著优势。现有方法通常将原始视频作为真实标注，并从包含各种伪影的连续帧中提取有效信息。然而，这些方法并未充分利用压缩视频和原始视频中的有效信息来指导压缩视频的质量增强。本文提出了一种独特的**有效信**息引导方案（VIG），通过挖掘压缩视频和原始视频中的有效信息来提升压缩视频的质量。具体而言，我们提出了一种高效的压缩冗余过滤（CRF）网络，以平衡速度与增强效果。通过过滤信息来消除冗余后，CRF能够利用压缩视频中的有效信息重建纹理。此外，我们提出了一种渐进式真实信息引导蒸馏（TGD）策略，无需设计额外的教师模型和蒸馏损失函数。TGD仅将真实标注作为输入，引导模型聚合原始帧之间正确的时空对应关系，从而能够在不增加额外训练成本的情况下显著提升增强效果。大量实验表明，在准确性和效率方面，我们的方法在压缩视频质量增强领域达到了最先进的性能。

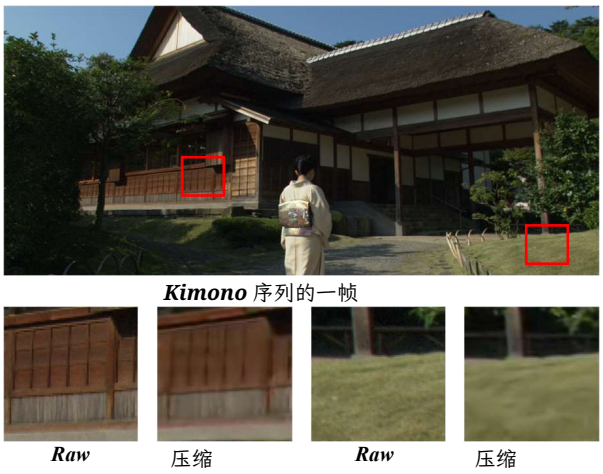


图 1： **压缩图像块和原始图像块的有效信息**。

1 引言

视频数据通常需要经过视频编码，以便高效传输。然而，现有的基于块的编码框架 [17, 22]往往采用不准确的量化和运动补偿技术，导致高频信息丢失并产生大量压缩伪影，如图1所示。压缩视频的主观效果会明显下降，尤其是在低比特率下。我们计算了表 64×64 中 18 个标准测试序列中每个块的像素方差平均值[1] 1。值得注意的是，平均方差

在低延迟（LD）配置下，原始视频的平均值大于压缩视频。换言之，压缩帧所包含的有效信息少于原始帧。此外，低质量的压缩视频会严重影响下游任务（例如分类、检测和分割）。借助终端强大的计算能力，无需增加传输数据量，即可在解码端增强传输后的视频质量。这能够在传输带宽资源有限的情况下，大幅提升解码视频的主观质量。因此，研究压缩视频质量增强（VQE）任务至关重要。

当前大多数工作 [25, 6, 3, 29, 4, 27, 10, 20, 24] 都专注于设计庞大的模型结构，以提升压缩视频的质量。然而，这些工作通常遵循视频增强任务（例如视频超分辨率重建）的设计思路，并未完全根据压缩视频的特性进行设计。一些工作[25, 6, 3]引入高效的网络架构，以减轻计算负担和所需的内存。尽管这些工作实现了速度与效果之间的平衡，但专门设计或复杂的架构可能会给在硬件设备上的实现带来挑战。

知识蒸馏是一种模型压缩方法，可以利用互补知识辅助学生网络的训练过程。它无需改变模型结构，就能显著提升模型性能。PISR[9] 是一种用于图像超分辨率重建的知名知识蒸馏方法。它继承了一般的知识蒸馏方法，需要设计一个庞大的教师模型来辅助学生模型，这显著增加了训练成本。据我们所知，目前还没有用于压缩视频增强的知识蒸馏方法。

本文提出了一种新颖的有效信息引导（VIG）方案，用于完成压缩视频增强任务。VIG 的核心思想是充分利用压缩视频和原始视频中的有效线索来引导视频重建。具体而言，我们首先设计了一种有效的压缩冗余过滤（CRF）网络，通过排除冗余内容来捕获最有效的上下文信息。此外，我们提出了一种反直觉的真实信息引导蒸馏（TGD）策略，通过理解原始视频中的原始像素分布，逐步增强重建效果。总而言之，我们的贡献如下：

- 我们提出一种新颖的有效信息引导（VIG）方案，用于建模压缩视频增强任务的时空依赖关系。
- 我们进行了全面的实验，以探索原始视频与压缩视频之间有效信息流的影响。
- VIG优于当前方法，并在VQE基准数据集上展现出新的最先进的性能。

2 相关工作

冗余过滤方法 过去十年间，许多研究 [3, 25, 6, 29, 13]已证明，利用帧间信息对视频质量增强的结果有着至关重要的影响。从信息提取的角度来看，视频质量增强技术可分为两类：直接提取方法和冗余过滤方法。直接提取方法是指直接聚合原始输入中的信息，例如 STDF[3], 、MFQE2[6] 和 RFDA[29]。相应地，冗余过滤方法是指在主要聚合之前对信息进行压缩的操作。在计算机视觉中，下采样常用于分类、分割和其他高层任务 [16, 30, 5, 14] ，以生成高层表示。然而，由于下采样可能导致大量细节丢失并影响重建质量，因此很少用于图像或视频增强任务。HUPN[18] 提出了一种将下采样操作引入图像超分辨率的框架，展示了在图像增强任务中于主要聚合前压缩信息的可能性。STDF[3] 采用类似 Unet[16] 的模块来预测压缩帧之间的偏移量，但并未将上下采样结构作为整体框架。需要注意的是，与待进行超分辨率处理的图像相比，压缩视频中有更多冗余信息需要过滤。在这项工作中，我们提出 CRF，通过多次

