端到端学习式视频流增强流水线：一种生成式人工智能方法

Emanuele Artioli
emanuele.artioli@aau.at 克拉根福阿尔卑斯-亚得里亚大学，奥地利克恩顿州

Farzad Tashtarian
farzad.tashtarian@aau.at 克拉根福阿尔卑斯-亚得里亚大学，奥地利克恩顿州

Christian Timmerer
christian.timmerer@aau.at 克拉根福阿尔卑斯-亚得里亚大学，奥地利克恩顿州

arXiv:2512.14185v1 [cs.MM] 16 Dec 2025

摘要

视频流媒体的主要挑战是在高视频质量与流畅播放之间取得平衡。传统编解码器针对这一权衡进行了充分优化，但由于无法利用上下文信息，它们必须对全部视频数据进行编码并传输至客户端。本文介绍了 ELVIS（**端**到端**学**习式**视**频**流**增强流水线），这是一种端到端架构，结合了服务器端编码优化与客户端生成式图像修复，以移除并重建冗余视频数据。其模块化设计使 ELVIS 能够集成不同的编解码器、图像修复模型和质量指标，从而适应未来的创新。结果表明，当前技术相较于基线基准最多可提升 11 个视频多方法评估融合（VMAF）分，但由于计算需求较高，实时应用仍面临挑战。ELVIS 是将生成式人工智能融入视频流管线的重要基础性一步，能够在不增加带宽需求的情况下提供更高质量的体验。

CCS 概念

• **计算方法** → **目标识别**；**人工智能**；并发算法；**图像压缩**；• **信息系统** → 联机分析处理；**多媒体流媒体**。

关键词

HTTP 自适应流、生成式人工智能、端到端架构、体验质量

1 引言

随着对高质量视频流媒体和存储的需求不断增长，视频压缩方法变得比以往任何时候都更加重要。传统编解码器在保持视觉质量的同时，显著减小了文件大小；每一代产品都比上一代提升约 50% [1]。然而，要满足带宽受限环境日益增长的需求，以及视频内容不断提高的分辨率，仍需进一步改进。一种很有前景的途径是神经编解码器 [2,3]，即将视频压缩为神经网络的权重，

随后由客户端提示其重建帧。除了服务器端压缩效率，视频增强技术也已被用于利用客户端计算能力，例如帧插值和超分辨率 [4, 5, 6, 7]。

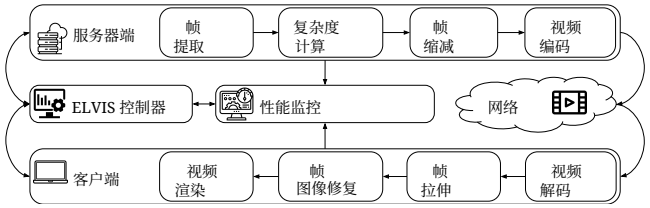


图1：ELVIS 流水线概览。

上述所有技术都面临一个共同的局限：它们只能处理视频的低级特征，例如边缘和块级光流。随着大型生成模型的出现，人工智能如今能够学习并复现高级视频特征、物体，甚至整段视频中长达几秒的内容 [8]。因此，提升视频压缩效果的一条全新且尚待探索的路径，是整合视频修复技术 [9, 10, 11]，对视频进行整体分析，理解其所呈现的内容，并填补缺失或受损的区域。借助机器学习的最新进展，例如注意力机制 [12]，以及使用规模日益庞大、经过精心筛选的数据集进行训练，最先进的修复算法能够学习物体在视频中的典型外观和运动方式，从而能够重建远超以往规模的内容 [9, 10, 11]。

本文的贡献有两方面：(*i*) 提出 ELVIS，这是一种创新方法，将视频修复与编码结合起来，旨在通过消除视频中难以编码、但客户端可使用修复算法重新生成的部分，提高压缩效率，同时避免显著损害观看体验。该方法使编码器能够专注于客户端不易复制的部分，从而在不增加带宽需求的情况下提升视频质量。我们使用多种指标评估该方法的有效性，以确保重建视频达到实际部署所要求的高标准。贡献 (*ii*) 是发布 [1] 一个端到端处理流水线，如图1所示。该流水线在服务器端为视频编码器提供补充，采用最先进的人工智能技术消除冗余，通过视频修复技术增强客户端解码，并协调和监控整个系统所需的资源及其输出质量。该处理流水线采用模块化设计，具有良好的适应性，便于集成新的改进组件，例如新编解码器、修复模型或视频质量指标。因此，所提出的架构不仅能立即带来质量提升，

谨此感谢奥地利联邦数字与经济事务部、国家研究、技术与发展基金会以及克里斯蒂安·多普勒研究协会提供的经费支持。克里斯蒂安·多普勒实验室 ATHENA：https://athena.itec.aau.at/。

本作品采用知识共享署名 4.0 国际许可协议授权。

[1]代码库位于 https://github.com/emanuele-artioli/elvis

还确保随着更优技术的出现，系统能够长期适应。

2 背景与相关工作

视频流媒体面临的挑战传统上包括（*i*）提高服务器端的编码效率，以及（*ii*）在客户端快速适应网络变化。随着时间推移，编解码器不断发展，如今若要进一步大幅缩减视频文件大小，通常需要将编码时间增加十倍 [13]，；与此同时，自适应比特率（ABR）算法已能够熟练地高效利用不断增长的网络带宽。随着强大的人工智能模型和移动处理器问世，如今可以分担计算负荷，从而提升客户端编码视频的质量。

