视频帧插值变换器

Zhihao Shi[*1] Xiangyu Xu[*†2] Xiaohong Liu[3] Jun Chen[1] Ming-Hsuan Yang[4,5,6]

[1]麦克马斯特大学 [2]南洋理工大学 [3]上海交通大学
[4]加利福尼亚大学默塞德分校 [5]延世大学 [6]谷歌研究

摘要

现有的视频插值方法严重依赖深度卷积神经网络,因此存在固有的局限性,如内容无关的核权重和受限的接受域。为了解决这些问题,我们提出了一种基于 Transformer 的视频插值框架,该框架允许内容感知聚合权重,并通过自注意力操作考虑长距离依赖关系。为了避免全局自注意力的高计算成本,我们将局部注意力的概念引入视频插值中,并将其扩展到时空域。此外,我们还提出了一种空间-时间分离策略以节省内存使用,这也提高了性能。此外,我们还开发了一种多尺度帧合成方案以充分实现 Transformer 的潜力。大量实验表明,所提出的模型在各种基准数据集上在数量和质量上都优于最先进的方法。代码和模型发布在 https://github.com/zhshi0816/Video-Frame-Interpolation-Transformer 上。

1. 引言

视频帧插值旨在通过在现有帧之间合成新帧来对输入视频进行时间上采样。这是计算机视觉中的一个基本问题，涉及对运动、结构和自然图像分布的理解，这有助于众多下游应用，如图像恢复[5, 52]、虚拟现实[1]和医学成像[22]。

大多数最先进的视频帧插值方法都是基于深度卷积神经网络（CNNs）[3, 20, 25, 29, 30, 32, 37, 53]。虽然这些基于CNN的架构实现了最先进的性能，但它们通常存在两个主要的缺点。首先

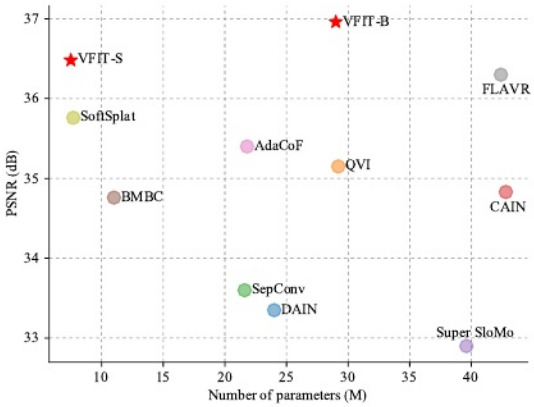

图 1 。使用 Vimeo-90K 数据集对性能和模型大小进行比较[54]。VFIT 以更少的参数优于最先进的方法。VFIT-S 和 VFIT-B 分别表示所提出的小型模型和基础模型。

卷积层是内容无关的，即使用相同的核与不同输入的不同位置进行卷积。虽然这种设计可以作为图像识别模型获取平移等性的理想归纳偏差[24]，但它并不总是适用于视频插值，因为视频插值涉及一个复杂的运动补偿过程，该过程是空间变化的并且与内容相关。因此，采用卷积神经网络（CNN）骨干网络可能会限制自适应运动建模的能力，并有可能限制视频插值模型的进一步发展。

其次，在视频插值中，捕获长距离依赖关系至关重要，其中大的运动场带来了最显著的挑战。然而，大多数卷积神经网络（CNNs）[25, 53]通常采用小卷积核（通常为 VGG 建议的 3×3），这在利用长距离信息方面效率低下，因此在合成高质量视频帧方面效果较差。虽然在卷积层中使用更大的核似乎是一个简单的解决方案，但这会显著增加模型参数数量和计算成本，从而在训练中导致不良的局部最小值，如果没有适当的正则化措施。

*These authors contributed equally.
†Corresponding author.

此外，仅仅堆叠多个小核层以获得更大的感受野也无法完全解决这个问题，因为多跳方式无法有效地学习远距离依赖关系[45]。

另一方面，Transformer[43]最初是为自然语言处理（NLP）而设计的，用于高效地对输入和输出之间的长距离依赖关系进行建模，它自然地克服了基于卷积神经网络（CNN）算法的上述缺点，尤其适用于视频插值任务。受其在NLP中取得成功的启发，最近有几种方法将Transformer应用于计算机视觉，并在各种任务上展示了有前景的结果，如图像分类[13, 41]、语义分割[44]、目标检测[8]和3D重建[51]。然而，如何将Transformer有效地应用于涉及额外时间维度的视频插值仍然是一个开放且具有挑战性的问题。

在这项工作中，我们提出了视频帧插值变换器（VFIT），以实现有效的视频插值。与典型的Transformer[8, 9, 13]相比，这些Transformer的基本模块大多是从原始的NLP模型[43]中借用的，而所提出的VFIT中有三个不同的设计来生成逼真且时间一致的帧。首先，原始的Transformer[43]基于自注意力层，该层与输入元素（例如像素）进行全局交互。由于这种全局操作在元素数量上的复杂度为二次方，直接将其应用于我们的任务会导致由于视频的高度维度特性而产生极高的内存和计算成本。一些方法[7, 9]通过将特征图划分为补丁，并将每个补丁视为自注意力中的新元素来规避这个问题。然而，这种策略无法对每个补丁内部的像素之间的细粒度依赖关系进行建模，而这种依赖关系对于合成逼真的细节至关重要。此外，它可能会在补丁边界周围引入边缘伪影。相比之下，我们将 Swin [27] 的局部注意力机制引入到 VFIT 中，以解决复杂性问题，同时保留其移位窗口方案对长程依赖关系的建模能力。我们证明，通过适当的开发和适应，最初用于图像识别的局部注意力机制可以有效地提高视频插值的性能，且参数数量较少，如图 1 所示。

