用于可控时空视频超分辨率任务的时间调制网络

Gang Xu[1] Jun Xu[2*] Zhen Li[1] Liang Wang[3] Xing Sun[4] Ming-Ming Cheng[1]
[1] College of Computer Science, Nankai University, Tianjin, China
[2] School of Statistics and Data Science, Nankai University, Tianjin, China
[3] National Lab of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[4] Youtu Lab., Tencent, Shanghai, China

Abstract

时空视频超分辨率（*STVSR*）旨在提高低分辨率和低帧率视频的空间和时间分辨率。最近，基于可变形卷积的方法已经实现了很好的 *STVSR* 性能，但是它们只能推断训练阶段中预先定义的中间帧。此外，这些方法低估了相邻帧之间的短期运动线索。在本文中，我们提出了一种时间调制网络（*TMNet*）来插值任意中间帧，并进行精确的高分辨率重构。具体来说，我们提出了一个时间调制块（*TMB*）来调制可变形卷积核，以实现可控特征插值。为了很好地利用时间信息，我们提出了一个局部时间特征比较（*LFC*）模块以及双向可变形 *ConvLSTM*，以提取视频中的短期和长期运动提示。在三个基准数据集上进行的实验表明，我们的 *TMNet* 优于以前的 *STVSR* 方法。代码开源在*https://github.com/CS-GangXu/TMNet*。

1. 引言

现如今，使用液晶显示器（LCD）或发光二极管（LED）技术的平板显示器可以以 120FPS 或 240FPS 的帧速率播放具有 4K（3840 × 2160）或 8K（7680 × 4320）全彩色像素的超高清电视（UHD TV）视频 [38]。然而，现在可用视频普遍是全高清的，

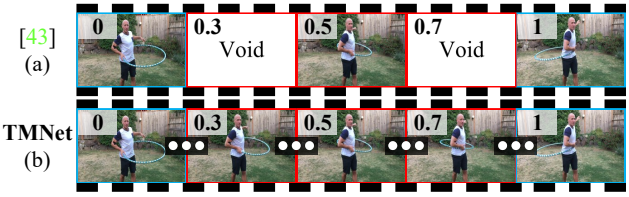

图 1: **TMNet 提供的 STVSR 灵活性能**。给 **0**（开始点）到 **1**（结束点）的输入帧序列，[44] 只能够插预先定义好的中间帧 **0.5**（a），然而我们的 TMNet 可以生成任意时刻的中间帧（例如，**0.3**, **0.5**, **0.7**）（b）。

分辨率为 2k（1920 × 1080），帧速率为 30FPS [44]。为了在超高清电视上广播全高清视频，有必要以超高清电视的广播标准恰当地增加其时空分辨率。尽管可以通过单图像超分辨率方法逐帧提高视频的空间分辨率 [4,22]，但增强后的视频感知质量会因时间失真而降低 [17]。为此，时空视频超分辨率（STVSR）方法被提出 [30,44]，以同时提高低帧率和低分辨率视频的空间和时间分辨率。

以前的基于模型的 STVSR 方法 [29–31] 严重依赖于精确的空间和时间配准 [37]，并且当配准不准确时会产生较差的重建结果 [21,24]。后来，深度卷积神经网络 [12,13,33,43] 已经在图像修复领域（例如，视频超分辨 [3,36] 率，视频插帧 [1,26,45] 和时空视频超分辨 [17,44]）中被广泛地使用。STVSR 的直接解决方案是对低分辨率和低帧率视频连续执行 VFI 和 VSR，以提高其空间分辨率和帧率 [44]。但是，这些两阶段方法忽略了时间和空间维度之间的固有关联。也就是说，具有高分辨率帧的视频包含

*Jun Xu is the corresponding author (email: nankaimathxu-jun@gmail.com). This work is supported by National Natural Science Foundation of China under Grant 62002176 and 61922046.

关于运动物体和背景的更丰富的细节，而高帧率的视频在相邻帧之间提供更精细的像素对齐 [8]。因此，这两个阶段的 STVSR 方法受到时间不一致问题的影响 [44] 而产生伪影，例如 STVSR 里面的“注意瞬脱”现象 [35]。

为了充分利用视频中时间和空间维度之间的相关性，几种单阶段 STVSR 方法被提出 [8, 17, 44]，以同时对低帧率和低分辨率视频执行 VFI 和 VSR 重建。STARnet 的工作 [8] 通过一个附加的光流分支 [5] 来估计运动线索，并执行两个相邻帧的特征变形以对中间帧进行插值。但是这种基于流的方法 [8] 需要学习一个额外的用于光流估计的分支，这会消耗昂贵的计算和内存成本。为了缓解这个问题，Xiang *et al.* [44] 使用了可变形卷积主干网络 [40]，并直接在特征空间上执行了 STVSR。尽管具有令人满意的性能，当前的 STVSR 网络只能生成网络体系结构中预定义的中间帧，因此仅限于具有固定帧速率视频的高度受控的应用场景。然而，在许多商业场景中，例如体育赛事，用户非常灵活地调整中间视频帧以实现更好的可视化是非常普遍的。因此，有必要开发用于平滑运动合成的可控 STVSR 方法。

为了满足广大场景的通用要求，在本文中，我们提出了一个时间调制网络（TMNet）来为 STVSR 内插任意数量的中间帧，如 1所示。但是当前基于可变形卷积的方法 [44] 只能生成预定义的中间帧。为了解决此问题，我们提出了一种时间调制块（TMB），将运动线索引入中间帧的特征插值中。具体来说，我们首先估计可变形卷积框架下两个相邻帧之间的运动 [40]，然后在由时间参数定义的任意时刻学习可控插值。此外，我们还提出了一个局部时间特征比较模块，以融合多帧特征以实现有效的空间对齐和特征变形，并提出一种全局时间特征融合，以探索整个视频的长期变化。这种两阶段的时间特征融合方案可以为 STVSR 准确地插补中间帧。在三个基准数据集上 [23, 34, 45] 进行的大量实验表明，我们的 TMNet 能够插补任意数量的中间帧，并在 STVSR 上达到了最新的性能。

这项工作的贡献有三方面：

- **为了灵活的 STVSR 性能，我们提出了一个时间调制网络（TMNet）来实现任意帧速率的可控插值**。这是通过可变形卷积框架下的时间调制块实现的。
- **我们提出了一种有效的 STVSR 的两阶段时间特征融合方案**。具体来说，我们提出了一个局部时空特征比较模块，以利用相邻帧的短期运动线索，并通过探索整个视频的长期变化来执行全局时空特征融合。
- 在三个基准数据集上进行的实验表明 **我们的 TMNet 能够以任意帧速率执行可控制的帧插值**，并优于最新的 STVSR 方法。

2. 相关工作

视频插帧（VFI）是为了在相邻帧之间合成新的中间帧 [2, 15, 20]。早期的 VFI 方法主要采用光流技术进行运动估计 [2, 15, 26]。Jiang *et al.* [15] 为任意帧率 VFI 建模了运动解释。Niklaus *et al.* [25] 使用上下文信息扭曲了输入帧，并插入了上下文感知的中间帧。Bao *et al.* 对 [2] 中的 VFI 进行了运动估计和补偿，并通过进一步探索深度信息获得了改进的性能 [1]。Niklaus *et al.* [26] 通过 softmax 喷溅解决了将多个像素映射到 VFI 中相同位置的冲突问题。然而，这些基于光流的方法在运动估计上需要巨大的计算成本。因此，最近的研究人员开始研究用于 VFI 的空间自适应卷积核 [27] 或可变形卷积核 [20]。