Mode	LD			RAW
QP	32	37	42	-
平均值	1099	1075	1032	1230

表1：**18个标准测试序列的像素方差平均值[1]**。平均方差越小，表示内容越平坦，像素块内保留的纹理越少。

排除冗余内容，更有效地利用时空信息。需要注意的是，与超分辨率图像相比，压缩视频中有更多冗余信息需要过滤。在这项工作中，我们提出 CRF模块，通过多次消除冗余内容，有效利用时空信息。

有效信息应用策略 面向计算机视觉的生成式自监督学习已取得巨大进展。许多工作[7, 23, 19, 11, 12]专注于通过前置任务利用有效信息，以改进下游视觉任务。许多工作[7, 23,19, 11, 12]专注于利用前置任务中的有效信息来改进下游视觉任务。具体而言，MAE[7]和 SimMIM[23]将输入令牌的随机子集替换为特殊的 MASK符号，并借助视觉变换器[5, 14]从受损图像中重建原始图像令牌。具体而言，MAE[7]和SimMIM[23] 将输入令牌的随机子集替换为特殊的MASK符号，旨在利用视觉变换器[5, 14]从受损图像中重建原始图像令牌。随后，VideoMAE[19] 证明，即使采用极高的掩码比例，在视频上仍能取得良好性能。A2MIM[11]在卷积神经网络（CNN）上采用掩码图像建模方法。此外，MixMIM[12] 发现，使用掩码符号会显著导致训练与微调不一致，因此将一幅图像中的掩码令牌替换为另一幅图像中的可见令牌。值得注意的是，上述所有方法都不适用于视频质量增强，因为重建掩码区域与增强压缩区域之间存在明显差异[26]。CutBlur[26] 提出了一种数据增强方法，将低分辨率图像块裁剪下来，并将其粘贴到对应的高分辨率图像上，重新区域。然而，它仅在图像任务中随机进行裁剪和粘贴，并未充分利用视频任务中真实标注的指导。PISR[9] 设计了一种知识蒸馏框架，将真实标注的高分辨率图像作为特权信息，这与我们的工作最为相似。然而，该方法遵循传统的蒸馏方法，并设计了一个额外的教师网络及模仿损失，显著增加了训练成本。受 PISR[9], 启发，我们提出了一种反直觉的知识蒸馏策略：利用原始帧作为输入，同时几乎不增加计算成本。

3 方法

3.1 总体架构

我们分两个阶段训练模型：预训练和微调。此外，VIG 的总体架构可分为两部分：真实信息引导蒸馏（TGD）策略和压缩冗余过滤（CRF）网络。我们假设$\mathbf{X}_i$ 是输入帧，$\mathbf{Y}_i$ 是时间 i的输出帧。总共将使用 $2k+1$ 帧来构成一个视频片段。

这些输入帧表示为$\mathbf{X}_t\{\mathbf{X}_{i-k}$、$\mathbf{X}^{i-k+1}$、...、$\mathbf{X}_i$、=...、$\mathbf{X}^{i+k-1}$、$\mathbf{X}_{i+k}\}$。我们将增强帧表示为$\mathbf{Y}_t$ ={$\mathbf{Y}_{i-k}$、$\mathbf{Y}^{i-k+1}$、...、$\mathbf{Y}_i$、...、$\mathbf{Y}^{i+k-1}$、$\mathbf{Y}_{i+k}$}。增强帧$\mathbf{Y}_t$可以通过以下方式生成

$$\mathbf{Y}_t = V_\theta\left(\mathbf{X}_t\right), \tag{1}$$

其中，$V_\theta(\cdot)$表示VIG的整个处理过程。θ 是VIG的可学习参数。具体而言，以$\mathbf{Y}_i$ 为例，增强过程可形式化表示为

$$\mathbf{Y}_i, \mathbf{F}_{1,i} = V_\theta\{\mathbf{X}_i, \mathbf{X}_{i+1}, \mathbf{X}_{i+2}, \mathbf{X}_{i+3}, \mathbf{F}_{1,i-1}\}, \tag{2}$$

其中，$\mathbf{F}_{1,i-1}$ 表示由RF模块生成的所有增强帧的隐藏特征。

3.2 TGD策略

TGD 的目标是将f_{GT} 中的随机区域剪切粘贴到相应的f_C中，从而生成新的训练样本。我们采用基于图像块的随机掩码

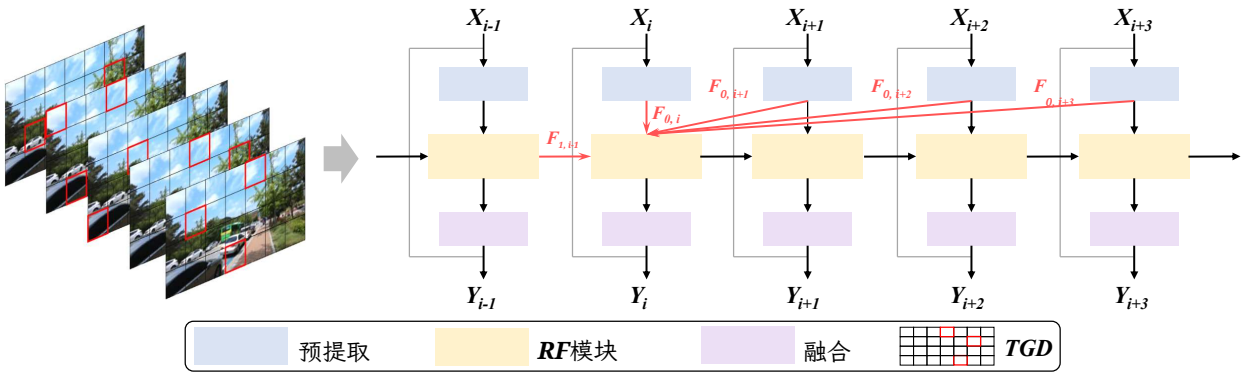


图 2：**VIG 概览**。VIG 通常采用单次传播方法。具体而言，它由两部分组成：真实信息引导蒸馏（TGD）策略和压缩冗余过滤（CRF）网络。对于 TGD，红色方框表示利用对应的压缩图像区域，黑色方框表示使用真实值（GT）的对应区域。对于 CRF，冗余过滤（RF）模块是捕获有效时空信息的关键部分。对于 TGD，红色方框表示利用对应的压缩图像区域，黑色方框表示使用真实值（GT）的对应区域。对于 CRF，冗余过滤（RF）模块对于捕获有效的时空信息至关重要。