其次，原始的局部注意力机制[27]仅适用于图像输入，无法轻易应用于涉及额外时间维度的视频插值任务。为了解决这个问题，我们将局部注意力的概念扩展到时空域，从而得到了与视频兼容的时空 Swin 注意力层（STS）。然而，这种简单的扩展在使用大窗口大小时可能会导致内存问题。为了使我们的模型更节省内存，我们进一步设计了一个时空可分离版本的 STS，称为 Sep-STS，通过分解时空自注意力来实现。有趣的是，Sep-STS不仅有效地减少了内存使用，还显著提高了视频插值性能。

为了充分发挥我们的 Sep-STS 的潜力，我们提出了一种新的多尺度核预测框架，能够更好地处理不同视频中的多尺度运动和结构，并以由粗到精的方式生成高质量的视频插值结果。所提出的 VFIT 简洁、灵活、轻量级、高性能、快速且内存高效。如图 1 所示，一个较小的模型（VFIT-S）仅用其 17.7%的参数就已经比最先进的 FLAVR 方法[21]高出 0.18 分贝，而我们的基础模型（VFIT-B）用其 68.4%的参数实现了 0.66 分贝的提升。

2. 相关工作

视频帧插值。现有的视频帧插值方法大致可以分为三类：基于流的[3,20,32,38,53]、基于核的[25,29 - 31]以及基于直接回归的方法[22]。

基于流的方法[3,20,32,53]通过根据预测的光流对源图像中的像素进行变形来生成中间帧。尽管这些方法表现良好，但它们通常基于简化的运动假设，如线性[20]和二次[53]，这限制了它们在许多违反这些假设的现实场景中的性能。

与基于流的方法不同，基于核的方法[25,29-31]不依赖于任何预设的假设，因此对不同的视频具有更好的泛化能力。例如，SepConv [30]预测自适应可分离核来聚合输入的源像素，而AdaCoF [25]学习可变形的空间可变核，用于与输入帧卷积以生成目标帧。然而，这些方法通常在单个尺度上应用核预测模块，因此无法有效处理不同尺度下可能出现的复杂运动和结构。此外，这些基于CNN的方法没有考虑像素之间的长距离依赖关系。相比之下，我们提出了一种基于多尺度Transformer的核预测模块，将在第4节中展示其在视频插值方面的更高质量结果。

最近，Kalluri 等人[21]提出了一种直接回归目标帧的卷积神经网络（CNN）模型，该模型取得了最先进的结果。如图 1 所示，所提出的 VFIT 以更少的参数明显优于该方法，这清楚地展示了 Transformer 在视频插值中的优势。

视觉 Transformer。Transformer 最近已被应用于各种视觉任务，例如视觉分类[4]

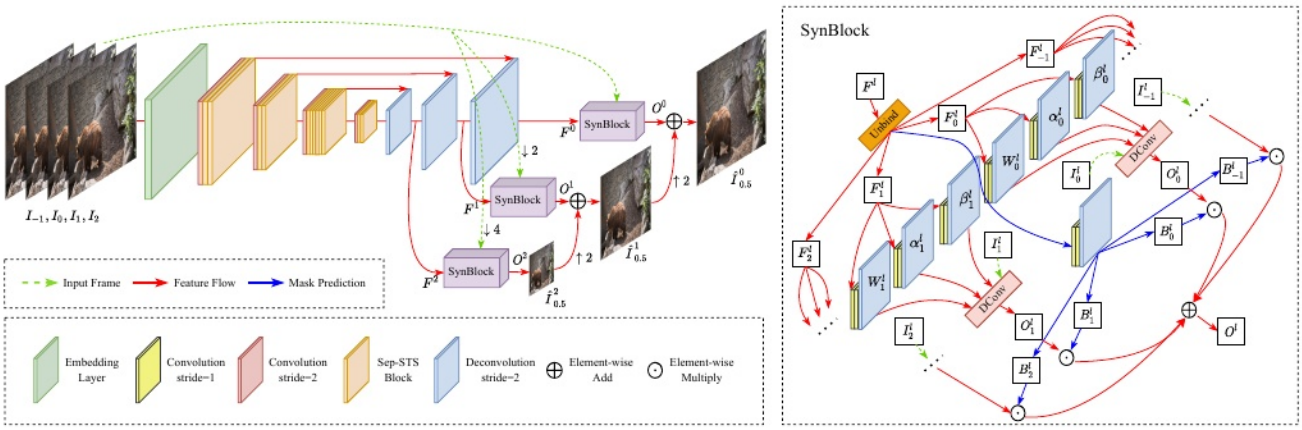

图 2 。所提出的视频帧插值（VFIT）概述。我们首先使用嵌入层将输入帧转换为浅层特征，然后使用基于 Transformer 的编码器-解码器网络提取深层分层特征。这些特征与输入帧一起被输入到多尺度帧合成网络中，该网络由三个合成块组成，以获得最终输出。“↓n”和“↑n”分别表示下采样和上采样，比例为 n。“DConv”表示[50]中的广义可变形卷积。请注意，合成块可以被视为源自 STPAN [50]的 AdaCoF [25]的多帧扩展。更多详细的解释请见第 3 节。