2.1 冗余消除

传统编解码器，例如 AVC/H.264 [15]，以及更新的迭代版本，如 VVC/H.266 [16] 和 AV1 [17]，，通过尽可能减少空间冗余和时间冗余，在压缩效率方面取得了显著提升：它们会搜索相似的像素组，将其编码一次并多次引用。也就是说，尽管这些编解码器取得了成功，但由于其暴力搜索在服务器端计算复杂度高，因此在处理超高分辨率视频内容和实时编码方面仍面临挑战。

NeRV [2] 及其后继版本 HNeRV [3] 等神经编解码器，通过深度神经网络学习视频内容的紧凑表示，使其权重能够发送至客户端，从而将冗余有效地建模为经过优化、用于压缩的隐式特征。这带来了显著的压缩效率提升，但也将计算负担转移到了客户端解码环节，阻碍了它们在实际流媒体场景中的广泛应用。

2.2 视频复原

客户端的主要任务是以最高质量重建编码视频。为此，除了采用高效的解码算法外，还可以应用多种技术来进一步提升视频质量。两种方法尤为突出：(*i*) 帧插值，着重在现有帧之间生成中间帧，以实现更流畅的播放，尤其适用于低帧率内容 [4, 5, 18], (*ii*) 超分辨率，即，通过着重于纹理细化和时间一致性，从低分辨率输入中重建高分辨率帧 [6, 19, 20]。

然而，实时应用所需的计算开销限制了这些技术的实际部署 [21]。这两种技术也被联合使用，以提升一致性 [22]；尽管表现出相当大的潜力，但它们仅限于边缘锐化和局部运动一致性等低级增强，因此在要求更高的场景中效果有限。例如，在处理高度动态或被遮挡的区域时，重影、不自然的纹理或运动等伪影十分常见 [21]。只有能够利用高级语义理解的方法，才能克服这些局限。

视频修复算法旨在利用周围区域的空间和时间线索，解决视频帧中缺失或被遮挡区域的重建难题，已有数十年的发展历史，最初主要依赖纹理合成和基于图像块的方法 [23, 24]，，但难以保持跨帧的时间一致性，导致出现闪烁伪影和不真实的重建结果 [25]。

生成式人工智能的最新进展彻底革新了视频修复，推出了规模庞大且效率足够高的模型，能够将复杂的空间和时间依赖关系纳入其上下文窗口 [26, 27, 10, 9]，，克服了以往技术的局限，并使视频修复成为从有限编码中恢复视频流内容的一条前景广阔的新途径。

2.3 视频质量评估

传统上，视频质量通过均方误差（MSE） [28]，、峰值信噪比（PSNR） [29]，、结构相似性指数（SSIM） [30] 和视频多方法评估融合（VMAF） [31], 等指标进行评估，这些指标用于衡量结构和感知保真度。然而，由于生成式人工智能增强内容具有独特的伪影模式，这些指标往往难以对其进行准确评估。专为图像修复和生成图像设计的学习型感知图像块相似度（LPIPS） [32]，，通过评估高层次感知质量补充了传统指标，因此对于评估人工智能驱动的流程至关重要。

3 系统设计

The ELVIS pipeline 如图 1 所示，该系统由服务器端处理、网络传输和客户端恢复组成，并通过一系列实验对配置参数进行系统性优化，同时由性能监控和编排器监督所有组件。

3.1 ELVIS –服务器端

帧提取。在服务器端，会对输入视频进行分析，并在必要时根据场景变化将其拆分为多个片段，使每个片段都包含连贯的帧，避免出现会对后续步骤产生负面影响的突兀镜头切换。随后，使用 FFmpeg [33], 从每个片段中提取视频帧，再根据实验配置确定的目标分辨率对其进行缩放。此缩放步骤不仅统一了输入尺寸，也便于在不同实验之间进行一致的比较。

复杂度计算。处理流水线的下一步是计算一个度量值，以平衡块的复杂度（*i*）与其感知重要性（*ii*）。使用算法 1 根据重要性度量值对块进行排序，使移除过程从那些编码所需数据量大且视觉上不太重要的块开始，从而实现最佳压缩，并充分发挥图像修复器的优势。

为了评估（*i*），我们采用增强型视频复杂度分析器（EVCA） [34]，，这是一种将每个视频帧按块划分的工具，即划分为由像素组成的正方形区域，其大小由实验配置指定，并分析每个块，生成用于衡量空间复杂度和时间复杂度的两个张量。空间复杂度张量衡量帧内每个块的细节程度和变化性，数值越高表示纹理越复杂或细节越精细。另一方面，时间复杂度评估连续块之间的运动或变化程度，数值越高表示运动越快或越不规则。这些张量的形状为：$(I,\ J,\ N)$，其中 I 是帧中每行的块数，J 是

算法1选择要移除的块

要求： T，$S \in \mathbb{R}^{I \times J \times N}$：时空复杂度张量
要求： w，h：视频的维度
要求： b：块大小
要求： $\alpha \in [0,1]$：空间和时间的加权因子重要性
输入： $\beta \in [0,1]$：平滑因子
输入： $r \in [0,1]$：待移除块的比例
输入： $M \in \mathbb{R}^{I \times J \times N}$：YOLO掩码
确保： $R \in \mathbb{R}^{I \times J \times N}$：待移除块的坐标张量