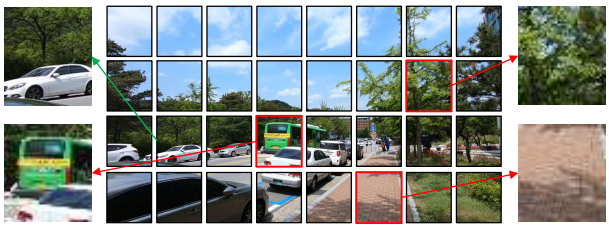


图 3：**TGD策略**。三个红色方框表示添加到真实值（GT）中的压缩图像块；其他方框表示原始图像块。

策略。预训练阶段的输入定义为

$$\mathbf{X}_i = M_{r,w} \odot f_i^{GT} + (1 - M_{r,w}) \odot f_i^{C}, \tag{3}$$

其中，$M_{r,w}$ 是对应 f_i^{GT}的所选区域。r 是 f_i^{GT}在 X_i 中的比例，w 是 X_i中掩码的大小，而 ⊙表示逐元素乘法。图 3 展示了 TGD 的过程。我们根据 w 将 f_i^{GT}切分为 n 个图像块，并基于 r 随机地用 f_i^{C}的对应区域替换若干图像块。我们考虑了不同分辨率阶段的图像块大小，从 2×2到 32×32。

由于替换的标记是在 f_i^{GT} 中随机选取的，TGD 可以通过学习像素之间的局部和全局关系，促使模型获得正则化效果。r 和 w 可以根据需要进行调整。本文使用的 r 设置为 0.9。换言之，TGD 策略几乎完全使用真实值作为输入。由于反向传播的计算需求，我们仅将极小比例的 f_i^{C}加入真实值中。此外，对于四个周期，我们将 w 分别设置为 32、16、8 和 4，以逐步学习从全局到局部的像素关系。我们在每个周期内对 VIG 训练 5 千次迭代。

值得注意的是，我们只在预训练阶段使用 TGD。因此，微调阶段的输入可以视为

$$\mathbf{X}_i = f_i^{C}. \tag{4}$$

3.3 CRF网络

如图2所示，CRF包含一个循环前向网络，由三个部分组成：预提取模块、冗余过滤（RF）模块和融合模块。预提取模块旨在获取输入帧$\mathbf{X}_i$的原始特征。对于原始特征，在第 i个时间戳，令$P_\alpha(\cdot)$ 表示特征

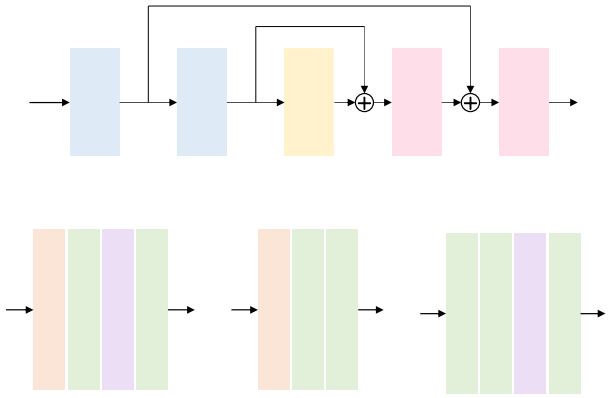


图4：**RF模块**。RF由三个部分组成：时空下采样（TSD）块、时间下采样（TD）块和空间上采样（SU）块。

提取模块，$R_\beta(\cdot)$表示RF模块，$\mathbf{F}_{0,i}$ 表示$P_\alpha(\cdot)$的输出。具体而言，RF模块接收来自当前帧和未来帧的原始特征 $\{\mathbf{X}_i$、.....、$\mathbf{X}_{i+k-1}$、$\mathbf{X}_{i+k}\}$，并获取隐藏特征，以利用过去帧$\{\mathbf{X}_{i-k}$、$\mathbf{X}_{i-k+1}$、.....、$\mathbf{X}_{i-1}\}$的全部信息。随后，我们利用$F_\gamma(\cdot)$作为融合模块来生成输出 $\mathbf{Y}_i$。CRF的运算可形式化表示为

$$\mathbf{F}_{0,i} = P_\alpha\left(\mathbf{X}_i\right), \tag{5}$$

$$\mathbf{F}_{1,i} = R_\beta\{\mathbf{F}_{0,i}, \mathbf{F}_{0,i+1}, \mathbf{F}_{0,i+2}, \mathbf{F}_{0,i+3}, \mathbf{F}_{1,i-1}\}, \tag{6}$$

$$\mathbf{Y}_i = F_\gamma\left(\mathbf{F}_{1,i}\right) + \mathbf{X}_i, \tag{7}$$

其中， α、 β 和 γ 是可学习参数。为了使设计范式简洁，$P_\alpha(\cdot)$和$F_\gamma(\cdot)$均由一个卷积层实现。

RF模块。 如图 4所示，我们采用 TSD 块 $D_\delta^{TSD}(\cdot)$ 和 TD 块 $D_\eta^{TD}(\cdot)$ 来过滤时空冗余，然后利用 SU 块 $U_\mu^{SU}(\cdot)$ 恢复分辨率。RF模块的下采样操作可形式化表示为