13、27、42、48]，目标检测[8]，语义分割[44]，3D 重建[51]，以及图像恢复[9]。然而，它尚未被应用于视频帧插值。在本研究中，我们提出了 VFIT，它以轻量级模型实现了最先进的性能。为了克服全局自注意力导致的高计算成本，我们引入了 Swin[27]的局部注意力机制，以避免复杂性问题的同时保留长距离依赖建模的能力。我们注意到，一项并发工作[26]也使用了局部注意力进行低级视觉任务。然而，它只考虑图像输入，无法处理视频，由于额外的时间维度，视频处理更具挑战性。相比之下，我们将局部注意力的概念扩展到时空域，以实现基于 Transformer 的视频插值，并提出了一种时空分离策略，该策略不仅节省了内存使用，还作为有效的正则化手段促进了性能提升。

3. 提议的方法

图2展示了所提出模型的概述。与现有方法[21, 29, 30, 53]类似，为了合成中间帧$I_{0.5}$，我们使用其T个相邻帧$I\{-(bTc-1), \cdots, 0,1, \cdots, dTe\}$作为输入。具体来说，

当 T 为 4 时，帧率为 I−1、I0、I1、I2 。

所提出的 VFIT 由三个模块组成：浅层特征嵌入、深层特征学习和最终的帧合成。首先，嵌入层接收输入帧并为深层特征学习模块生成浅层特征。与[27]类似，浅层嵌入是通过卷积层实现的，其中我们采用3D卷积而不是[27]中的2D卷积，以更好地编码输入序列的时空特征。接下来，我们将浅层特征输入到深层模块中，以提取分层特征表示$\{F_l, l = 0, 1, 2\}$，以捕获多尺度运动信息（第3.1节）。最后，通过帧合成块（如图2中的SynBlocks）使用深层特征F^1（第3.2节）可以生成中间帧$I_{0.5}$。

3.1. 学习深层特征

如图 2 所示，我们使用基于 Transformer 的编解码器架构来学习特征。编码器由四个阶段组成，每个阶段都以一个步长为 2 的 3D 卷积层开始，对输入特征进行下采样，下采样层之后是几个 Sep-STS 模块，这是我们框架的主要组成部分。对于解码器，我们使用一种仅具有三个步长为 2 的 3D 反卷积层的轻量级结构，对低分辨率特征图进行上采样。请注意，在整个网络中，我们只调整特征的空间维度，而保持时间大小不变。接下来，我们将对所提出的 Sep-STS 模块进行更多解释。

局部注意力。现有的 Transformer 模型[8, 13, 43]主要采用全局注意力机制来聚合输入中的信息，这可能导致视频帧插值产生极高的内存和计算成本。解决这个问题的一个简单方法是将特征图直接划分为补丁，并将每个补丁视为全局注意力中的新元素[7, 9]。这种策略相当于使用像素重排[36]（下采样因子等于补丁大小）对输入进行激进的下采样，并且无法很好地重建需要像素之间精细依赖建模的高质量图像细节。

在这项工作中，我们引入了 Swin Transformer 的局部注意力机制[27]，它能够有效地.....

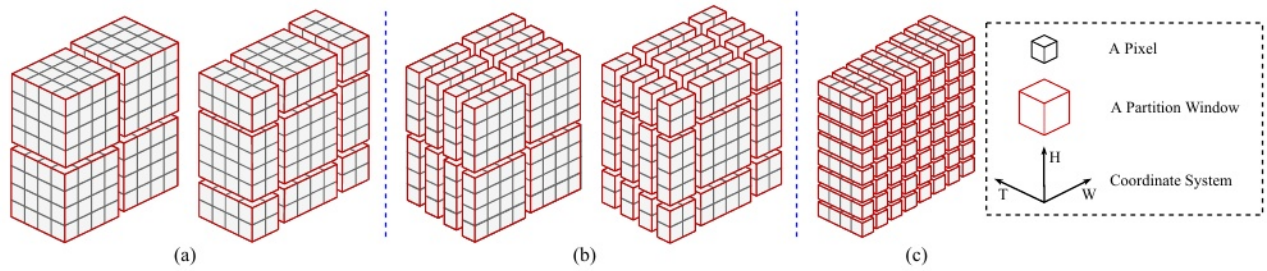

图 3 。不同局部划分策略的图示。（a）STS 时空立方体的常规划分和偏移划分。（b）分离式 STS 空间窗口的常规划分和偏移划分。（c）分离式 STS 的时间向量划分。

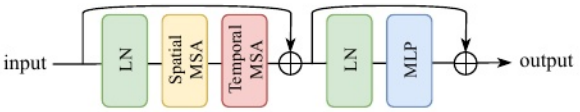

图 4 。Sep-STS 模块的图示。空间多头自注意力（Spatial MSA）和时间多头自注意力（Temporal MSA）分别表示空间和时域局部窗口中的多头自注意力（第 3.1 节）。

解决上述问题。首先，由于 Swin 的自注意力是在局部窗口中计算的，它自然避免了全局注意力带来的沉重计算负担。其次，Swin 采用了移位窗口划分策略来连接不同的局部区域，交替使用常规和移位窗口划分能够实现长距离依赖建模。然而，这种方法是为图像应用设计的，不能轻易应用于视频。

时空局部注意力。为了使 Swin Transformer 与视频输入兼容，我们将局部注意力机制泛化到时空空间，并提出了 STS 注意力。如图 3(a)所示，STS 在概念上与 Swin 相似，但涉及一个额外的时间维度。

给定一个大小为 C×T×H×W 的输入特征，其中 C、T、H、W 分别表示通道、时间、高度和宽度维度，我们首先将其划分为 $_{HW}$ 非

将重叠的 3D 子立方体的形状设定为 T×M×M ^{M_2}（图 3(a)-左），然后在每个子立方体上执行标准的多头自注意力（MSA）。请注意，这个立方体的每个元素都是一个 C 维特征向量，为了简单起见，我们在描述分区策略时省略了通道维度。一旦处理完所有的子立方体，我们将它们合并回以恢复输入的原始形状。为了在相邻的立方体之间建立连接，我们采用一种移位立方体分区策略，将立方体向左上方移动（bM ）