视频超分辨率（VSR）是提高低分辨率（LR）视频的空间分辨率的任务 [16, 36, 40]. 现有的 VSR 方法 [16, 36, 40] 主要是在光流技术的帮助下聚合多个帧的空间信息以进行高分辨率（HR）重建 [5]。Jo *et al.* [16] 生成了动态的上采样滤波器，以通过残差学习增强 LR 帧 [12]。Wang *et al.* [40] 提出了金字塔，级联和可变形（PCD）模块来执行帧对齐，然后通过时空关注将多个帧融合为单个帧。Haris *et al.* [7] 通过集成多个帧的时空语境设计了一个迭代优化框架。Tian *et al.* [36] 利用学习到的可变形卷积核的采样偏移量将支撑帧与参考帧对齐，它们都用于重建 HR 帧。

时空视频超分辨率（STVSR）旨在增加低帧率和低分

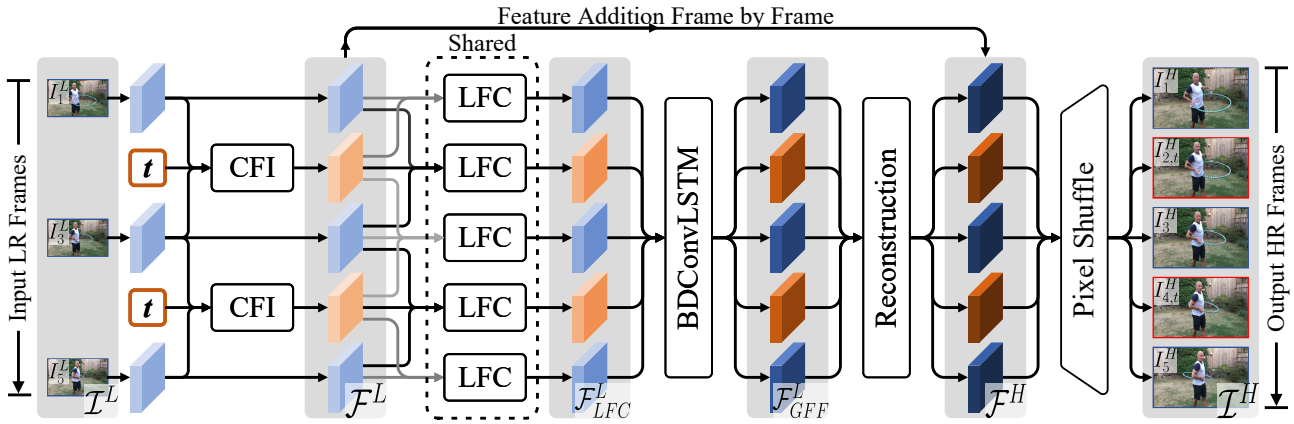

图 2: **STVSR 的时间调制网络（TMNet）整体框架**。给定输入低帧率和低分辨率视频 $\mathcal{I}^L$，我们首先提取初始特征并对中间帧在 $t \in (0,1)$ 中间的任意时刻执行可控特征插值（CFI，由我们的时间调制模块实现）。然后，我们将获得的特征图 $\mathcal{F}^L$ 送入两阶段的时间特征融合方案。为了短期运动的一致性，特征图 $\mathcal{F}^L$ 通过 LFC 模块被改善成 $\mathcal{F}^L_{LFC}$。为了利用长期运动线索，特征图 $\mathcal{F}^L_{LFC}$ 继续 GFF 模块改善，GFF 模块是通过双向可变形 ConvLSTM 实现的 [44]。最后，我们使用 40 个残差块来重建高分辨率特征图 $\mathcal{F}^H$，并使用两个 Pixel-Shuffle 层来输出高帧率和高分辨率视频 $\mathcal{I}^H$。

辨率视频的时空维度 [8,17,44]。Shechtman *et al.* [30] 通过对 HR 视频重建问题采用指向性时空平滑正则化解决 STVSR 问题。Mudenagudi *et al.* [24] 在 Markov Random Field 框架下制定了他们的 STVSR 方法 [6]。STARnet [8] 利用额外的光流分支来利用空间和时间维度之间的固有运动关系 [5]，并对两个相邻帧进行特征变形以对中间帧进行插值。Xiang *et al.* [44] 开发了一个统一的框架，通过 PCD 对齐模块插入多帧特征 [40]，通过双向可变形 ConvLSTM 插入中间特 [33] 征，最后通过多帧特征融合执行 STVSR。在这项工作中，我们的目标是为强大而灵活的 STVSR 开发一个时间可控的网络。尽管基于此 [44]，我们的 TMNet 由于提出的局部时域特征比较（LFC）模块而在基准数据集上获得了更好的性能。

调制网络。最近，研究人员提议通过附加的调制分支来控制主网络的恢复强度 [9,10,39,41]。对这些调制网络进行了训练，以权衡由超参数控制的修复质量和灵活性。He *et al.* [9] 在每个卷积层之后都放置了特征调制滤波器，以根据用户的喜好来调制输出。后来，He *et al.* [10] 将该设计扩展到了多个维度，并根据多种退化类型的水平调制了输出。Wang *et al.* [39] 从不同目标的调整块和残差块中学习特征，以控制降噪和细节保留之间的平衡。在这项工作中，我们考虑对时间维度的调制，而不是像在这些调制网络中那样对恢复强度进行调制。据我们所知，我们的工作是最早在 STVSR 问题中实现时间调制的工作之一。正如 §4中所示，我们的 TMNet 可以探索可控 STVSR 的时间调制潜力。

3. 提出的方法

在本节中，我们首先在 §3.1中概述用于 STVSR 的时间调制网络 (TMNet)。然后，我们 §3.2中在介绍了用于可控特征插值的时间调制块。我们在 §3.3中介绍时序特征融合，在 §3.4中介绍高分辨率的重建。最后，训练的细节在 §3中给出。

3.1. 网络概述

如 Fig. 2 所示，我们的 TMNet 由三个阶段组成：可控特征插值、时间特征融合和高分辨率重建。**可控特征插值**。给定一系列低帧率和低分辨率视频 $\mathcal{I}^L = \{\boldsymbol{I}^L_{2i-1}\}^n_{i=1}$，我们的 TMNet 首先通过五个残差块提取相应的初始特征图 $\{\boldsymbol{F}^L_{2i-1}\}^n_{i=1}$ 。为了执行时间可控的特征插值，我们提出了一个时间调制块 (TMB) 来调制具有时间超参数 t 的可变形卷积核。这里，$t \in (0,1)$ 表示（任意的）时刻，在这个