$$\mathbf{F}_{TSD1,i} = D_{\delta 1}^{TSD}\{\mathbf{F}_{0,i}, \mathbf{F}_{0,i+1}, \mathbf{F}_{0,i+2}, \mathbf{F}_{0,i+3}, \mathbf{F}_{1,i-1}\}, \tag{8}$$

$$\mathbf{F}_{TSD2,i} = D_{\delta 2}^{TSD}\left(\mathbf{F}_{TSD1,i}\right), \tag{9}$$

$$\mathbf{F}_{TD,i} = D_\eta^{TD}\left(\mathbf{F}_{TSD2,i}\right), \tag{10}$$

其中， $\delta 1$、 $\delta 2$和 η 是可学习参数。RF模块的上采样操作可描述为

$$\mathbf{F}_{SU1,i} = U_{\mu 1}^{SU}\left(\mathbf{F}_{TD,i} + \mathbf{F}_{TSD2,i}\right), \tag{11}$$

$$\mathbf{F}_{SU2,i} = U_{\mu 2}^{SU}\left(\mathbf{F}_{TD,i} + \mathbf{F}_{TSD2,i}\right), \tag{12}$$

其中， $\mu 1$ 和 $\mu 2$ 是可学习参数。

3.4 训练方案

在两个阶段中，我们均采用 Charbonnier 损失$\mathcal{L}$ [2] 训练模型。在第一阶段，使用 TGD 时，仅计算压缩图像块与对应原始图像块之间的 Charbonnier 损失。这样，VIG 能更好地学习真实值中有效信息的流动。预训练过程可形式化表示为

$$\mathcal{L} = \sqrt{((1 - M_{r,w})\,\mathbf{Y}_i - (1 - M_{r,w})\,f_i^{GT})^2 + \epsilon}, \tag{13}$$

其中， ϵ 在此处设为 10^{-6} 。在第二阶段，按照以往的工作，我们使用$\mathcal{L}$ 计算$\mathbf{Y}_i$ 与f_i^{GT}之间的损失，以优化模型。

$$\mathcal{L} = \sqrt{(\mathbf{Y}_i - f_i^{GT})^2 + \epsilon}. \tag{14}$$

请注意，本文的预训练阶段仅需 2 万次迭代，而微调阶段还需 10 万次迭代。

4 实验

为了展示 VIG 的有效性和优越性，我们按照 [6, 3, 29] 在 MFQE 2.0 数据集上进行了细致的实验。此外，我们还进行了大量消融实验，以分析 VIG 各组成部分的重要性，并全面理解 VIG。

类别	序列	AR-CNN	DnCNN	RNAN	MFQE2.0	STDF-R3	Ours
A (2560*1600)	PeopleOnStreet	0.37/0.76	0.54/0.94	0.74/1.30	0.92/1.57	1.18/1.82	**1.30/2.13**
	Traffic	0.27/0.50	0.35/0.64	0.40/0.86	0.59/1.02	0.65/1.04	**0.74/1.24**
B (1920*1080)	Kimono	0.20/0.59	0.27/0.73	0.33/0.98	0.55/1.18	0.77/1.47	**0.92/1.74**
	ParkScene	0.14/0.44	0.17/0.52	0.20/0.77	0.46/1.23	0.54/1.32	0.55/1.44
	Cactus	0.20/0.41	0.28/0.53	0.35/0.76	0.50/1.00	0.70/1.23	**0.73/1.37**
	BQTerrace	0.23/0.43	0.33/0.53	0.42/0.84	0.40/0.67	0.58/0.93	**0.55/0.99**
	篮球驾驶	0.23/0.51	0.33/0.63	0.43/0.94	0.47/0.83	0.66/1.07	**0.77/1.35**
C (832*480)	赛马	0.23/0.49	0.31/0.70	0.35/0.99	0.39/0.80	0.48/1.09	0.50/1.31
	BQMall	0.28/0.69	0.38/0.87	0.45/1.15	0.62/1.20	0.90/1.61	**0.94/1.88**
	PartyScene	0.14/0.52	0.22/0.69	0.30/0.98	0.36/1.18	0.60/1.60	**0.63/2.00**
	篮球训练	0.23/0.48	0.42/0.89	0.50/1.07	0.58/1.20	0.70/1.26	**0.86/1.62**
D (416*240)	赛马	0.26/0.59	0.34/0.80	0.42/1.02	0.59/1.43	0.73/1.75	0.76/2.07
	BQSquare	0.21/0.30	0.30/0.46	0.32/0.63	0.34/0.65	0.91/1.13	0.86/**1.36**
	吹泡泡	0.16/0.46	0.25/0.76	0.31/1.08	0.53/1.70	0.68/1.96	**0.73/2.39**
	篮球传球	0.26/0.63	0.38/0.83	0.46/1.08	0.73/1.55	0.95/1.82	**1.07/2.31**
E (1280*720)	四人	0.40/0.56	0.54/0.73	0.70/0.97	0.73/0.95	0.92/1.07	**1.06/1.33**
	约翰尼	0.24/0.21	0.47/0.54	0.56/0.88	0.60/0.68	0.69/0.73	**0.90/1.09**
	KristenAndSara	0.41/0.47	0.59/0.62	0.63/0.80	0.75/0.85	0.94/0.89	**1.09/1.13**
LD	QP37 平均值	0.25/0.50	0.36/0.69	0.44/0.95	0.56/1.09	0.75/1.32	**0.83/1.60**
	QP32 平均值	0.19/0.17	0.33/0.41	0.41/0.62	0.52/0.68	0.73/0.87	0.77/1.03
	QP27 平均值	0.16/0.09	0.33/0.26	-	0.49/0.42	0.67/0.53	0.73/0.64
	QP22 平均值	0.13/0.04	0.27/0.14	-	0.46/0.27	0.57/0.30	0.63/0.36
LD	总计	0.18/0.20	0.32/0.38	-	0.51/0.62	0.68/0.76	0.74/0.91