1：实例化 $C \in \mathbb{R}^{I \times J \times N}$ ▷ 重要性张量
对于 $n = 0$ **到** $N-1$ **执行**
  如果 $n = N-1$ **则**
    $C_{:,:,n} \leftarrow S_{:,:,n}$
  else
    $C_{:,:,n} \leftarrow \alpha \cdot S_{:,:,n} + (1-\alpha) \cdot T_{:,:,n}$ ▷ 公式 1
  结束如果
</style}结束循环
对于 $n = 0$ **到** $N-1$ **执行**
  对于 $i = 0$ **到** I **执行**
    对于 $j = 0$ **到** J **执行**
      如果 $M_{i,j,n} > 0$ **则** $C_{i,j,n} * = -1$ ▷ 反转重要性
      结束如果
    结束循环
  结束循环
  对于 $i = 0$ **到** $I-1$，**执行**
    实例化 $row \in \mathbb{R}^{J}$
    $row \leftarrow C_{i,:,n}$
    如果 $C_{i,:,n-1} \neq None$，**则**
      $row \leftarrow \beta \cdot row + (1-\beta) \cdot C_{i,:,n-1}$ ▷ 方程 2
    结束如果
    按降序排序
    $row_{r \cdot J:} \leftarrow None$ ▷ 选择 r*J 个最不重要的块
    $R_{i,:,n} \leftarrow row$
  结束循环
结束循环
返回 R

每列的块数，N是帧数。三维张量可以沿帧维度切片，得到两个二维矩阵$S_{i,\ j}$和$T_{i,\ j}$，如图 2 的步骤（1-2）所示。每个块的复杂度通过加权公式汇总其空间复杂度和时间复杂度来确定，如公式 1 所示：

$$C_{i,j} = \alpha \cdot S_{i,j} + (1-\alpha) \cdot T_{i,j} \tag{1}$$

其中，α是一个配置参数，用于控制空间因素和时间因素的相对影响。这种灵活性使 ELVIS 能够针对每个实验案例确定空间复杂度与时间复杂度的最佳影响权重。帧的时间复杂度取决于后续帧的信息，因此最后一帧仅取决于空间复杂度。

为了评估（*ii*），我们考虑到，与前景中的物体相比，视频的背景受到的关注要少得多。因此，我们利用 You Only Look Once (YOLO) [35] 对每个帧中的物体形状进行分割，并生成用于标示哪些块属于前景的掩码，如图 2 的步骤（3）所示。

算法 2 帧缩减

要求： c：颜色通道（3）
输入： $F \in \mathbb{R}^{I \times J \times c}$: 输入帧
输入： n: 帧编号
输入： b: 块大小
输入： $R \in \mathbb{R}^{I \times J \times N}$: 要移除的块的坐标张量
确保： $F' \in \mathbb{R}^{I \times J - r \cdot J}$: 缩小帧
确保： $O \in \mathbb{R}^{I/r \times J/r \times N}$: 二值掩码张量

实例化 $L \in \mathbb{R}^{I/b \times J/b \times b \times b \times c}$
对于 $i = 0$ **到** I/b **执行**
  对于 $j = 0$ **到** J/b **执行**
    $L_{i,j} \leftarrow F_{i \cdot b:(i+1) \cdot b,\ j \cdot b:(j+1) \cdot b,\ :}$
  结束循环
</style}结束循环
对 $i = 0$ **到** I/b **执行**
  对 $j = 0$ **到** J/b **执行**
    如果 $j \notin R_{i,j,n}$ **则**
      $F'_{i,j} \leftarrow F_{i,j}$ ▷ 用帧块填充缩小帧
      $O_{i,j,n} \leftarrow 0$ ▷ 掩码中指示该块已填充的信号
    else
      $O_{i,j,n} \leftarrow 1$ ▷ 指示该块未填充的信号
    结束 if
  结束 for
</style}结束循环
</style}返回 $F'_{i,j}$
返回$O_{i,j,n}$

这些前景块的复杂度值乘以 −1，以降低其相对于背景块被移除的优先级。此步骤还确保，如果算法已无背景可移除，就会从复杂度最低且更易进行图像修复的前景块开始移除。为了进一步优化块移除决策，最终的重要性度量使用公式 2 所示的公式，在帧之间对复杂度进行平滑处理：

$$I_{i,j} = \beta \cdot C_{i,j} + (1-\beta) \cdot C_{i-1,j} \tag{2}$$

其中，β是另一个可调参数，用于控制时间平滑的程度。较低的值会促使连续帧采用相似的移除决策，从而避免快速运动场景中出现闪烁；而当 β接近 1 时，则会忽略过去的决策，确保在慢速运动场景中，图像修复模型每隔几帧就能看到每个块。

帧缩减。确定应以何种顺序移除块，以优化视频文件大小与重建质量之间的权衡后，便可以收缩帧，即通过移除上一步识别出的块来降低帧分辨率。当前编解码器要求视频帧具有固定的、