$_{M}^{2c}$，b$_{2c}$）

像素（图 3（a - 右）

空间与时间的分离。尽管上述的 STS 能够处理视频输入，但在处理大立方体尺寸*(即大的 T 或 M)时可能会受到内存问题的困扰*。为了缓解这个问题，我们提出了 Sep-STS，将时空计算分离为空间和时间。

首先，对于空间计算，给定一个大小为 C×T×H×W 的输入特征图，我们首先将其划分为 $_{T}$$_{M}^{HW}$$_{2}$ 个大小为 M×M的非重叠二维子窗口，如图 3(b)-左所示，然后对每个子窗口执行标准的 MSA。对于不同窗口的连接，由于我们在此将计算限制在二维中，我们简单地对每一帧使用 Swin 的平移窗口划分策略，如图 3(b)-右所示。

其次，对于时间维度的计算，我们将输入特征图重塑为如图 3(c)所示的长度为 T 的 HW 时间向量，并在每个向量内部执行多头自注意力机制（MSA），以便对帧之间的依赖关系进行建模。此步骤补充了空间域中的自注意力机制，因此这两个操作需要一起使用来处理视频。

分离式时空注意力模块。基于分离式时空注意力，我们设计了主要组件——分离式时空注意力模块，它由独立的空间和时间注意力模块以及一个多层感知机（MLP）组成（图4）。MLP采用两层结构，并使用GELU函数进行激活。与[27]类似，我们在该模块中应用层归一化（LN）[2]和残差连接[16]以稳定训练。与Swin类似，我们在连续的分离式时空注意力模块中交替使用常规和移位分区来建模长距离时空依赖关系。

内存使用。Sep-STS 注意力机制将计算成本高昂的操作在空间和时间上分解为两个计算成本较低的操作，这有效地减少了内存使用，从 STS 的 O（（TMM）·THW）减少到我们的 Sep-STS 的 O（（T + MM）·THW）。

在训练期间，与 STS 基线相比，我们使用 Sep-STS 观察到 GPU 内存减少了 26.2%。由于窗口大小 MM 通常远大于输入帧数 T，这种减少率本质上取决于 T，按照最先进算法的设置[11,21,53]，我们默认将其设置为 4。由于所提出的框架是灵活的，可用于任意数量的帧，Sep-STS 对于更大的 T 有可能实现更显著的内存减少。此外，时空分离策略还可以像内存使用一样降低计算成本。然而，由于 Sep-STS 是简单地用两个独立的

在我们的实验中，PyTorch 的运行时间实际上与 STS 的运行时间相似。有可能，使用定制的 CUDA 内核对其实现进行优化可能会进一步提高效率。

讨论。在本研究中，我们探讨了基于 Transformer 的视频插值中局部注意力的概念。类似的概念在其他近期方法中已被采用，例如局部关系网络[19]、独立网络[35]和 Swin[27]。然而，这些算法是为图像设计的，由于额外的时域维度带来的困难，对利用视频的局部注意力机制的关注较少。此外，现有方法主要集中于通常被视为高级视觉任务的图像识别任务，而在本研究中，我们更强调运动建模和外观重建。在本研究中，我们专注于局部注意力模块的时间扩展，以实现有效的视频帧插值。我们探索了时空可分离的局部注意力，其精神类似于 MobileNet[18]，后者通过将标准卷积分解为深度卷积和点态卷积来改进它。此外，我们提出了一个多尺度核预测框架，以充分利用局部注意力学习到的特征，接下来将详细介绍。

3.2. 帧合成

利用所提出的编解码器网络的特征，我们的 VFIT 通过预测空间变化的核来合成输出图像，以自适应地融合源帧。与现有的基于核的视频插值方法[25, 29, 30, 37]不同，我们提出了一个使用分层特征{F[l, l] = 0, 1, 2}的多尺度核预测框架，如图 2 所示。

VFIT 的框架合成网络由三个在不同尺度进行预测的合成模块组成，并且每个合成模块都是一个核心预测网络。VFIT 通过以下方式融合这些多尺度预测来生成最终结果：

$$\hat{I}^l_{0.5} = f_{\text{up}}(\hat{I}^{l+1}_{0.5}) + O^l, \tag{1}$$

$$O^l = f^l_{\text{syn}}(F^l, I^l_{\{-(\lfloor \frac{T}{2} \rfloor - 1), \cdots, \lceil \frac{T}{2} \rceil\}}), \tag{2}$$

其中 l = 0、1、2 表示从精细到粗略的不同尺度，f_{up} 表示双线性上采样函数。在更精细的尺度I⊠$_{0.5^l}$ 上的合成帧可以通过合并来自粗尺度(f_{up}(I⊠$_{0.5^{l+1}}$)) 的上采样输出和当前合成块（Ol）的预测来获得。在最精细的尺度I⊠$_{0.5^0}$ 上的输出是我们 VFIT 的最终结果，*即*$I_{0.5} = I_{0.5}$,*初始值为*I⊠$_{0.5}$ ⊠ ⊠*0* 。