表2： **Δ峰值信噪比（dB）/ Δ结构相似性（$\times 10^{-2}$）在4种不同量化参数下测试视频上的定量结果**。表中每个序列的具体数值均在量化参数为37时测得。我们展示了各方法在4种量化参数下的平均值。

4.1 数据集

MFQE 2.0 数据集包含 126 个视频，分辨率范围广泛：SIF (352×240)、CIF (352×288)、NTSC (720×486)、4CIF (704×576)、240p (416×240)、360p (640×360)、480p (832×480)、720p (1280×720)、1080p (1920×1080) 和 WQXGA (2560×1600)。其中 106 个视频用于训练，其余用于验证。测试时，我们采用视频编码联合协作组（JCT-VC）的 18 个标准测试序列 [1] 作为测试集。该测试数据集涵盖不同的场景条件，能够更好地验证开发 HEVC 标准时广泛使用的各种方法的鲁棒性。所有序列均采用 HEVC Low-Delay-P（LDP）配置进行编码，使用 HM 16.20 和四种不同的量化参数（QP），即 22、27、32和 37 [6]。

4.2 实验设置

VIG基于PyTorch框架实现，并使用NVIDIA GeForce RTX 3090显卡进行训练。我们使用Adam优化器 [8]，$\beta_1 = 0.9$，$\beta_2 = 0.99$，并使用余弦退火 [15] 将学习率从3×10^{-4} 衰减至 10^{-4}。我们从原始视频和相应的压缩视频中随机裁剪图像片段，作为训练样本进行训练。图像尺寸为 128 ×128，批量大小为32。我们还采用翻转和旋转作为数据增强策略，以进一步扩充数据集。我们将训练集中的每个视频划分为若干个包含13帧的视频片段。我们仅报告YUV/YCbCr色彩空间中Y通道上的质量增强结果，

方法	Δ峰值信噪比(dB)	Δ结构相似性($\times 10^{-2}$)	速度（FPS）
STDF-R3	0.75	1.32	59.0
VIG	0.83	1.60	60.8

表3：**VIG 的速度**。我们的方法与 STDF-R3 之间的推理速度比较。为进行公平比较，所有方法均使用480p视频，并在 NVIDIA RTX 3090 上进行测试。结果以帧每秒（FPS）表示。

评估。我们使用峰值信噪比（PSNR）和结构相似性指数（SSIM） [21] 来评估VQE方法生成图像的质量。

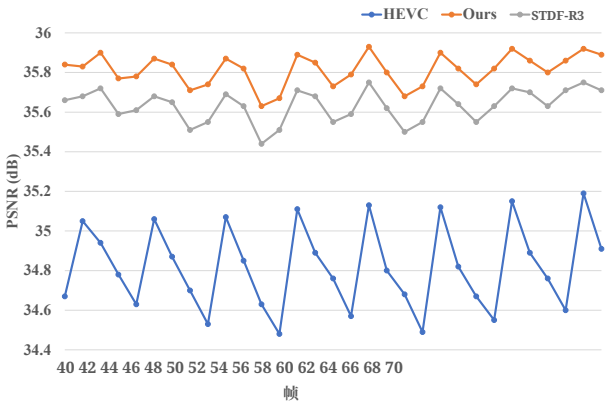


图 5：**量化参数为 QP=37 时，HEVC 基线、STDF-R3 和我们的方法在 FourPeople 序列上的峰值信噪比曲线。**

4.3 与最先进方法的比较

4.3.1 定量评估

我们将 VIG 与以下最先进的视频质量增强模型进行了比较，包括 AR-CNN [4]，DnCNN[27]，RNAN [28],MFQE1.0 [25],MFQE2.0[6], MRDN[13], 和 STDF [3]。定量结果汇总于表2中，速度和性能比较也列于表3中。如表2所示，在四个量化参数下，VIG 在 MFQE2.0 数据集上优于现有最先进方法，这证明了 VIG 的鲁棒性。此外，在大多数序列上，VIG 都显著优于其他方法，展现出更强的鲁棒性和泛化能力。特别是，在量化参数为 37 时，其峰值信噪比比 STDF-R3 [3] 高出 +10.67%/+0.08 dB，比 MRDN [13] 高出 +6.41%/+0.05 dB。需要注意的是，我们的训练数据量仅为 STDF-R3 的 1/3。

此外，我们使用每帧浮点运算次数 来衡量计算复杂度。每帧浮点运算次数 表示每个输出帧所需的浮点运算次数。VIG采用双向传播方法，因此能够同时生成与输入对应的多个帧。因此，VIG的每帧浮点运算次数为56.9 GB，表现出很强的竞争力。VIG的参数权重为1.74 MB。

VIG的效率。我们将 VIG 与 STDF-R3[3] （如表 3所示）进行了比较，后者是速度最快的视频质量增强模型。与 STDF-R3 相比，VIG 以更快的速度运行 +10.67%/+0.08 dB。得益于上下采样的设计，大量计算在低分辨率特征图上完成此外，VIG 的输出是连续的多帧，因此推理效率表现良好。

质量波动。 为展示我们的方法能够改善质量波动，我们绘制了图 5所示的 FourPeople 序列峰值信噪比曲线。可以看出，我们的 VIG 能够有效提升所有帧的质量，并缩小高质量帧与低质量帧之间的差距。