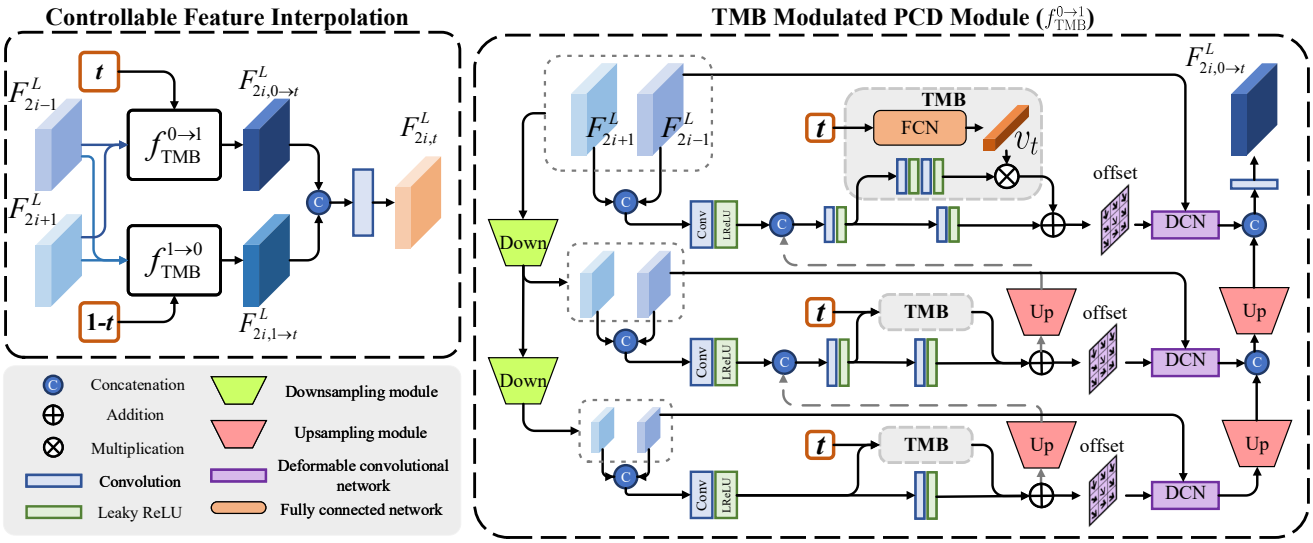

图 3: **提出的时间调制模块（TMB）**用于可控特征插值的调制 PCD 模块 [40]。$f^{0\to1}_{\text{TMB}}$（$f^{1\to0}_{\text{TMB}}$）是由我们的 TMB 模块调制的 PCD 模块，用于模拟向前（向后）运动。我们的 TMB 由三个卷积层组成的全连接网络 (FCN) 将时间超参数 t 转换为调制向量 v_t 来调制 PCD 模块的所有三个级别。

时刻我们从 $\boldsymbol{I}^L_{2i-1}$ 和 $\boldsymbol{I}^L_{2i+1}$ 两帧的特征图 $\boldsymbol{F}^L_{2i-1}$ 和 $\boldsymbol{F}^L_{2i+1}$ 中要插出一个特征图 $\boldsymbol{F}^L_{2i,t}$ 。最后，我们得到一个高帧率和低分辨率视频序列的特征序列 $\mathcal{F}^L = \{\boldsymbol{F}^L_1, \boldsymbol{F}^L_{2,t}, \boldsymbol{F}^L_3, ..., \boldsymbol{F}^L_{2n-2,t}, \boldsymbol{F}^L_{2n-1}\}$。

时间特征融合。$\mathcal{F}^L$ 中提取（或插值）的特征图通常质量较低，因为它们是从单个 LR 帧中提取的（或通过相邻 LR 帧的初始特征图进行插值）。因此，我们提出了一个局部时间特征比较（LFC）模块，借助相邻帧的特征图来微调精细化 $\mathcal{F}^L$ 中的每个特征图。局部特征细化之后，我们通过执行全局时域特征融合（GFF）进一步改善 $\mathcal{F}^L$ 中的特征图。这是通过采用双向可变形 ConvLSTM (BDConvLSTM) 网络来实现的 [44]，以沿时间方向连续聚合来自各个特征图的有用信息。LFC 和 GFF 融合模块都很好地利用了空间和时间维度之间的内部相关性来提高 $\mathcal{F}^L$ 中特征图的质量。最后，我们得到改进的特征图序列 $\mathcal{F}^L_{GFF}$。

高分辨率重建。在这里，我们将特征映射序列 $\mathcal{F}^L_{GFF}$ 送入 40 个残差块中，以提高其沿空间维度的质量。接下来，我们通过广泛使用的 Pixel-Shuffle 层 [32] 增加这些改进特征图的空间分辨率，输出最后的高帧率高分辨率的视频序列 $\mathcal{I}^H = \{\boldsymbol{I}^H_1, \boldsymbol{I}^H_{2,t}, ..., \boldsymbol{I}^H_{2n-2,t}, \boldsymbol{I}^H_{2n-1}\}$。

3.2. 可控特征插值

给定一系列低帧率和低分辨率的视频帧 $\mathcal{I}^L = \{\boldsymbol{I}^L_{2i-1}\}^n_{i=1}$，我们首先通过五个残差快提取相应的初始特征 $\mathcal{F}^L = \{\boldsymbol{F}^L_{2i-1}\}^n_{i=1}$。每个残差块包含一系列带有跳过连接的“Conv-ReLU-Conv”操作。对于任意两个相邻帧 $\boldsymbol{I}^L_{2i-1}$ 和 $\boldsymbol{I}^L_{2i+1}$ $(i \in \{1, ..., n-1\})$，我们这里的目标是插出 $t \in (0, 1)$ 中任意时刻的特征。为此，我们需要估计前向从 $\boldsymbol{I}^L_{2i-1}$ 到该中间帧运动线索，同时也需要反向 $\boldsymbol{I}^L_{2i+1}$ 到该中间帧的运动线索。之前的 STVSR 方法 [40, 44] 在可变性卷积框架下 [47] 结合了 PCD 模块来检测 $\boldsymbol{F}^L_{2i-1}$ 和 $\boldsymbol{F}^L_{2i+1}$ 之间的偏移量作为运动线索，来对齐和插出中间帧的特征。但是，朴素的 PCD 模块只能将运动估算到预定义的时刻，该时刻在训练和推理阶段均固定。

为了克服这个限制，我们提出了一个时间调制块（TMB）来调制 $\boldsymbol{F}^L_{2i-1}$ 和 $\boldsymbol{F}^L_{2i+1}$ 之间学习到的偏移量调制由超参数 $t \in (0, 1)$ 控制，表示我们计划插入新帧的任意时刻。这使我们的 TMNet 能够根据输入视频的两个相邻帧 $\boldsymbol{I}^L_{2i-1}$ 和 $\boldsymbol{I}^L_{2i+1}$ 的初始特征图 $\boldsymbol{F}^L_{2i-1}$ 和 $\boldsymbol{F}^L_{2i+1}$ 控制特征插值过程。由我们的 TMB 模块调制的 PCD 模块可以估计向前和向后运动，并在任意时刻 $t \in (0, 1)$ 插入新帧的特征图 $\boldsymbol{F}^L_{2i,t}$。

定义 $f^{0\to1}_{\text{TMB}}$ 和 $f^{1\to0}_{\text{TMB}}$ 分别表示建模前向和反向

运动的经过 TMB 块调制的 PCD 模块。在这里，我们从前向和后向方向执行调制特征插值：

$$\begin{aligned} \boldsymbol{F}^L_{2i,0\to t} &= f^{0\to 1}_{\text{TMB}}(\boldsymbol{F}^L_{2i-1}, \boldsymbol{F}^L_{2i+1}, t), \\ \boldsymbol{F}^L_{2i,1\to t} &= f^{1\to 0}_{\text{TMB}}(\boldsymbol{F}^L_{2i-1}, \boldsymbol{F}^L_{2i+1}, 1-t), \end{aligned} \tag{1}$$