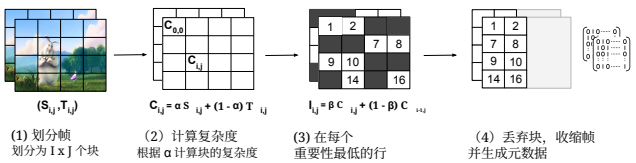


图 2：复杂度计算和帧缩减。

矩形形状；因此，需要以符合这一要求的方式移除块。有两种可行的方法：一种是彼此独立地移除块，然后重新排列这些块，使结果形成分辨率完全相同的矩形帧。例如，移除块较少的帧可以向受块移除影响更严重的帧“借出”块。然而，将块从一帧移动到另一帧，或在同一帧内从一个位置移动到另一个位置，都会破坏视频连续性，从而降低编解码器的效率；编解码器随后需要使用更多信息来弥补这种不连续性，反而违背了这一方法的目的。第二种方法是用占位符替换移除的块，例如黑色块；黑色块看似不包含任何信息，因此理论上可以在几乎不产生信息开销的情况下进行编码，但现代编解码器并非如此实现。实际上，用完整的黑色块替换图像块几乎不会带来编码效率方面的提升，除非该黑色块在短时间内连续重复多次，从而可以被引用。ELVIS 通过逐行迭代收缩帧来解决这一问题：每次移除重要性最高的块，直到帧按照配置参数中定义的幅度完成收缩。因此，在每次迭代中，每个帧都会变窄一个块，并且可以以保持编解码器所需形状和视觉一致性的方式重新拼合，如图 2 的步骤 (4) 所示。收缩帧所需的全部信息都包含在重要性矩阵中，因此可以通过并行化大幅加快收缩过程，即使用 c oncurrent [36] Python 软件包中的 ProcessPoolExecutor，将每个帧发送至不同的线程，由相应线程执行算法 2。

算法3 帧拉伸

需要： c: 颜色通道 (3)
需要： $F' \in \mathbb{R}^{I \times J-r.J}$: 缩小帧
需要： n: 帧编号
需要： b: 块大小
要求： $O \in \mathbb{R}^{I/r \times J/r \times N}$：二值掩码张量
确保： $F \in \mathbb{R}^{I \times J \times c}$：拉伸帧

对于 $i = 0$ **到** I/b **执行**
  对于 $j = 0$ **到** J/b **执行**
    如果 $O_{i,\ j,\ n} = 0$ **则**
      $F_{i,\ j} \leftarrow F'_{i,\ j}$ ▷ 用缩小块填充拉伸帧
    else
      $F_{i,j} \leftarrow 0$ ▷ 用黑色块填充拉伸帧
    结束 if
  结束 for
结束 for

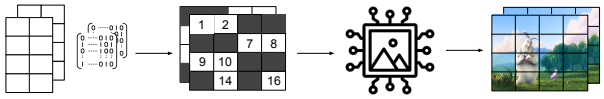


图3：帧拉伸与图像修复。

视频编码。缩小帧后，可以使用任何编解码器对其进行编码，这体现了 ELVIS 的适应性；而且根据缩小程度，其所需比特率低于完整帧所需的比特率。为了准确评估缩小后产生的大量分辨率所对应的最佳比特率，我们利用 ARTEMIS 的 mega-manifest [37]。本文展示的实验涵盖了最常见的传统编码器 AVC，以及较新一代的学习型编解码器 HNeRV。我们认为这些编码器足以展示 ELVIS 的潜力与多样性，

不过，我们未来会着手测试新的组件。最终生成最小元数据，如图 2 的步骤（5）所示，用于告知客户端已移除块的坐标，以逆转这一过程，并将缩小后的视频恢复至全分辨率。因此，需要将此类元数据的大小纳入考量。为尽量减小元数据大小带来的影响，我们将其保存为二维矩阵，其中每一行由单帧中的所有行拼接而成，并将每帧的元数据保存到一个压缩的 NumPy 数组（.npz [38] format）中。除了能将该文件的大小控制在编码视频的 5% 以内（分辨率越高，占比越低）之外，这种压缩方式也非常稳健，客户端解压并使用起来也很快。对元数据压缩进行更精确的研究或许还能进一步减小其影响，但这超出了本研究的范围。

3.2 ELVIS – 客户端

在客户端，框架会解码接收到的缩小文件，根据元数据将帧恢复至全分辨率（在被移除的块所在位置添加临时黑色块），并将视频传递给图像修复模型以恢复细节，如图3所示。

解码。客户端收到编码视频和元数据后，首先开始解码缩小帧。如果视频使用 AVC 等传统编解码器编码，FFmpeg 会解析视频并直接提取帧。对于 HNeRV 等神经编解码器，客户端会运行 HNeRV解码模型，输出经过填充的帧。随后，FFmpeg 会移除编码过程中引入的填充，使帧的分辨率保持一致，无论使用何种编解码器。这确保 ELVIS 不受编码方式限制，能够顺利进入后续步骤。

帧拉伸。 解码后，根据元数据标记的位置重新插入黑色占位块，将缩小帧拉伸回原始分辨率，具体过程如算法 3 所示。对于每一帧，算法都会遍历对应的掩码，以确定下一个块应取自缩小帧，还是需要进行图像修复。对于标记为图像修复的块，算法会填入黑色占位块，从而生成可用于重建的拉伸帧。与缩小过程一样，拉伸过程也经过了高效的并行化处理。