4.3.2 定性评估

定性比较如图6所示。与 AR-CNN 相比，我们的 VIG 能恢复更精细的细节 [28], 和更锐利的边缘，也优于 STDF-R3 [3]。例如，与其他方法相比，只有 VIG 能够成功恢复地板上更明显的纹理、椅子上更清晰的字母以及更精准的窗框。

4.4 消融实验

为更好地理解 VIG，我们对各个关键组件进行消融，并在 MFQE2.0 数据集上评估其性能。

4.4.1 VIG 的组件分析

在本部分中，我们通过多项实验展示所提出的 VIG 的有效性。首先，我们验证各个组件的必要性。随后，我们研究不同 TGD 配置的影响。

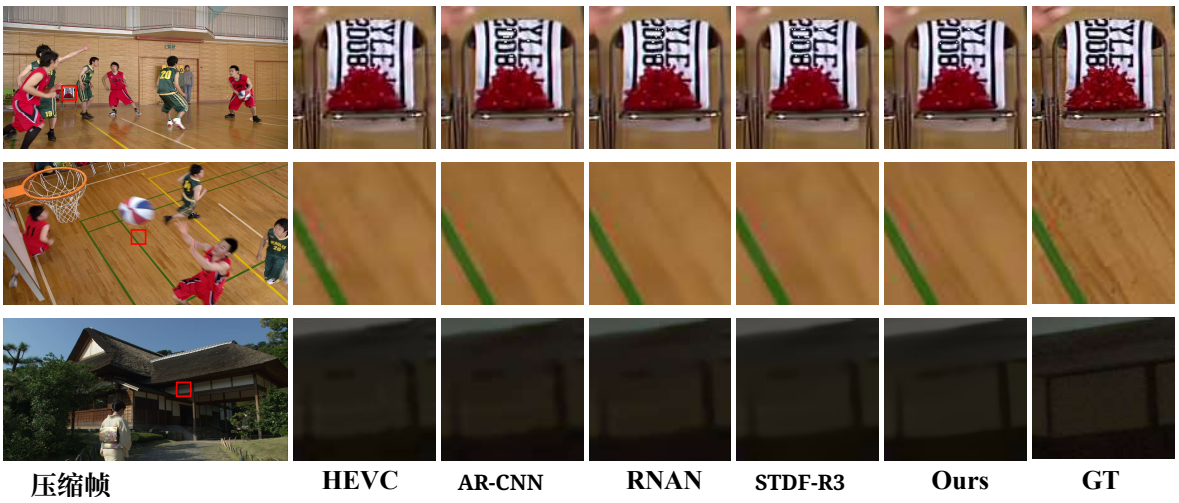


图 6：**MFQE2.0 上的定性比较。**

方法	Base	TF	UD	TGD	Δ峰值信噪比(dB)	Δ结构相似性($\times 10^{-2}$)
(i)	✓				0.62	1.18
(ii)	✓	✓			0.66	1.20
(iii)	✓		✓		0.76	1.45
CRF	✓	✓	✓		0.77	1.48
VIG	✓	✓	✓	✓	0.83	1.60

表4：**组件的影响。** 为了更好地理解 VIG，我们对各个组件进行了消融实验。 ✓表示 VIG 包含当前模块。

组件的重要性。 本节中，我们进行了多项消融实验，如表4 所示，以验证 VIG 的有效性以及各个所提模块的必要性。我们验证时序融合（TF）模块、上下采样（UD）模块和 TGD 策略的必要性。TF 指使用 3D 卷积神经网络层。UD 表示基于图4中像素重排和像素逆重排操作的模块。为便于理解，我们从“基线”开始，它表示由若干普通卷积层构成的简化结构。得益于 UD 模块，每个块只需一个 TF 模块即可实现精确的时序对齐。高层语义特征能够去除冗余，因此更容易进行对齐。由此可见，完整的对比结果表明，每个所提模块都显著提升了性能。纳入所有必要模块后，VIG 达到 +33.87%/+0.21

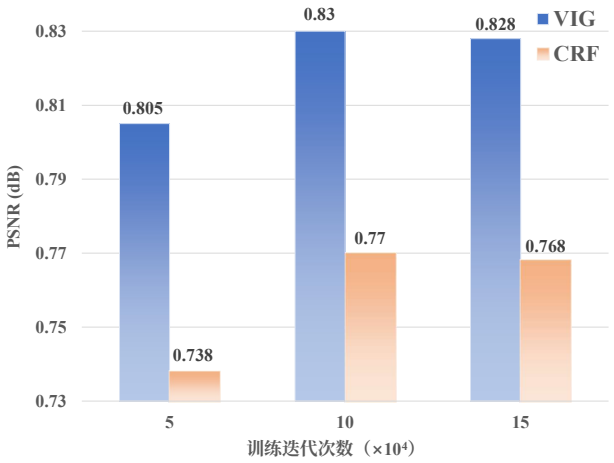


图7： **TGD 的影响。**

比基线高出 Δ dB 的峰值信噪比。值得注意的是，采用上下采样时性能提升 +20.60%/+0.13 dB，采用 TGD 时性能提升 +7.79%/+0.06 dB，这表明有效信息会显著影响视频质量增强。

TGD 加快了收敛速度。图中展示了 7 对 TGD 在各个阶段的性能进行的研究。我们对比实验训练了 12 万次迭代，其中预训练 2 万次迭代，微调 10 万次迭代，

。值得注意的是，每经过 5 万次迭代，采用 TGD 训练的模型都优于未采用 TGD 训练的模型。尽管预训练阶段仅占总训练成本的 16.67%，但模型在整个微调阶段仍表现出稳定的优势。

4.4.2 TGD 掩码策略分析

在本部分中，我们通过大量实验全面探究原始视频中的有效信息如何影响最终性能。首先，我们探索掩码大小对 TGD 的影响。然后，我们研究掩码比例的影响。最后，我们展示多掩码策略如何影响最终性能。