这里 $\boldsymbol{F}^L_{2i,0\to t}$ 和 $\boldsymbol{F}^L_{2i,1\to t}$ 是插出来的对齐后特征。请注意，两个 TMB 调制的 PCD 模块共享相同的网络结构但具有不同的权重。这里我们仅以 $f^{0\to 1}_{\text{TMB}}$ 为例来说明我们的 TMB 调制的 PCD 模块如何对前向运动进行建模。PCD 模块 $f^{1\to 0}_{\text{TMB}}$ 建模反向运动可以类似推导。

如 Fig. 3所示，PCD 模块具有三个级别来估计不同尺度的运动。为了在时间维度上实现灵活的调制，我们将 TMB 块独立地嵌入到朴素 PCD 模块的每个级别中，以在可变形卷积网络 (DCN) 之中调制偏移。将我们的 TMB 块添加到 PCD 模块的所有三个级别的好处将在 §4 中得到验证。通过 TMB 在不同的 PCD 水平上自适应地调制偏移，我们使用三个卷积层来将时间超参数 t 映射到大小为 $1\times1\times64$ 的调制向量 $\boldsymbol{v}_t$ 上。为了更好地利用运动线索进行精确调制，我们将每个普通 PCD 级别中的特征输入到两个卷积层中，以扩大它们的感受野。然后，生成的特征与调制向量 $\boldsymbol{v}_t$ 沿通道维度相乘，以产生 TMB 调制的特征。为了稳健性，我们在 DCN 之前添加了 TMB 调制特征和相应的预调制特征。

一旦获得调制的特征图 $\boldsymbol{F}^L_{2i,0\to t}$ 和 $\boldsymbol{F}^L_{2i,1\to t}$，我们通过通道维度的连接 “$[\cdot,\cdot]$” 和一个 1×1 卷积层 $f_{1\times1}$ 插得中间帧 $\boldsymbol{F}^L_{2i,t}$ ，如下所示：

$$\boldsymbol{F}^L_{2i,t} = f_{1\times1}([\boldsymbol{F}^L_{2i,0\to t}, \boldsymbol{F}^L_{2i,1\to t}]). \tag{2}$$

现在，我们获得了高帧率低分辨率的插值序列的特征 $\mathcal{F}^L = \{\boldsymbol{F}^L_1, \boldsymbol{F}^L_{2,t}, \boldsymbol{F}^L_3, ..., \boldsymbol{F}^L_{2n-2,t}, \boldsymbol{F}^L_{2n-1}\}$。接下来，我们沿时间维度进行特征融合。

3.3. 时序特征融合

在这里，初始特征是从单个（或相邻）帧中提取（或内插）的。有相当大的空间来提高他们的质量。但我们也将初始特征输入到 TMNet 的 Pixel-Shuffle 部分。

局部时间特征比较（LFC）。保持每个当前帧的短期时间一致性至关重要。为此，我们提出了一个局部时域特征比较（LFC）模块，以利用相邻帧中的补充信息（例如运动提示）。如 Fig. 4所示，为了细化两相邻帧 $\boldsymbol{F}^L_{2i-1}$ 和 $\boldsymbol{F}^L_{2i+1}$ 当前帧的特征图 $\boldsymbol{F}^L_{2i,t}$，我们堆叠当前帧 ($\boldsymbol{F}^L_{2i,t}$) 和相邻帧 ($\boldsymbol{F}^L_{2i-1}$, $\boldsymbol{F}^L_{2i+1}$)，然后使用两个卷积层来学习可变性卷积 [47] 中的偏移量。请注意，我们学习了两个偏移量来描述向前（$\boldsymbol{I}^L_{2i-1}$ 到当前帧）和向后方向（$\boldsymbol{I}^L_{2i+1}$ 到当前帧）的运动线索。然后，使用从前向（或后向）方向学习的偏移量，通过一个可变形卷积层将前一帧（或者后一帧 $\boldsymbol{F}^L_{2i+1}$ ）与当前帧的特征图 $\boldsymbol{F}^L_{2i-1}$ 对齐。对齐后，我们将两个相邻帧的对齐特征图与当前帧的特征图连接起来，并通过四个 1×1 卷积层和加法运算进行特征比较。对于第一个（或最后一个）帧，前一个（或下一个）相邻帧就是它本身。现在我们得到一个细化的特征序列 $\mathcal{F}^L_{LFC}$。

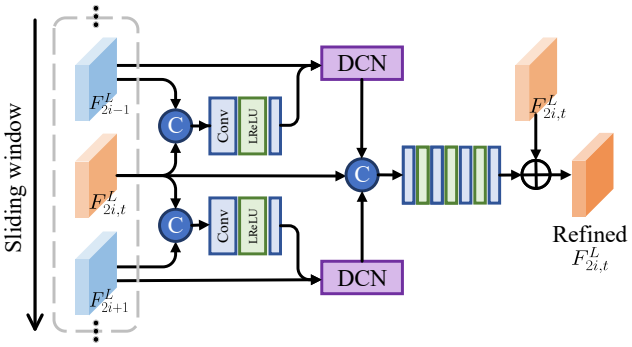

图 4: **提出的局部时间特征比较** (LFC) 模块通过利用相邻帧之间的短期运动线索来细化插值特征 $\boldsymbol{F}^L_{2i,t}$。

全局时间特征融合。我们的 LFC 模块细化的特征序列能够在插值视频中保持短期一致性。但它会在大运动或快速运动上失败，因为 LFC 缺乏对整个视频的运动进行建模的能力。为了解决这个问题，我们建议通过全局时间特征融合来利用视频中的长期信息。受到 [44] 的启发，我们把 LFC 生成的特征序列 $\mathcal{F}^L_{LFC}$ 输入到双向可变 ConvLSTM 网络中，并且得到具有长期时序一致性的特征 $\mathcal{F}^L_{GFF}$。

正如实验部分将说明的那样，我们的短期 LFC 模块和长期 BDConvLSTM 确实提高了我们 TMNet 在 STVSR 上的性能。

3.4. 高分辨率重建

到目前为止,已经很好地探索了时间和空间维度的内相关以获得整个视频的高质量特征序列 $\mathcal{F}_{GFF}^{L}$。然后，我们通过 40 个残差块对特征图进行空间细化，得到细化的特征图 $\mathcal{F}^{H}$。然后我们将特征 $\mathcal{F}^{H}$ 与 $\mathcal{F}^{L}$ 中对应的初始特征图相加，得到重构的特征图 $\mathcal{F}_{final}^{H}$。最后，我们将重建的特征图 $\mathcal{F}_{final}^{H}$ 馈入两个 Pixel-Shuffle 层，然后进行一系列的"Conv-LeakyReLU-Conv" 操作，以输出重建的特征图 HR 视频帧 $\mathcal{I}^{H} = \{\boldsymbol{I}_{1}^{H}, \boldsymbol{I}_{2,t}^{H}, ..., \boldsymbol{I}_{2n-2,t}^{H}, \boldsymbol{I}_{2n-1}^{H}\}$。

3.5. 训练细节

实现细节。我们使用 Adam 优化器 [18]，参数 $\beta_1 = 0.9$，$\beta_2 = 0.999$，损失函数为 Charbonnier loss function [19]。学习率初始化为 4×10^{-4}，然后它每 150,000 个迭代衰减 1×10^{-7}。使用余弦退火算法。我们通过 Kaiming 初始化 [11] 在没有预训练权重的情况下初始化 TMNet 的参数。Batchsize 是 24。我们的 TMNet 在 PyTorch [28] 和 Jittor [14] 中实现，在四个 RTX 2080Ti GPU 上总共训练了 600,000 次迭代，大约需要 8.71 天（209.04 小时）。对于每个输入视频剪辑，我们随机将其裁剪成一系列大小为 32×32 的下采样图像块。对于数据增强，我们水平翻转每一帧，并用 90°、180° 或 270° 随机旋转它。