帧修复。准备好拉伸帧后，下一步是使用生成式图像修复模型恢复缺失内容。ELVIS 支持任何遵循标准输入输出格式的模型，即以视频帧和掩码为输入，并以修复后的帧为输出。本研究测试了两种最先进的图像修复模型，即 ProPainter [9] 和 E2FGVI [10],，但只需进行少量配置更改，处理流水线便可兼容其他模型。图像修复模型利用周围视频内容的时空上下文以及自身对场景的理解，重建缺失的块，

表 1：ELVIS 配置参数及其可能的实例化。

参数	值
视频	DAVIS 数据集
块大小	8、16、32、64
移除比例	0.25、0.50、0.75
α	0, 0.25, 0.50, 0.75, 1
β	0、0.25、0.50、0.75、1
高度	360、540、720、900
宽度	640, 960, 1280, 1600
编解码器	AVC、HNeRV
视频修复工具	ProPainter、E2FGVI

上下文。随后，使用 FFmpeg 将这些帧无损编码为视频，完成重建过程。

4 性能评估

ELVIS架构的一个重要组成部分是监控各组件的运行情况和结果，以识别瓶颈并寻求改进。监控范围涵盖服务器端和客户端组件，记录处理流水线各阶段所用的时间，并评估最终输出的质量。

基准比较。 为评估处理流水线的有效性，我们开展了一系列实验，将修复后视频与基准视频进行比较。两者采用相同的分辨率和文件大小，但基准视频未经客户端增强。比较涵盖了传统视频评估指标（均方误差、峰值信噪比、结构相似性指数、视频多方法评估融合），以及专为图像修复内容设计的指标（学习型感知图像块相似度）。这样，ELVIS便能量化图像修复过程带来的益处，并确定实验的参数配置是否带来了显著改进。

实验设置。 我们开展了一系列实验来评估 ELVIS 的性能：在一台运行 Ubuntu 20.04 LTS 的服务器上运行处理流水线，该服务器配备一颗拥有 64 个核心、运行频率为 2.30GHz 的 Intel Xeon Gold 5218 CPU，以及一块拥有 48GB 显存的 NVIDIA Quadro RTX 8000 图形处理器。实验旨在测试各个主要组件：编码、图像修复和质量测量，并采用了多种参数配置，如表 1 所示。我们处理了 DAVIS 数据集 [39],中的视频序列。该数据集以场景构成和摄像机运动的多样性著称，其分辨率（480p 和 1080p）也符合视频流媒体的要求。鉴于这项任务计算量巨大，我们最终在 10 段不同的视频上完成了 200 次有效实验，并将其视为一个可用于得出有价值见解的多样化数据集。

4.1 结果

相关性分析。图 4 展示了 ELVIS 的参数与关键质量指标（包括编码和图像修复时间）之间的皮尔逊相关系数。该分析有助于深入了解参数配置如何影响最终的视频质量。正如预期，比特率和块移除比例的相关性最强。较高的比特率可提升视频质量，但会增加计算需求；而较高的块移除比例会使精确图像修复更加困难，从而降低质量。有趣的是，块移除比例与视频多方法评估融合和学习型感知图像块相似度等现代指标之间的相关性有限，但深入探究这一问题超出了本文的范围。

参数影响。图 5 展示了 α（空间与时间复杂度权重参数）的变化如何影响视频多方法评估融合得分。我们观察到一个有趣的趋势：α 趋于极端的取值时，会在块移除决策中更侧重空间复杂度或时间复杂度，往往能获得更好的视频质量。对于 β 也观察到了类似趋势；该参数控制时间平滑和块大小。

基准比较。 图 6 比较了所有实验中基准视频和修复后视频的质量指标。请注意，均方误差和学习型感知图像块相似度越低越好，而峰值信噪比、结构相似性指数和视频多方法评估融合越高越好。在大多数情况下，图像修复能够提升视频多方法评估融合和学习型感知图像块相似度得分，这些指标与人类感知最为契合，最高分别提升 10 个视频多方法评估融合点和 0.2 个学习型感知图像块相似度点。此外，整体质量往往相对较低，主要原因是未压缩输入视频与基准视频和修复后视频之间的分辨率差异显著。不过，有些实验中质量有所下降；在这种情况下，ELVIS 会传输基准视频，确保图像修复不会降低观众体验到的视频质量。

视频特定改进。 图 7 按视频标题展示了视频多方法评估融合的提升情况。平均而言，图像修复可使视频多方法评估融合提升 2–3 点，部分视频的提升幅度高达 11 点，带来非常明显的改善。