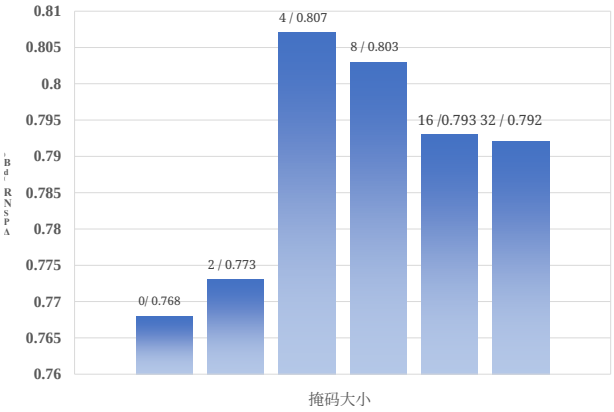


图8： **不同掩码大小的影响**。 在掩码比例设为 0.9 的前提下，研究不同掩码大小的性能。例如，“4/0.807”表示 w 在公式 4×4 中被设为 3 ，且峰值信噪比提升了 0.807 dB。为简化实验，我们对比训练了 17 万次迭代，其中包括预训练阶段的 2 万次迭代和微调阶段的 15 万次迭代。

掩码尺寸。 在这里，我们尝试了解哪种尺寸能够带来更高的准确率。如图8所示，我们通过改变公式 w 中的参数进行实验 3。值得注意的是，所有单尺寸掩码策略都能提升性能。如果使用过大的掩码尺寸，网络可能会偏离对细节恢复的关注，即 16×16 和 32×32。此外，使用不合适的较小尺寸也会损害最终性能，这可能导致模型忽略非局部特征。因此， 4×4 是最合适的尺寸，能够兼顾局部特征和非局部语义信息。

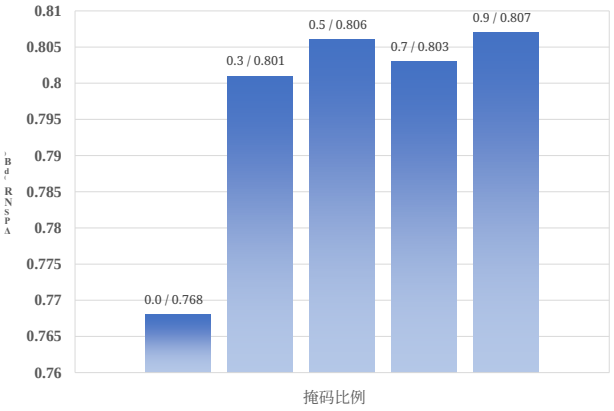


图9： **不同掩码比例的影响**。 在掩码大小设为 4×4 的条件下，研究不同掩码比例的结果。例如，“0.9/0.807”表示 r 在公式3 中被设为 0.9，且峰值信噪比提升了 0.807 dB。与图 8一样，我们还对比训练了五组实验，每组进行 17 万次迭代。

掩码比例。我们还研究了不同掩码比例对效果的影响。图9总结了不同方法在多种掩码比例下微调的准确率。当采用 4 的掩码图块尺寸时，在 0.3 到 0.9 的宽广掩码比例范围内，不同掩码比例下的表现都稳定良好。我们推测，原始像素关系可能已经足够有效。因此，即使使用较低的掩码比例（例如 0.3），网络也会被促使学习原始的长程连接。值得注意的是，最合适的掩码比例是 0.9。换言之，几乎完全使用原始视频作为输入时，模型的表现最佳，这与直觉相悖。

多掩码策略。 如表 5所示，我们将掩码设为 4、比例设为 0.9，作为基础训练策略；这是图中单掩码策略的最佳组合

方法	多掩码	余弦退火	Δ峰值信噪比(dB)	Δ结构相似性($\times 10^{-2}$)
(i)			0.81	1.54
(ii)		✓	0.82	1.58
VIG	✓	✓	0.83	1.60

表5：**多掩码的影响**。为更好地理解 TGD，我们对掩码策略进行了消融实验。✓ 表示 VIG 采用当前设置。

策略中的最佳组合。8。接着，我们使用余弦退火 [15] 将学习率从 3×10^{-4} 衰减至 10^{-4}，并在预训练阶段和微调阶段均采用该方法。最后，使用余弦退火时，我们将掩码设为 32、16、8 和 4。实验结果表明，多掩码训练策略的表现最佳。

5 结论

本文提出了一种增强压缩视频的新方法，其核心思想是充分挖掘压缩视频和原始视频中的有效线索。具体而言，CRF 可通过排除冗余内容来捕获压缩视频中的有效上下文。此外，我们提出了 TGD，使模型能够基于原始视频更好地理解原始像素分布。大量实验表明，我们的方法优于最先进方法。所提出的模块还可以轻松适配现有的多帧方法和视频相关低层任务。

参考文献

[1] F. Bossen 等，“通用测试条件和软件参考配置”，JCTVC-L1100，7，2013 12，1，3，6卷号。 [2] P. Charbonnier、L. Blanc-Feraud、G. Aubert 和 M. Bar-laud，“计算成像的两种确定性半二次正则化算法”，载于《第1届国际图像处理会议论文集》，第2卷。IEEE，1994年，第168–172页。5[3] J. Deng、L. Wang、S. Pu 和 C. Zhuo，“用于压缩视频质量增强的时空可变形卷积”，载于《AAAI人工智能会议论文集》，第34卷，第07期，2020年，第10 696–10 703页。2，5，7