网络训练。当直接使用建议的 TMB 块进行训练时，我们的 TMNet 在 STVSR 上的性能明显下降，如我们的实验所示。一个可能的原因是我们的 TMNet 无法准确估计运动线索以在任意时刻 $t \in (0, 1)$ 内插中间帧，因为我们的 TMB 不知道训练调制特征之前的时刻。为了解决这个问题，我们建议通过两步策略来训练我们的 TMNet：步骤 1，我们在没有提出的 TMB 块的情况下训练我们的主 TMNet；步骤 2，我们只训练我们的 TMB 块，同时完善训练好的主网络。

在步骤 1 中, 我们在 Vimeo-90K 数据集 [45] 上训练我们的 TMNet，这将在 §4.1 中介绍。该数据集由 7 帧视频剪辑组成。对于每个视频片段，第 1、第 3、第 5 和第 7 帧 LR 帧作为低帧率和低分辨率视频输入到我们的 TMNet。我们设置 $t = 0.5$ 从我们的 TMNet 中去除 TMB 块，并学习生成 7 帧高分辨率和高帧率视频。这使我们的 TMNet 能够与以前的 STVSR 方法进行公平比较 [8, 34, 40, 44]。为了监督学习，我们计算 Vimeo-90K 数据集中相应 7 帧 HR 视频剪辑的损失函数 [45]。

在步骤 2 中，我们完善了主网络的学习权重，并且只训练 TMB 块进行时间调制。训练是在 Adobe240fps 数据集上进行的，它是高帧率的，适合训练 TMB 块。我们还把它分成 7 帧视频片段组。对于每个视频片段，第 1 和第 7 帧 HR 被下采样作为 TMNet 的输入。我们设置时间超参数 $t \in \{\frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}\}$ 来插入 5 个中间帧。此步骤需要 35.26 分钟。

4. 实验

4.1. 实验准备

数据集。我们使用 Vimeo-90K `septuplet` 数据集 [45] 作为训练集。它包含 91,701 个视频序列，这些序列是从选自 Vimeo-90K 的 39K 视频片段中提取的。每个序列包含 7 个连续帧，分辨率为 448×256。Vid4 [23] 和 Vimeo-90K `test` 集用作评估数据集。正如 [44] 中所建议的，我们将 Vimeo-90K `septuplet` `test` 集分成 `Fast` 运动、`Medium` 运动和 `Slow` 运动的三个子集，其中分别包括 1225、4977 和 1613 个视频片段。我们还从原始 `Medium` 运动集中删除了 5 个视频片段，并从 `Slow` 运动集中删除了 3 个视频片段，它们仅包含全黑背景。

为了使我们的 TMNet 可用于可控特征插值，我们在 Adobe240fps 数据集 [34] 上单独训练我们的 TMB 块。它有 133 个用手持相机拍摄的视频(720P)，并随机分成 `train`、`val` 和 `test` 子集，分别有 100、16 和 17 个视频。对于每个视频，我们将其分成 7 帧视频片段组。我们将每个片段中的第 1 和第 7 帧输入到我们的 TMNet 中，以生成 5 个中间帧。

我们通过双三次插值对 HR 帧进行降采样以创建 LR 帧，其倍数为 4。

评价指标。我们采用广泛使用的峰值信噪比 (PSNR)

表 1: **不同 STVSR 方法的 PSNR, SSIM [42], fps 的速度, 和参数数量（单位百万）对比** on Vid4 [34], Vimeo-Fast, Vimeo-Medium, Vimeo-Slow [45]。"↑" 表示越大越好。速度是在 Vid4 [34] 上面测试的。最好，次好，第三好的结果都分别用红色，蓝色和黑色粗体标出来了。

Method	Vid4 [34]		Vimeo-Fast		Vimeo-Medium		Vimeo-Slow		Speed	Parameters
VFI+(V)SR / STVSR	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	fps↑	million↓
SuperSloMo [15] + Bicubic	22.84	0.5772	31.88	0.8793	29.94	0.8477	28.37	0.8102	-	**19.8**
SuperSloMo [15] + RCAN [46]	23.80	0.6397	34.52	0.9076	32.50	0.8884	30.69	0.8624	2.49	19.8+16.0
SuperSloMo [15] + RBPN [7]	23.76	0.6362	34.73	0.9108	32.79	0.8930	30.48	0.8584	2.06	19.8+12.7
SuperSloMo [15] + EDVR [40]	24.40	0.6706	35.05	0.9136	33.85	0.8967	30.99	0.8673	6.85	19.8+20.7
SepConv [27] + Bicubic	23.51	0.6273	32.27	0.8890	30.61	0.8633	29.04	0.8290	-	21.7
SepConv [27] + RCAN [46]	24.92	0.7236	34.97	0.9195	33.59	0.9125	32.13	0.8967	2.42	21.7+16.0
SepConv [27] + RBPN [7]	26.08	0.7751	35.07	0.9238	34.09	0.9229	32.77	0.9090	2.01	21.7+12.7
SepConv [27] + EDVR [40]	25.93	0.7792	35.23	0.9252	34.22	0.9240	32.96	0.9112	6.36	21.7+20.7
DAIN [1] + Bicubic	23.55	0.6268	32.41	0.8910	30.67	0.8636	29.06	0.8289	-	24.0
DAIN [1] + RCAN [46]	25.03	0.7261	35.27	0.9242	33.82	0.9146	32.26	0.8974	2.23	24.0+16.0
DAIN [1] + RBPN [7]	25.96	0.7784	35.55	0.9300	34.45	0.9262	32.92	0.9097	1.88	24.0+12.7
DAIN [1] + EDVR [40]	**26.12**	0.7836	35.81	0.9323	34.66	0.9281	**33.11**	0.9119	5.20	24.0+20.7
STARnet [8]	26.06	**0.8046**	**36.19**	**0.9368**	**34.86**	**0.9356**	33.10	**0.9164**	**14.08**	111.61
Zooming Slow-Mo [44]	**26.31**	**0.7976**	**36.81**	**0.9415**	**35.41**	**0.9361**	**33.36**	**0.9138**	**16.50**	**11.10**
TMNet (Ours)	**26.43**	**0.8016**	**37.04**	**0.9435**	**35.60**	**0.9380**	**33.51**	**0.9159**	**14.69**	**12.26**

图 5: **STVSR 不同方法的客观和主管的结果**。测试视频片段来自 Adobe240fps [34] (1-st row), Vimeo-Fast [45] (2-nd row, left) 和 Vimeo-Slow [45] (2-nd row, right) 数据集。

和结构相似性指数 (SSIM) [42] 来评估 STVSR 任务的不同方法。PSNR 和 SSIM 指标是在 YCbCr 颜色空间的 Y 通道上计算的，这是之前 VSR [7, 40] 和 STVSR [44] 方法所青睐的。