3 等于 0 。这里 f_{syn} 是第 l 个合成块，它具有时空 l 。

特征 F 以及帧序列 $I_{\{-(b\ T}$ 如下

2c−1），⋯，d T e} 2

输入，并且I^l_t 表示通过双线性插值以2

l 为因子下采样得到的帧I_t，其中I^0_t 等同于未下采样的原始帧。

SynBlock。给定输入特征图 F l ∈RC×T×H×W，SynBlock 通过估计一组广义可变形核[50]来在 l 尺度生成其预测，以聚合来自源帧的信息。

如图 2 所示，我们首先在时间维度上解除 F1 的绑定，以获得所有输入帧的 T 个分离的特征图，记为$F_{\{-(b}$ l

T, 2 并且对于每一个

2c−1），⋯，d T e}

2

对于帧 t 和 Ftl ∈RC×H×W 。然后我们将 Ftl 输入到三个小的二维卷积神经网络中，以获得帧 Itl 的每个像素的可变形核，包括核权重 Wtl ∈RK×H×W 、水平偏移量αtl ∈RK×H×W 和垂直偏移量βtl ∈RK×H×W，其中 K 是每个核的采样位置数量。

利用预测的内核，我们得到第 t 帧中位于（x，y）位置的合成块的输出：

$$O^l_t(x,y) = \sum_{k=1}^{K} W^l_t(k,x,y) I^l_t(x + \alpha^l_t(k,x,y), y + \beta^l_t(k,x,y)),$$

它使用类似于[50]的自适应权重 W 对（x，y）周围的相邻像素进行聚合。

最后，我们通过将所有帧的O^l_t与学习到的掩码进行融合来生成尺度为 l 的输出。具体而言，我们将特征图$F_{\{-(b}$进行连接。

−T T 在通道 dt - 2 c - 1），⋯，d 2 e 处

将连接后的特征进行维度和发送至一个小的二维卷积神经网络，以生成 T 融合掩码 $B_{\{-(b}$

l −T T, 2 c−1），⋯, d 2 e }

请注意，我们使用 softmax 函数作为卷积神经网络（CNN）的最后一层，以沿时间维度对掩码进行归一化。SynBlock 的最终输出 f_{syn}

是由以下产生的：

$$O^l = \sum_t B^l_t \cdot O^l_t. \tag{3}$$

请注意，这种合成块可以被视为[25, 37]的多帧扩展，其源自 STPAN 的广义可变形核[49]。

4. 实验

4.1. 实施细节

网络。如图 2 所示，VFIT 编码器由四个阶段组成，分别具有 2 个、2 个、6 个和 2 个 Sep-STS 模块。编码器和解码器之间的跳跃连接通过连接来实现。对于所有三个 Syn-Blocks，我们将可变形核大小设置为 K = 5×5。我们展示了 VFIT 的两个变体：基础模型 VFIT-B 和较小的模型 VFIT-S，其中 VFIT-S 的模型大小约为 VFIT-B 的 25%。这两个模型使用相同的架构，唯一的区别在于每个阶段的通道维度，对于 VFIT-S，我们将通道缩小了一半。

表 1 。在 Vimeo-90K、UCF101 和 DAVIS 数据集上进行定量比较。加粗的数字表示最佳性能，下划线数字表示次优性能。

方法	# 参数（米）	Vimeo - 九万		UCF101		戴维斯	
		峰值信噪比（↑）	结构相似性指数（↑）	峰值信噪比（↑）	结构相似性指数（↑）	峰值信噪比（↑）	结构相似性指数（↑）
超级慢动作 [20]	39.6	32.90	0.957	32.33	0.960	25.65	0.857
戴恩 [3]	24.0	33.35	0.945	31.64	0.957	26.12	0.870
分离卷积 [30]	21.6	33.60	0.944	31.97	0.943	26.21	0.857
英国数学学会[32]	11.0	34.76	0.965	32.61	0.955	26.42	0.868
凯恩 [12]	42.8	34.83	0.970	32.52	0.968	27.21	0.873
AdaCoF [25]	21.8	35.40	0.971	32.71	0.969	26.49	0.866
QVI [53]	29.2	35.15	0.971	32.89	0.970	27.17	0.874
软溅 [28]	7.7	35.76	0.972	32.89	0.970	27.42	0.878
风味 [21]	42.4	36.30	0.975	33.33	0.971	27.44	0.874
VFIT-S	7.5	36.48	0.976	33.36	0.971	27.92	0.885
VFIT-B	29.0	36.96	0.978	33.44	0.971	28.09	0.888

训练。对于训练我们的网络，我们使用一个简单的“1 损失：||I0.5 - I^0.5||，其中 I0.5 是真实值。我们使用 AdaMax 优化器[23]，其中 $\beta_1 = 0.9$，$\beta_2 = 0.999$。训练批次大小设定为 4。我们对模型进行 100 个训练周期，初始学习率设定为 2e−4，并逐渐衰减到 1e−6 。

数据集。与[21]类似，我们使用Vimeo-90K七帧训练集[54]来学习我们的模型，该训练集包含64,612个七帧序列，分辨率为448×256。每个序列的第一、第三、第五和第七帧对应于图2中的I−1、I_0、I_1和I_2，用于预测第四帧对应的$I_{0.5}$。为了进行数据增强，我们从帧中随机裁剪256×256的图像块，并进行水平和垂直翻转以及时间顺序反转。

我们在广泛使用的基准数据集上对模型进行评估，包括广泛使用的基准数据集，包括 Vimeo-90K 七元组测试集[54]、UCF101 数据集[40]和 DAVIS 数据集[34]。遵循[21,53]，我们报告了在由 UCF101 生成的 100 个五元组和由 DAVIS 生成的 2847 个五元组上的性能。