时间复杂度。图 8 展示了 ELVIS 编码和图像修复阶段的时间复杂度。生成式人工智能模型虽然功能强大，但目前尚未针对效率进行优化，且需要大量

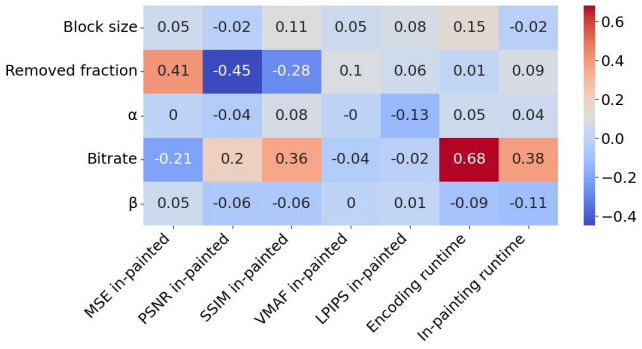


图 4：ELVIS 各参数对质量指标的影响。

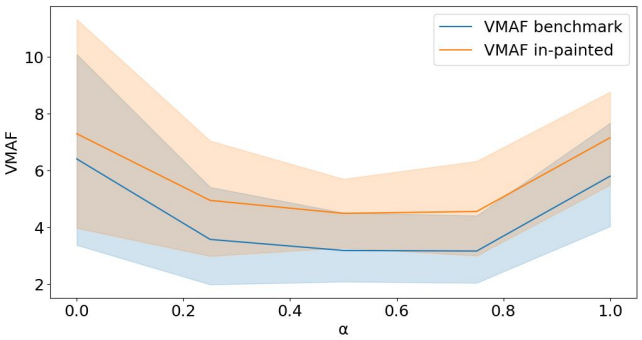


图 5：α 上的视频多方法评估融合分布。

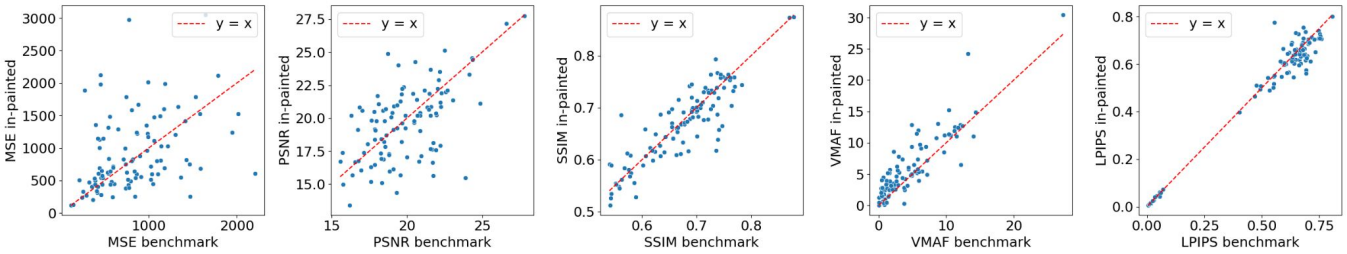


图 6：基准与修复后指标的比较。

图形处理器算力，因此无法处理更高分辨率的视频。实验中，图像修复所需时间往往是视频时长的 10 倍，导致在当前消费级系统上无法实现实时应用。此外，使用 HNeRV 进行编码的速度比图像修复慢一个数量级，这可能是 HNeRV 复杂度较高，加上 ELVIS 内部配置欠佳所致，而配置欠佳又是因为缺乏文档。遗憾的是，我们未能联系到作者寻求帮助。这凸显了图像修复和神经编码都亟待改进，以支持实时流媒体。

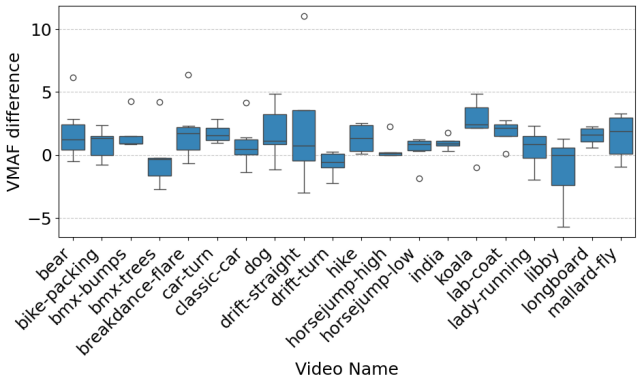


图 7：按视频名称划分的图像修复相较于基准的 VMAF 提升。

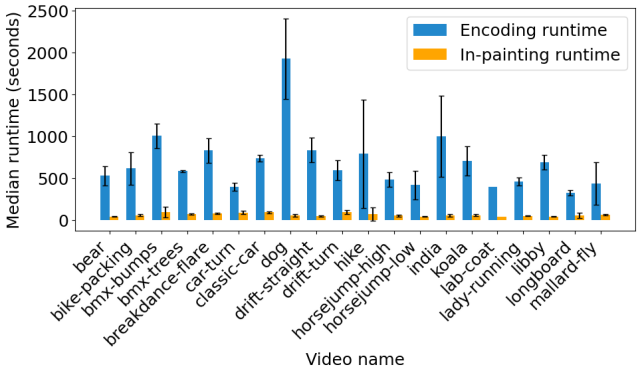


图 8：按视频名称划分的编码和图像修复时间复杂度。

5 结论与未来工作

本文探讨了在视频流媒体领域尚未得到探索的生成式人工智能模型如何在不增加比特率的情况下显著提升视频质量。我们介绍了 ELVIS，这是一种端到端架构，旨在将通用图像修复模型用于流媒体任务，并提供一个模块化框架，以便随着更强大、更高效的人工智能技术不断涌现而将其集成其中。在实际部署中，ELVIS 会利用服务器空闲时间，找出能够通过图像修复优化质量的参数配置。收到客户端请求后，服务器会选择性能最佳的配置进行传输。通过利用服务器空闲资源，并确保在必要时回退到未修复视频，ELVIS 保证其实现不会导致质量下降。