4.2. 与最先进技术的比较

比较方法。我们将我们的 TMNet 与最先进的两阶段和一阶段 STVSR 方法进行比较。对于两阶段 STVSR 方法，我们通过 SuperSloMo🄵 cite jiang2018super，DAIN🄵 cite bao2019depth 或 SepConv🄵 cite niklaus2017video 执行视频帧插值 (VFI)，并执行视频超分辨率 (VSR) 通过双三次插值 (BI)、RCAN [46]、RBPN [7] 或 EDVR [40]。对于单阶段 STVSR 方法，我们将我们的 TMNet 与 Zooming SlowMo [44] 和 STARnet [8] 进行比较。为了与这些竞争对手公平比较，我们在 TMNet 中设置 $t = 0.5$ 以在任何两个相邻帧的中间时刻生成帧。也就是说，将 Vimeo-90K 中每个剪辑的第 1、第 3、第 5 和第 7 帧 LR 帧输入我们的 TMNet 以重建 7 个 HR 帧。所有这些方法都在 Vimeo-90K `septuplet` 数据集 [45] 上训练，在 Vimeo-90K `test` 集 [45] 和

Vid4 [34] 数据集上进行评估。

客观结果。我们在 Table 1 中列出了定量比较结果。正如 [44] 中所建议的，我们在比较速度时省略了使用双三次插值的基线模型。可以看到我们模型相较于 Zooming SlowMo [44] 在 Vid4, Vimeo-`Fast`, Vimeo-`Medium`, 和 Vimeo-`Slow` 数据集上 PSNR 分别提升了 0.12dB, 0.23dB, 0.19dB, 和 0.15dB。在 SSIM [42] 上，我们的 TMNet 在大多数情况下取得了更好的结果，但在 Vid4 [23] 和 Vimeo-`Slow` 上仅略逊于 STARnet [8]。然而，我们的 TMNet 只需要 STARnet 中参数的九分之一。在速度上，一阶段方法 [8, 44] 要快于二阶段的方法 [1, 7, 15, 27, 40]。TMNet 速度是 14.69fps，仅慢于 Zooming Slow-Mo [44]。所有这些结果都验证了我们的 TMNet 在 STVSR 上的有效性。

可视化。在 Figure 5的第一行，我们展示了 TMNet 插出来的五帧 (帧 1550 到帧 1554)，以 Adobe240fps `test` set [34] 数据集上的视频序列 "0056" 的帧 1549 和帧 1555 作为输入。可以看出，我们的 TMNet 能够为 STVSR 执行灵活的帧插值。在图 5 的第 2 行中，我们展示了来自 Vimeo-`Fast` 和 Vimeo-`Slow` 数据集 [45] 的不同 STVSR 方法重建的帧。我们观察到，带有提出的 LFC 模块的 TMNet 可以比竞争对手更清晰地恢复结构和纹理。例如，在 Vimeo-`Slow` 数据集序列 "0070" 中的视频片段 "0001" 上，我们的 TMNet 清楚地重建了包上的纹理图案。总之，我们的 TMNet 在数量和质量上都展示了灵活而强大的 STVSR 能力。因为页面限制，*Supplementary File* 中提供了对 Vid4 [23]、Vimeo-90K `test` set [34] 和 Adobe240fps [45] 数据集的更多视觉比较。

4.3. 消融实验

在这里，我们对 STVSR 上的 TMNet 进行了详细检验。具体来说，我们评估 1) 时间调制块 (TMB) 对于可控特征插值的重要性；2) TMB 模块调制 PCD 模块的不同策略；3) 如何设计 TMB 块；4) 局部时间特征比较 (LFC) 模块如何有助于 TMNet 中的时间特征融合；5) STVSR 的高质量特征图 $\mathcal{F}^H$ 和初始特征图 $\mathcal{F}^L$ 的组合。

1. 我们的 TMB 块是否有助于可控特征插值？为了回答这个问题，我们将我们的 TMNet 与之前的 STVSR 方法进行比较 [8, 44] 从两个相邻帧生成中间帧。由于篇幅有限，我们在 *Supplementary File* 中提供了 Adobe240fps `test` set [34] 上的视觉效果对比。我们观察到我们的带有 TMB 块的 TMNet 确实表现出时间可控的 STVSR 性能。

2. 我们的 TMB 模块调制 PCD 模块的不同策略如何影响我们在 STVSR 上的 TMNet？PCD 模块 [40] 具有三层金字塔结构：第一层 `L1`；第 2 级 `L2` 从 `L1` 中的特征中通过卷积滤波器以 2 的步幅进行下采样；类似地，第 3 级 `L3` 从 `L2` 下采样，步长为 2。在我们的 TMNet 中，提出的 TMB 调制 PCD 模块的所有三个级别。但是我们的 TMB 也只能调制 PCD 的一个级别 (`L1`、`L2` 或 `L3`)，从而导致我们 TMNet 的三个变体称为 TMB-`L1`、TMB- texttttL2 和 TMB-`L3`。这些变体在 Adobe240fps 训练集 [34] 上进行训练，并在 测试集上对其进行评估。如表 2所示，三个变体按降序执行，表明 PCD 的第一级对于时间调制更重要。通过调制所有三个级别的 PCD，我们的 TMNet 通过更好地利用视频的运动线索，在 STVSR 上的表现优于三个变体。

表 2: **由我们的 TMB 调制 PCD 的不同策略设置的 Adobe240fps 测试集上的 PSNR 结果**。

Variant	TMB-L1	TMB-L2	TMB-L3	TMNet
PSNR (dB)	26.92	26.82	26.60	**26.95**

3. 如何设计我们的 TMB 块？我们 TMB 的目标是将超参数 t 转换为适合 PCD 模块的调制向量 v_t。我们 TMB 的一个简单设计是一个线性卷积层。我们称之为 TMB-`Linear`。我们在 Adobe240fps 训练集 [34] 上训练我们的 TMB 和 TMB-`Linear`，并在 测试集上评估它们。PSNR 结果列在 Table 3 中，其中 TMB-`Linear` 比我们具有三个非线性卷积层的 TMB 低 0.02dB。这表明非线性变换只比线性变换好一点。

表 3: **TMB 使用线性或非线性设计设置的 Adobe240fps 测试集上的 PSNR 结果**。

Variant	TMB-Linear	TMB
PSNR (dB)	26.93	26.95

4. 提出的 LFC 模块对我们的 TMNet 有多重要? 我们的 TMNet 执行两阶段时间特征融合：首先通过 LFC 进行局部融合，然后通过 GFF 进行全局融合。因此，我们的 TMNet 可以称为 "LFC→GFF"。颠倒顺序，例如，GFF→LFC，使我们的 TMNet 在训练期间崩溃。主要原因是在 LFC 之前执行 GFF 带来了噪声长期信息，混淆了 LFC 中可变形卷积的学习。因此，我们不评估此变体。为了研究我们的 LFC 如何为我们的 TMNet 中的两阶段融合做出贡献，我们从 TMNet 中删除了 LFC 并将此变体称为 "GFF"。此外，来自 LFC 和 GFF 的特征可以通过卷积层连接和融合，从而产生一个变体 "LFC+GFF"。这些变体都在 Vimeo-90K septuplet 数据集上训练并且在 Vid4 [23], Vimeo-Fast, Vimeo-Medium, and Vimeo-Slow 数据集上测试。PSNR 结果列在 Table 4 中。可以看出 TMNet(LFC→GFF) 得到了最好的结果，在数据集 Vid4，Vimeo-Fast，Vimeo-Medium 和 Vimeo-Slow 上分别提升了 0.07dB，0.17dB，0.15dB 和 0.11dB。这表明我们的 LFC 模块通过利用相邻帧之间的短期运动线索，对于我们在 STVSR 上的 TMNet 的成功至关重要。

表 4: **我们 TMNet 的不同变体在 STVSR 数据集上的 PSNR (dB) 结果比较。**

Variant	GFF	LFC+GFF	LFC→GFF
Vid4 [23]	26.36	26.35	**26.43**
Vimeo-Fast	36.87	36.90	**37.04**
Vimeo-Medium	35.45	35.47	**35.60**
Vimeo-Slow	33.40	33.43	**33.51**