4.2. 与现有技术的对比评估

我们将所提出的算法与最先进的视频插值方法进行了评估：SepConv [30]、DAIN [3]、SuperSloMo [20]、CAIN [12]、BMBC [32]、AdaCoF [25]、SoftSplat [28]、QVI [53]和 FLAVR [21]。在这些方法中，SuperSloMo、DAIN、CAIN、QVI、AdaCoF和FLAVR与我们的模型使用相同的训练数据进行训练。对于SepConv和BMBC，由于训练代码不可用，我们直接使用预训练模型进行评估。SoftSplat [28]的结果由作者提供。

我们在表 1 中展示了定量评估，其中峰值信噪比（PSNR）和结构相似性指数（SSIM）[46]被用于图像质量评估，这与之前的工作类似。得益于分离式空间时间变换（Sep-STS）模块的学习能力，所提出的 VFIT 实现了

表 2 。每帧评估方法的运行时间（以秒为单位）。这些模型是在一台配备英特尔酷睿 i7 - 8700K CPU 和 NVIDIA GTX 2080 Ti GPU 的台式电脑上进行测试的。结果是在 Vimeo - 90K 数据集上平均得出的。

方法	最佳管理业务案例		风味	VFIT-S	VFIT-B
运行时	0.57	0.08	0.15	0.08	0.14

比评估的基于卷积神经网络（CNN）的方法表现更好，展示了使用 Transformer 进行视频插值的优势。具体而言，仅使用 750 万个参数，VFIT-S 就能在所有评估数据集上优于迄今为止最好的视频插值方法 FLAVR。此外，VFIT-B 相对于 FLAVR 实现了更显著的改进（在 Vimeo-90K 上 0.66 分贝，在 DAVIS 上 0.65 分贝）。由于 UCF101 的视频质量相对较低，图像分辨率低且运动缓慢，如[53]中所解释，我们的性能提升不太显著。请注意，VFIT 的显著改进完全来自架构设计，不依赖于任何外部信息，这与之前的一些工作[3,28,53]有很大不同，这些工作使用预训练的光流和/或深度模型，从而隐性地受益于额外的运动和/或深度标签。

此外，我们在图5中进行了定性比较，其中所提出的VFIT在视觉上比基准方法产生了更令人愉悦的结果，结构更清晰，失真更少。此外，为了评估插值结果的准确性，我们在图6中展示了插值帧与相应地面真理的重叠。VFIT的重叠图像比基准方法清晰得多，即*更接近*地面真理，表明VFIT在运动建模方面具有更好的能力。

我们在表 2 中还展示了我们方法的运行时间。VFIT 的运行时间性能与基于卷积神经网络（CNN）的最佳算法相当，这有助于其在视觉应用中的部署。

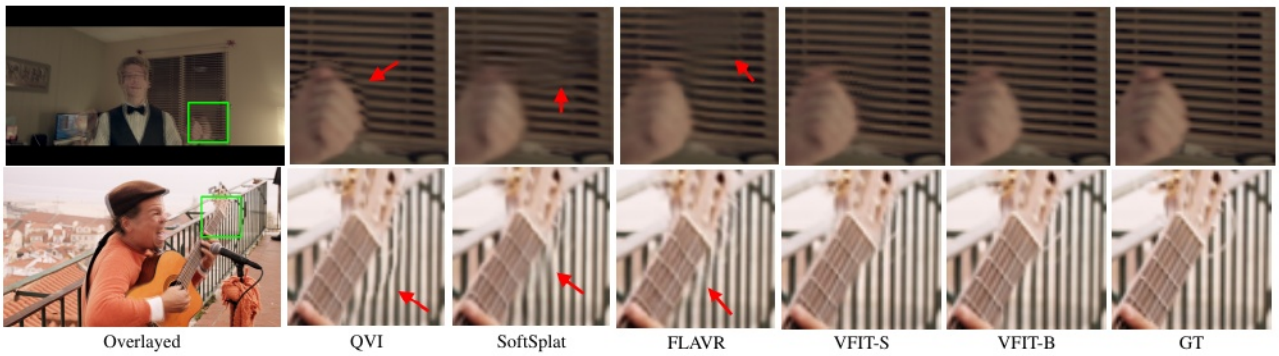

图 5. 与最先进的视频插值算法的定性比较。VFIT 生成的结果质量更高，结构更清晰，失真更少。

图 6. 插值帧与相应真实值的重叠，重叠图像越清晰表明预测越准确。请注意，对于第二个示例，由于基线方法的预测与真实值未良好对齐，红色和白色区域的重叠呈现出模糊的粉色。

4.3. 烧蚀研究

我们在 Vimeo-90K 数据集上进行消融研究。由于我们注意到在早期训练阶段，训练过程很快收敛，此时模型之间的差异已经可以区分，因此在本研究中，我们对所有模型训练 20 个周期，以加快开发进度，并专注于 VFIT 最关键的部分。

局部注意力。与引入局部注意力机制的我们的模型不同，最近的一些方法[7,9]遵循自然语言处理中传统 Transformer 的基本结构，在视觉应用中使用全局注意力，通过将输入分割成补丁并将每个补丁重新定义为自注意力中的新元素来避免全局注意力的高计算成本。在我们的实验中，我们也尝试了这一策略，用基于补丁的全局注意力块替换 VFIT-B 的每个 Sep-STS 块，这被称为 VFIT-Global。如表 3 所示，VFIT-Global 的结果比 VFIT-B 低多达 0.84 分贝，这强调了基于 Transformer 的视频帧插值中局部注意力的重要作用。

分离空间时间序列（Sep-STS）。为了进一步验证 Sep-STS 模块的有效性，我们将我们的 VFIT-B 与其两个变体进行了比较：1）VFIT-CNN，其中所有的 Sep-STS 模块都被卷积残差块[16]所取代，并且每个残差块由两个 3D 卷积层组成；2）VFIT-STS，其中 Sep-STS 模块被其不可分离的对应模块，*即* STS 模块所取代。