尽管本研究奠定了坚实的基础，但仍有若干未来研究方向值得探索。首先，提升生成式图像修复模型的效率，对于降低计算需求并支持实时应用至关重要。此外，了解某些配置与质量指标之间相关性有限的原因，或可推动引入不同的参数和新的帧缩减策略。改进元数据压缩技术可进一步减少传输开销；同时，调整 ELVIS 以更好地支持 HNeRV 等神经编解码器，或引入新的编解码器，也有望解锁新的能力。最后，深入探索客户端集成，包括为资源受限设备开发轻量级图像修复模型，将显著拓展 ELVIS 的实际适用范围。

随着生成式人工智能领域不断发展，ELVIS 有望成为未来视频流媒体系统的关键组成部分。通过将先进的客户端增强与服务器端优化相结合，它有望在不增加带宽消耗的情况下改善用户体验，标志着视频流媒体技术向前迈出了重要一步。

参考文献

[1] Zhuoran Li、Zhengfang Duanmu、Wentao Liu 和 Zhou Wang。2019年。AVC、HEVC、VP9、AVS2 还是 AV1？——对 4K 视频领域先进视频编码器的比较研究，2019年8月，162–173。ISBN：978-3-030-27201-2。DOI：10.1007/978-3-030-27202-9。_[2] Hao Chen、Bo He、Hanyu Wang、Yixuan Ren、Ser Nam Lim 和 Abhinav Shrivastava。2021年。NERV：视频神经表征。神经信息处理系统进展，34，21557–21568。[3] Hao Chen、Matthew Gwilliam、Ser-Nam Lim 和 Abhinav Shrivastava。2023年。HNeRV：视频混合神经表征。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，10270–10279。

[4] Simon Niklaus、Long Mai 和 Feng Liu。2017年。基于自适应可分离卷积的视频帧插值。载于*IEEE*国际计算机视觉会议论文集，261–270。[5] Wenbo Bao、Wei-Sheng Lai、Chao Ma、Xiaoyun Zhang、Zhiyong Gao 和 Ming-Hsuan Yang。2019年。深度感知视频帧插值。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，3703–3712。[6] Xintao Wang、Ke Yu、Shixiang Wu、Jinjin Gu、Yihao Liu、Chao Dong、Yu Qiao 和 Chen Change Loy。2018年。ESRGAN：增强型超分辨率生成对抗网络。载于欧洲计算机视觉会议（*ECCV*）研讨会论文集，0–1。[7] Christian Ledig 等。2017年。利用生成对抗网络实现照片级真实感单图像超分辨率。载于*IEEE*计算机视觉与模式识别会议论文集，4681–4690。[8] OpenAI。2021年。DALL·E：根据文本描述生成图像。https://openai.com/dall-e。访问日期：2024年11月20日。（2021年）。[9] Shangchen Zhou、Chongyi Li、Kelvin C.K. Chan 和 Chen Change。2023年。ProPainter：改进视频修复中的传播与Transformer。载于*2023 IEEE/CVF*国际计算机视觉会议（*ICCV*），10443–10452。DOI：10.1109/ICCV51070.2023.00961。[10] Zhen Li、Cheng-Ze Lu、Jianhua Qin、Chun-Le Guo 和 Ming-Ming Cheng。2022年。迈向光流引导视频修复的端到端框架。载于*2022 IEEE/CVF*计算机视觉与模式识别会议（*CVPR*），17541–17550。DOI：10.1109/CVPR52688.2022.01704。[11] Fei-Yue Wang、Qinghai Miao、Lingxi Li、Qinghua Ni、Xuan Li、Juanjuan Li、Lili Fan、Yonglin Tian 和 Qing-Long Han。2024年。Sora何时展现：道迈向想象智能与场景工程的开端。*IEEE/CAA*自动化学报，11，4，809–815。DOI：10.1109/JAS.2024.124383。[12] A Vaswani。2017年。注意力就是一切。神经信息处理系统进展。[13] Georgios Minopoulos、Vasileios A Memos、Konstantinos E Psannis 和 Yutaka Ishibashi。2020年。实时传输中视频编解码器性能比较。载于*2020*年第*2*届国际计算机通信与互联网会议（*ICCCI*）。IEEE，110–114。[14] T. Wiegand、G.J. Sullivan、G. Bjontegaard 和 A. Luthra。2003年。H.264/AVC视频编码标准概述。*IEEE*视频技术电路与系统汇刊，13，7，560–576。DOI：10.1109/TCSVT.2003.815165。[15] Gary J. Sullivan、Jens-Rainer Ohm、Woo-Jin Han 和 Thomas Wiegand。2012年。高效视频编码（HEVC）标准概述。*IEEE*视频技术电路与系统汇刊，22，12，1649–1668。DOI：10.1109/TCSVT.2012.2221191。[16] Benjamin Bross、Ye-Kui Wang、Yan Ye、Shan Liu、Jianle Chen、Gary J. Sullivan 和 Jens-Rainer Ohm。2021年。多功能视频编码（VVC）标准及其应用概述。*IEEE*视频技术电路与系统汇刊，31，10，3736–3764。DOI：10.1109/TCSVT.2021.3101953。[17] Jingning Han 等。2021年。AV1技术概述。*IEEE*会刊，109，9，1435–1462。DOI：10.1109/JPROC.2021.3058584。[18] Zhenxing Shi、Xiangyu Xu、Xiaoming Liu 和 Jianjun Chen。2022年。视频帧插值Transformer。载于*IEEE/CVF*计算机视觉与模式识别会议论文集（*CVPR*）。IEEE，10474–10483。https://openaccess.thecvf.com/content/CVPR2022/papers/Shi_视频_帧_插值_Transformer_CVPR_2022_论文.pdf。[19] Jiezhang Cao、Yawei Li、Kai Zhang 和 Luc Van Gool。2021年。视频超分辨率Transformer。*arXiv*预印本 *arXiv:2106.06847*。[20] Yan Wang、Yusen Li、Gang Wang 和 Xiaoguang Liu。2024年。用于单图像超分辨率的多尺度注意力网络。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，5950–5960。