5. 将高质量特征映射 $\mathcal{F}^H$ 和初始特征映射 $\mathcal{F}^L$ 结合用于 STVSR 的好处。在我们的 TMNet 中，我们将高质量特征 $\mathcal{F}^H$ 与最终 STVSR Pixel-Shuffle 层之前的初始特征 $\mathcal{F}^L$ 结合起来。由于初始特征 $\mathcal{F}^L$ 在很大程度上影响我们的 LFC 模块，我们将它们从 TMNet 中删除并获得一个变体 "Baseline"。然后我们加入 $\mathcal{F}^L$ 到 "Baseline"，得到一个变体模型 "$+\mathcal{F}^L$"。TMNet 和一个变体在 Vimeo-90K 数据集上训练，并且在 Vimeo-90K 测试集和 Vid4 上测试。如表 5所示，变体 "$+\mathcal{F}^L$" 明显超过 "Baseline"。这验证了将高质量特征 $\mathcal{F}^H$ 与初始特征 $\mathcal{F}^L$ 相结合对我们在 STVSR 上的 TMNet 有帮助。

表 5: **TMNet 及其变体在不同 STVSR 数据集上的 PSNR (dB) 比较。**

Variant	Baseline	$+\mathcal{F}^L$	TMNet
Vid4 [23]	26.33	26.36	**26.43**
Vimeo-Fast	36.75	36.87	**37.04**
Vimeo-Medium	35.35	35.45	**35.60**
Vimeo-Slow	33.28	33.40	**33.51**

5. 结论

在这项工作中，我们提出了一个时间调制网络（TMNet）来灵活地为时空视频超分辨率（STVSR）插入中间帧。具体来说，我们引入了一个时间调制块来调制可变形卷积框架的学习，以实现可控特征插值。为了很好地利用运动线索，我们分别执行了由我们提出的局部时间特征比较 (LFC) 模块和双向可变形 ConvLSTM 组成的短期和长期时间特征融合。在三个基准上的实验证明了我们的 TMNet 在插入中间帧方面的灵活性、我们的 TMNet 相对于以前方法的定量和定性优势，以及我们的 LFC 模块对于 STVSR 的有效性。

参考文献

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3703–3712, 2019. 1, 2, 7, 8, 14, 19, 20, 21, 22

[2] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019. 2

[3] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4778–4787, 2017. 1

[4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convo-

lutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(2):295–307, 2015. 1

[5] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Int. Conf. Comput. Vis.*, December 2015. 2, 3

[6] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, 1984. 3

[7] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3897–3906, 2019. 2, 7, 8, 14, 19, 20, 21, 22

[8] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Space-time-aware multi-resolution video enhancement. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2, 3, 6, 7, 8, 14, 19, 20, 21, 22

[9] Jingwen He, Chao Dong, and Yu Qiao. Modulating image restoration with continual levels via adaptive feature modification layers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11056–11064, 2019. 3

[10] Jingwen He, Chao Dong, and Yu Qiao. Multi-dimension modulation for image restoration with dynamic controllable residual learning. *arXiv preprint arXiv:1912.05293*, 2019. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Int. Conf. Comput. Vis.*, pages 1026–1034, 2015. 6, 13

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 1, 2

[13] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 1

[14] Shi-Min Hu, Dun Liang, Guo-Ye Yang, Guo-Wei Yang, and Wen-Yang Zhou. Jittor: a novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences*, 63, 222103, 2020. 6

[15] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9000–9008, 2018. 2, 7, 8, 14, 19, 20, 21, 22

[16] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3224–3232, 2018. 2

[17] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. Fisr: Deep joint frame interpolation and super-resolution with a multi-scale temporal loss. In *Association for the Advancement of Artificial Intelligence*, pages 11278–11286, 2020. 1, 2, 3

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2015. 6

[19] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 624–632, 2017. 6

[20] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5316–5325, 2020. 2

[21] Tao Li, Xiaohai He, Qizhi Teng, Zhengyong Wang, and Chao Ren. Space-time super-resolution with patch group cuts prior. *Signal Processing: Image Communication*, 30:147–165, 2015. 1

[22] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 136–144, 2017. 1

[23] Ce Liu and Deqing Sun. A bayesian approach to adaptive video super resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 209–216. IEEE, 2011. 2, 6, 8, 9, 13, 14

[24] Uma Mudenagudi, Subhashis Banerjee, and Prem Kumar Kalra. Space-time super-resolution using graph-cut optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):995–1008, 2010. 1, 3

[25] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1701–1710, 2018. 2

[26] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5437–5446, 2020. 1, 2

[27] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Int. Conf. Comput. Vis.*, pages 261–270, 2017. 2, 7, 8, 14, 19, 20, 21, 22

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Inform. Process. Syst.*, 2019. 6

[29] Oded Shahar, Alon Faktor, and Michal Irani. Space-time super-resolution from a single video. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011. 1

[30] Eli Shechtman, Yaron Caspi, and Michal Irani. Increasing space-time resolution in video. In *Eur. Conf. Comput. Vis.*, pages 753–768. Springer, 2002. 1, 3

[31] Eli Shechtman, Yaron Caspi, and Michal Irani. Space-time super-resolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4):531–545, 2005. 1

[32] Wenzhe Shi, Jose Caballero, Ferenc Huszar, 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1874–1883, 2016. 4

[33] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Adv. Neural Inform. Process. Syst.*, pages 802–810, 2015. 1, 3

[34] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1279–1288, 2017. 2, 6, 7, 8, 13, 14

[35] Matthew F Tang, Lucy Ford, Ehsan Arabzadeh, James T Enns, Troy AW Visser, and Jason B Mattingley. Neural dynamics of the attentional blink revealed by encoding orientation selectivity during rapid visual presentation. *Nature Communications*, 11(1):1–14, 2020. 2

[36] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3360–3369, 2020. 1, 2

[37] Roger Y. Tsai and Thomas S. Huang. Multiframe image restoration and registration. In *Advances in Computer Vision and Image Processing*, pages 317–339, 1984. 1

[38] ETSI TS 101 154 V2.3.1. *Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream.* ETSI, Feb. 2017. 1

[39] Wei Wang, Ruiming Guo, Yapeng Tian, and Wenming Yang. Cfsnet: Toward a controllable feature space for image restoration. In *Int. Conf. Comput. Vis.*, pages 4140–4149, 2019. 3

[40] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 0–0, 2019. 2, 3, 4, 6, 7, 8, 13, 14, 19, 20, 21, 22

[41] Xintao Wang, Ke Yu, Chao Dong, Xiaoou Tang, and Chen Change Loy. Deep network interpolation for continuous imagery effect transition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1692–1701, 2019. 3

[42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 7, 8

[43] Yu-Huan Wu, Shang-Hua Gao, Jie Mei, Jun Xu, Deng-Ping Fan, Rong-Guo Zhang, and Ming-Ming Cheng. Jcs: An explainable covid-19 diagnosis system by joint classification and segmentation. *IEEE Transactions on Image Processing*, 30:3113–3126, 2021. 1

[44] 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 *IEEE Conf. Comput. Vis. Pat-*

tern Recog., pages 3370–3379, June 2020. 1, 2, 3, 4, 5, 6, 7, 8, 13, 14, 17, 18, 19, 20, 21, 22