如表3所示，尽管VFIT-CNN使用的参数是VFIT-STS的两倍多，但这两个模型取得了相似的结果，展示了使用Transformer进行视频插值的优势。此外，我们的基础模型VFIT-B，使用所提出的Sep-STS作为构建模块，比VFIT-STS表现更好。应该强调的是，性能的提升是显著的，因为Sep-STS模块最初的设计目的是减少内存使用，如第3.1节所讨论的。这可以归因于STS中大尺寸子立方体的自注意力相对难以学习，而Sep-STS中的时空分离可以作为低秩正则化[6]来解决这个问题。

为了更好地分析我们模型的性能，我们进一步在不同运动条件下与基准进行比较。遵循[15,47]，我们将Vimeo-90K测试集分别划分为快速、中等和慢速运动。表4显示，VFIT-B在快速运动上比VFIT-CNN高出0.43 dB，在中等运动上高出0.16 dB，在慢速运动上高出0.10 dB，这突出了所提出的Sep-STS在处理具有挑战性的大运动场景方面的卓越能力。我们还提供了来自一段视频的插值帧。

表 3 。所提出的 Sep-STS 模块的有效性

方法	峰值信噪比	结构相似性指数（SSIM）	#参数（米）
VFIT-B	36.02	0.975	29.0
VFIT-STS	35.84	0.974	29.1
VFIT-CNN	35.82	0.973	65.4
VFIT-全球	35.18	0.971	42.4
M = 4	35.82	0.974	29.0
M = 6	35.90	0.974	29.0
M = 8	36.02	0.975	29.0
M = 10	35.93	0.974	29.0

表 4 。在不同运动条件下与基础模型的比较。

方法	快速	中等	慢
VFIT-B	33.23/0.954	35.91/0.976	38.36/0.987
VFIT-STS	32.91/0.950	35.77/0.975	38.27/0.987
VFIT-CNN	32.80/0.950	35.75/0.975	38.26/0.987
VFIT-全球	32.15/0.945	35.10/0.972	37.62/0.985

在图 7 中与快速运动进行比较。

为了分析 Sep-STS 不同窗口大小的效果，我们分别对 M = 4、6、8、10 的 VFIT-B 进行评估。表 3 显示，随着窗口大小的增加，我们的模型表现更好，直到 M > 8。因此，在本研究中，我们选择 M = 8 作为默认设置。

多尺度帧合成。在 3.2 节中，我们为最终的帧合成提出了一个多尺度核预测网络。为了验证这种设计的效果，我们通过移除图 2 中的第二个和第三个合成块，对 VFIT 的单尺度变体（称为 VFIT-Single）进行了实验。这种单尺度策略本质上与[25,29,30]中的普通核预测网络类似。VFIT-Single 实现的峰值信噪比（PSNR）为 35.54 分贝，比我们的基础模型 VFIT-B 低 0.48 分贝。巨大的性能差距表明了多尺度框架对于充分实现 Transformer 潜力的重要性。

请注意，我们仅将损失函数应用于最终输出，*即如*第4.1节所述的多尺度框架的最精细级别输出I⊠$_{0.5}$0。或者，可以考虑对网络的所有尺度输出添加监督。然而，我们通过实验发现，这种方案表现不佳。

调整大小的模块。如图 2 所示，我们使用 3D 卷积和反卷积层对特征图进行下采样和上采样。鉴于我们的 Sep-STS 相对于基于 CNN 的模型在性能上有所提升，探索将 Transformer 层用作视频帧插值的调整大小模块是非常有趣的。

为了回答这个问题，我们采用了[14]中的方法，该方法通过下采样自注意力层的查询来引入一个基于 Transformer 的调整大小模块用于视频分类。为了实现基于 Transformer 的上采样，我们扩展了[14]中的想法，通过上采样查询。

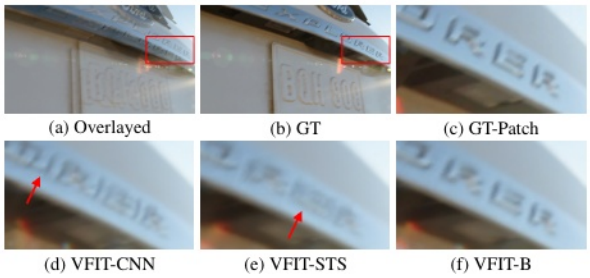

图 7 。来自快速运动视频的插值帧。VFIT-CNN 由于无法处理大幅运动而产生了严重的重影失真，而 VFIT-STS 的结果显得模糊。相比之下，VFIT-B 生成的中间帧质量更高，更接近真实情况。

表 5 。与基于 Transformer 的调整大小模块的比较

方法	峰值信噪比	结构相似性指数（SSIM）	运行时间（秒）
VFIT-B	36.02	0.975	0.14
VFIT-TD	35.92	0.974	0.17
VFIT-TU	35.97	0.974	0.20

双线性插值。我们分别用基于 Transformer 的下采样和上采样模块替换了 VFIT-B 的卷积和反卷积层，并将这两个变体分别称为 VFIT-TD 和 VFIT-TU。如表 5 所示，VFIT-TD 和 VFIT-TU 的表现均略逊于我们的基础模型，且运行时间性能下降，这表明当前计算机视觉中基于 Transformer 的调整大小操作的当前设计对于复杂的运动建模效果较差。这是我们当前工作的一个局限性，也是未来研究的一个有趣问题。