[21] Hongying Liu、Zhubo Ruan、Peng Zhao、Chao Dong、Fanhua Shang、Yuanyuan Liu、Linlin Yang 和 Radu Timofte。2022年。基于深度学习的视频超分辨率：全面综述。*Artificial Intelligence Review*，55，8，5981 6035–。[22] Soo Ye Kim、Jihyong Oh 和 Munchurl Kim。2020年。Fisr：结合多尺度时间损失的深度联合帧插值与超分辨率。载于*AAAI*人工智能会议论文集第07期。第34卷，11278–11286。[23] Antonio Criminisi、Patrick Perez 和 Kentaro Toyama。2003年。通过基于样例的图像修复移除物体。载于*2003*年*IEEE*计算机学会计算机视觉与模式识别会议论文集第2卷。IEEE，II–II。[24] Kedar A Patwardhan、Guillermo Sapiro 和 Marcelo Bertalmio。2005年。对遮挡物体与被遮挡物体进行视频修复。载于*IEEE 2005*年国际图像处理会议。第2卷。IEEE，II–69。[25] Yonatan Wexler、Eli Shechtman 和 Michal Irani。2007年。视频的时空补全。*IEEE*模式分析与机器智能汇刊，29，3，463 476–。[26] Yanhong Zeng、Jianlong Fu 和 Hongyang Chao。2020年。学习用于视频修复的联合空间—时间变换。载于计算机视觉—*ECCV 2020*：第*16*届欧洲计算机视觉会议，格拉斯哥，英国，*2020*年*8*月*23*日至*28*日，论文集，第*XVI*部分（第*16*卷）。施普林格，528–543。[27] Bohai Gu、Yongsheng Yu、Heng Fan 和 Libo Zhang。2023年。用于视频修复的流引导扩散。*arXiv*预印本 *arXiv:2311.15368*。[28] Ian Goodfellow、Yoshua Bengio 和 Aaron Courville。2016年。深度学习。麻省理工学院出版社。[29] Quan Huynh-Thu 和 Mohammed Ghanbari。2008年。峰值信噪比在图像/视频质量评估中的适用范围。*Electronics Letters*，44，13，800–801。[30] Zhou Wang、Alan C Bovik、Hamid R Sheikh 和 Eero P Simoncelli。2004年。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，13，4，600–612。[31] 网飞。2016年至今。VMAF开发工具包。网址：https://github.com/Netflix/vmaf，访问日期：2024年11月18日。（2016年至今）。[32] Richard Zhang、Phillip Isola、Alexei A Efros、Eli Shechtman 和 Oliver Wang。2018年。深度特征作为感知度量的惊人有效性。载于*IEEE*计算机视觉与模式识别会议论文集。[33] Fabrice Bellard 和 FFmpeg开发者。2000年至今。FFmpeg。网址：https://ffmpeg.org，访问日期：2024年11月18日。（2000年至今）。[34] Hadi Amirpour、Mohammad Ghasempour、Lingfeng Qu、Wassim Hamidouche 和 Christian Timmerer。2024年。EVCA：增强型视频复杂度分析器。载于第*15*届*ACM*多媒体系统会议论文集（MMSys '24）。计算机协会，纽约，美国。[35] J Redmon。2016年。You Only Look Once：统一的实时目标检测。载于*IEEE*计算机视觉与模式识别会议论文集。[36] Python软件基金会。2023年。*Python*标准库：*concurrent.futures*。网址：https://docs.python.org/3/library/concurrent.futures.html，访问日期：2024年11月18日。[37] Farzad Tashtarian、Abdelhak Bentaleb、Hadi Amirpour、SergeyGorinsky、Junchen Jiang、Hermann Hellwagner 和 Christian Timmerer。2024年。ARTEMIS：面向直播视频流的自适应比特率梯级优化。载于*USENIX*网络系统设计与实现研讨会论文集（*NSDI*）。USENIX协会。[38] Charles R Harris 等。2020年。585，7825，357 362。使用 NumPy 进行数组编程。*Nature*，–。[39] Jordi Pont-Tuset、Federico Perazzi、Sergi Caelles、Pablo Arbeláez、Alexander Sorkine-Hornung 和 Luc Van Gool。2017年。2017年DAVIS视频目标分割挑战赛。*arXiv:1704.00675*。