[4] C. Dong、Y. Deng、C. C. Loy 和 X. Tang，“通过深度卷积网络减少压缩伪影”，载于《IEEE国际计算机视觉会议论文集》，2015年，第576–584页。2，7[5] A. Dosovitskiy、L. Beyer、A. Kolesnikov、D. Weissenborn、X. Zhai、T. Unterthiner、M. Dehghani、M. Minderer、G. Heigold、S. Gelly 等，“一张图像值16×16个词：用于大规模图像识别的Transformer”，arXiv预印本 arXiv:2010.11929，2020年。2，3[6] Z. Guan、Q. Xing、M. Xu、R. Yang、T. Liu 和 Z. Wang，“MFQE 2.0：一种用于压缩视频多帧质量增强的新方法”，IEEE模式分析与机器智能汇刊，第43卷，第3期，第949–963页，2019年。2，5，6，7[7] K. He、X. Chen、S. Xie、Y. Li、P. Doll´ar 和 R. Gir-shick，“掩码自编码器是可扩展的视觉学习器”，载于《IEEE/CVF计算机视觉与模式识别会议论文集》，2022年，第16 000–16 009页。3[8] D. P. Kingma 和 J. Ba，“Adam：一种随机优化方法”，arXiv预印本 arXiv:1412.6980，2014年。6[9] W. Lee、J. Lee、D. Kim 和 B. Ham，“利用特权信息进行高效图像超分辨率学习”，载于《欧洲计算机视觉会议》。施普林格，2020年，第465–482页。2，3[10] K. Li、B. Bare 和 B. Yan，“用于压缩图像去块效应的高效深度卷积神经网络模型”，载于《2017年IEEE国际多媒体与博览会会议（ICME）》。IEEE，2017年，第1320–1325页。2

[11] S. Li、D. Wu、F. Wu、Z. Zang、K. Wang、L. Shang、B. Sun、H. Li、S. Li 等人，“架构无关的掩码图像建模——从视觉Transformer回归卷积神经网络”，arXiv 预印本 arXiv:2205.13943，2022年。3

[12] J. Liu, X. Huang, Y. Liu, and H. Li，《MixMIM：用于高效视觉表征学习的混合与掩码图像建模》，arXiv预印本 arXiv:2205.13137，2022年。3[13] J. Liu, M. Zhou, and M. Xiao，《用于压缩视频质量增强的可变形卷积密集网络》，载于 ICASSP 2022-2022 IEEE国际声学、语音与信号处理会议（ICASSP）。IEEE，2022年，第1930–1934页。2，7[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo，《Swin Transformer：采用移位窗口的分层视觉Transformer》，载于 IEEE/CVF国际计算机视觉会议论文集，2021年，第10 012–10022页。2，3

[15]I. Loshchilov and F. Hutter, 《SGDR：带热重启的随机梯度下降》，arXiv预印本 arXiv:1608.03983，2016年。6，10[16] O. Ronneberger, P. Fischer, and 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国际会议。施普林格，2015年，第234–241页。2[17] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, 《高效视频编码（HEVC）标准概述》，IEEE图像处理汇刊：电路与系统视频技术，卷22，第12期，第1649–1668页，2022年。1[18]B. Sun, Y. Zhang, S. Jiang, and Y. Fu, 《用于轻量级图像超分辨率的混合像素反重排网络》，arXiv预印本 arXiv:2203.08921，2022年。2[19]Z. Tong, Y. Song, J. Wang, and L. Wang, 《Video-MAE：掩码自编码器是用于自监督视频预训练的数据高效学习器》，arXiv预印本 arXiv:2203.12602，2022年。3[20]T. Wang, M. Chen, and H. Chao, 《一种从解码端提升HEVC编码效率的新型深度学习方法》，载于2017年数据压缩会议（DCC）。IEEE，2017年，第410–419页。2[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, 《图像质量评估：从误差可见性到结构相似性》，IEEE图像处理汇刊，卷13，第4期，第600–612页，2004年。7[22] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, 《H.264/AVC视频编码标准概述》，IEEE电路与系统视频技术汇刊，卷13，第7期，第560–576页，2003年。1[23] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, 《SimMIM：一种用于掩码图像建模的简单框架》，载于IEEE/CVF计算机视觉与模式识别会议论文集，2022年，第9653–9663页。3[24]R. Yang, M. Xu, T. Liu, Z. Wang, and Z. Guan, 《提升HEVC压缩视频的质量》，IEEE电路与系统视频技术汇刊，卷29，第7期，第2039–2054页，2018年。2[25] R. Yang, M. Xu, Z. Wang, and T. Li, 《压缩视频的多帧质量增强》，载于IEEE计算机视觉与模式识别会议论文集，2018年，第6664–6673页。2，7[26] J. Yoo, N. Ahn, and K.-A. Sohn, 《重新思考图像超分辨率的数据增强：全面分析与新策略》，载于IEEE/CVF计算机视觉与模式识别会议，2020年，第8375–8384页。3[27] K. Zhang、W. Zuo、Y. Chen、D. Meng 和 L. Zhang，“超越高斯去噪器：用于图像去噪的深度卷积神经网络残差学习”，IEEE图像处理汇刊，第26卷，第7期，第3142–3155页，2017年。2，7[28] Y. Zhang、K. Li、K. Li、B. Zhong 和 Y. Fu，“用于图像复原的残差非局部注意力网络”，arXiv预印本 arXiv:1903.10082，2019年。7[29] M. Zhao、Y. Xu 和 S. Zhou，“用于视频压缩伪影消除的递归融合与可变形时空注意力”，载于第29届 ACM国际多媒体会议论文集，2021年，第5646–5654页。2，5[30] Z. Zhou、M. M. R. Siddiquee、N. Tajbakhsh 和 J. Liang，“Unet++：重新设计跳跃连接以利用图像分割中的多尺度特征”，IEEE医学成像汇刊，第39卷，第6期，第1856–1867页，2019年。2