5. 结论

在本文中，我们提出了一种参数、内存和运行时间高效的 VFIT 框架，用于视频帧插值，具有最先进的性能。我们的很大一部分工作集中于将局部注意力机制扩展到时空空间，并且该模块可以集成到其他视频处理任务中。此外，我们展示了新颖的时空分离方案的有效性，这意味着视频 Transformer 中需要结构良好的正则化。VFIT 的架构简单紧凑，能够有效地应用于众多下游视觉任务。

与大多数现有的基于核的方法[25, 29, 30, 37]类似，我们使用 VFIT 仅进行 2× 插值。然而，通过预测与不同时间步长相关的核，可以轻松地将其扩展到多帧插值，甚至通过类似于[10]的方式将时间作为额外的输入进行任意时间插值。这将是未来工作的一部分。

致谢。杨敏慧（音译）的部分研究得到了美国国家科学基金会职业发展奖 #1149783 的支持。

参考文献

[1] Robert Anderson, David Gallup, Jonathan T Barron, Janne Kontkanen, Noah Snavely, Carlos Hern´andez, Sameer Agar-wal, and Steven M Seitz. Jump: virtual reality video. *ACM Transactions on Graphics*, 35(6):1–13, 2016. 1

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hin-ton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 6

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning*, 2021. 2

[5] Tim Brooks and Jonathan T Barron. Learning to synthe-size motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[6] Emmanuel J Cand`es, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011. 7

[7] Jiezhang Cao, Yawei Li, Kai Zhang, and Luc Van Gool. Video super-resolution transformer. *arXiv preprint arXiv: 2106.06847*, 2021. 2, 3, 7

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, 2020. 2, 3

[9] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Pro-ceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3, 7

[10] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 8

[11] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Juwei Lu, Jin Tang, and Konstantinos N Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *Proceedings of the European Conference on Computer Vision*, 2020. 4

[12] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 6

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*, 2020. 2, 3

[14] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 8

[15] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 7

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 4, 7

[17] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[18] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 5

[19] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 5

[20] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 6

[21] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. *arXiv preprint arXiv:2012.08512*, 2020. 2, 3, 4, 6

[22] Alexandros Karargyris and Nikolaos Bourbakis. Three-dimensional reconstruction of the digestive wall in capsule endoscopy videos using elastic video interpolation. *IEEE Transactions on Medical Imaging*, 30(4):957–971, 2010. 1, 2

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2014. 6

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. 1

[25] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive col-laboration of flows for video frame interpolation. In *Pro-ceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 3, 5, 6, 8

[26] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. *arXiv preprint arXiv:2108.10257*, 2021. 3

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Com-puter Vision*, 2021. 2, 3, 4, 5

[28] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 6

[29] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni-tion*, 2017. 1, 2, 3, 5, 8

[30] Simon Niklaus, Long Mai, and Feng Liu. Video frame inter-polation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1, 2, 3, 5, 6, 8

[31] Simon Niklaus, Long Mai, and Oliver Wang. Revisiting adaptive convolutions for video frame interpolation. In *Pro-ceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2021. 2

[32] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost vol-ume for video interpolation. In *Proceedings of the European Conference on Computer Vision*, 2020. 1, 2, 6

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019. 5

[34] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6

[35] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. In *Advances in Neural Infor-mation Processing Systems*, 2019. 5

[36] Wenzhe Shi, Jose Caballero, Ferenc Husz´ar, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[37] Zhihao Shi, Xiaohong Liu, Kangdi Shi, Linhui Dai, and Jun Chen. Video frame interpolation via generalized deformable convolution. *IEEE Transactions on Multimedia*, 2021. 1, 5, 8

[38] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: ex-treme video frame interpolation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 2

[39] Karen Simonyan and Andrew Zisserman. Very deep convo-lutional networks for large-scale image recognition. In *Pro-ceedings of the International Conference on Learning Rep-resentations*, 2015. 1

[40] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6

[41] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herv´e J´egou. Training data-efficient image transformers & distillation through at-tention. In *Proceedings of the International Conference on Machine Learning*, 2021. 2

[42] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni-tion*, 2018. 2

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko-reit, Llion Jones, Aidan N Gomez, ☒ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 2, 3

[44] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *Proceed-ings of the European Conference on Computer Vision*, 2020. 2, 3

[45] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaim-ing He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni-tion*, 2018. 2

[46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Si-moncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Process-ing*, 13(4):600–612, 2004. 6

[47] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P Allebach, and Chenliang Xu. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 7

[48] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Pro-ceedings of the European Conference on Computer Vision*, 2018. 2

[49] Gang Xu, Jun Xu, Zhen Li, Liang Wang, Xing Sun, and Ming-Ming Cheng. Temporal modulation network for con-trollable space-time video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6388–6397, 2021. 5

[50] Xiangyu Xu, Muchen Li, Wenxiu Sun, and Ming-Hsuan Yang. Learning spatial and spatio-temporal pixel aggrega-tions for image and video denoising. *IEEE Transactions on Image Processing*, 2020. 3, 5

[51] Xiangyu Xu and Chen Change Loy. 3D human texture esti-mation from a single image with transformers. In *Proceed-ings of the IEEE International Conference on Computer Vi-sion*, 2021. 2, 3

[52] Xiangyu Xu, Jinshan Pan, Yu-Jin Zhang, and Ming-Hsuan Yang. Motion blur kernel estimation via deep learning. *IEEE Transactions on Image Processing*, 27(1):194–205, 2017. 1

[53] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *Advances in Neural Information Processing Systems*, 2019. 1, 2, 3, 4, 6

[54] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 1, 6