[45] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *Int. J. Comput. Vis.*, 127(8):1106–1125, 2019. 1, 2, 6, 7, 8, 13, 14, 17, 20, 21, 22

[46] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Eur. Conf. Comput. Vis.*, pages 286–301, 2018. 7, 14, 19, 20, 21, 22

[47] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9308–9316, 2019. 4, 5

Appendices

1. 内容

在此补充文件中，我们提供了用于时空视频超分辨率 (STVSR) 的时间调制网络 (TMNet) 的更多详细信息。具体来说，我们提供

- TMNet 具体网络结构在 §2;
- 更多的训练细节在 §3;
- TMNet 在插入任意数量的中间帧时的灵活性在 §4;
- TMNet 与之前的 STVSR 方法的更多视觉比较在 §5;
- 单阶段训练(而不是两阶段)如何影响使用 TMB 的 TMNet 在 §6。

2. 我们 TMNet 的详细网络结构

在这里，我们在图 6 中说明了我们提出的 TMNet 的详细网络架构。我们首先通过五个残差块来提取对应的最初始的特征 $\mathcal{F}^L = \{\boldsymbol{F}^L_{2i-1}\}^n_{i=1}$。每个残差块包含一系列带有跳过连接的 “Conv-ReLU-Conv” 操作。可控特征插值 (CFI) 主要由 PCD 模块 [40] 组成，该模块由我们提出的时间调制块 (TMB) 调制，如图 2 所示。我们的 TMB 块的详细结构显示在 7 (右侧)。提出的局部时间特征比较 (LFC) 模块在 Figure 7 (左侧)。BDConvLSTM 部分直接采用 Bi-directional Deformable ConvLSTM 网络实现 [44]。Upsampling 部分包含两个 “Convolutions (Conv), Pixel-Shuffle, and LeakyReLU”” 和一个 “Conv-LeakyReLU-Conv” 的操作。

3. 更多的训练细节

在这里，我们为我们的 TMNet 提供了两步训练策略的更多细节。

在步骤 1，我们使用 Vimeo-90K `septuplet` dataset [45] 作为训练集，Vid4 [23]，Vimeo-`Fast`，Vimeo-`Medium`，和 Vimeo-`Slow` 作为测试集。我们对所有原始 HR 帧进行下采样，以通过双三次插值获得低分辨率 (LR) 输入帧，系数为 4。训练 TMNet 时，我们通过不进行预训练权重的 Kaiming 初始化 [11] 来初始化 TMNet 的参数。我们设置 $t = 0.5$ 以去除 TMB 块，并将每个序列的第 1、3、5 和 7 帧 LR 帧作为低帧率和低分辨率输入视频来训练我们的 TMNet。因此，在 Vimeo-90K `septuplet` 数据集 [45] 中相应 7 帧 HR 视频序列的监督下，我们的 TMNet 可以学习生成 7 帧高分辨率和高帧率视频序列。训练我们的 TMNet 600,000 次迭代需要花费 8.71 天 (209.04 小时)。

在步骤 2, 我们完善了在 `Step 1` 中学到的主网络的权重，并且只训练我们的 TMB 块进行时间调制。在这里，我们在 Adobe240fps 数据集 [34] 上训练我们的 TMNet，它有 133 个 720P 高帧率 (240fps) 视频。我们首先以 1280×720 的分辨率对原始 HR 帧进行 2 倍下采样，并将它们作为真实值 (GT)。然后我们对 GT 进行下采样以创建相应的 LR 输入帧，系数为 4。所有下采样操作都是通过双三次插值执行的。每个视频序列的第 1 帧和第 7 帧 LR 帧被输入到我们的 TMNet。我们设置时序超参数 $t \in \{\frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}\}$ 来插出 5 帧。在 Adobe240fps `test` 中设置为 GT 的相应 7 帧 HR 视频序列的监督下，我们的 TMNet 能够根据时间超参数灵活地插入中间帧。在 Adobe240fps 数据集 [34] 上训练我们的 TMB 块进行 1,500 次迭代需要 35.26 分钟。

4. 在插入任意数量的中间帧时的灵活性

为了展示我们的 TMNet 在 STVSR 上插入任意数量的中间帧的灵活性，我们使用多个时间超参数 t 在输入的两个帧之间提供了我们的 TMNet 生成的结果。由于 Adobe240fps [34] 数据集中的运动极其缓慢，我们在 Vimeo-90K 数据集 [45] 上验证了我们的 TMNet 的灵活性。为此，我们设置时间超参数 $t \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ 在任意两个相邻的中间帧之间插值帧，尽管我们的 TMNet 被训练为在第 1 帧和第 7 帧之间插入 5 个中间帧。结果如图 8所示。可以看到插值帧随着 t 从 0.1 到 0.9 的变化而不断变化。这表明我们的 TMNet 可以生成

多个中间帧。也就是说，根据时间超参数 $t \in (0,1)$，我们的 TMNet 在插入任意数量的中间帧时非常灵活。

在图 9 中，我们在 Vimeo-Fast 集的视频片段 0277 的“00006”上可视化了我们的 TMNet 和 Zooming SlowMo [44] 的时间一致性 [45]。我们的 TMNet 插值了 9 帧，而 Zooming Slow-Mo [44] 在第 1 帧和第 3 帧之间插值了 1 帧。为了说明视频的时间运动，我们从左图所示的红线上提取整个帧的一维像素向量，并将一维像素向量连接成二维图像。我们观察到我们的 TMNet（图 9，右上）比 Zooming SlowMo [44]（图 9，右下）产生更一致的时间运动轨迹，Zooming SlowMo [44] 有明显的断裂变化。这证明了我们的 TMNet 在 STVSR 的灵活帧插值方面的优越性。

5. STVSR 上的更多视觉比较

在 Vid4 [23] 和 Vimeo-90K [34] 数据集上，我们将我们的 TMNet 与之前的一阶段和两阶段 STVSR 方法进行了比较。对于单阶段 STVSR 方法，我们将 TMNet 与 Zooming SlowMo [44] 和 STARnet [8] 进行比较。对于两阶段 STVSR 方法，我们通过 SuperSloMo [15]、DAIN [1] 或 SepConv [27] 执行视频帧插值（VFI），并经过 RCAN [46], RBPN [7], 或者 EDVR [40] 执行视频超分辨率（VSR ）。我们在 TMNet 中设置 $t = 0.5$ 以在任意两个相邻帧的中间时刻生成帧，这意味着每个剪辑的第 1、第 3、第 5 和第 7 帧 LR 帧 Vimeo-90K 被送入我们的 TMNet 以重建 7 个 HR 帧。所有这些方法都在 Vimeo-90K `septuplet` 数据集 [45] 上进行训练，并在 Vimeo-90K 测试集 [45] 和 Vid4 [34] 上进行评估。对比结果的可视化结果如图 10-13所示。

6. 一阶段训练 TMNet

尽管通过两步方案进行训练，但我们的 TMNet 可以直接使用提出的 TMB 块进行训练，从而产生一步训练方案。也就是说，在这个一步方案中，我们的主 TMNet 和 TMB 块的所有参数都在没有预训练的情况下同时优化。在我们的两步方案中，分别对主 TMNet 和 TMB 块中的两组参数进行了优化（首先是主 TMNet，然后是 TMB 块）。在这里，我们比较了经过训练的 TMNet 与我们的两步和一步方案的性能，产生了两个变体，分别称为 TMNet-`two`（原始 TMNet）和 TMNet-`one`。两种变体都在 Adobe240fps 训练集 [34] 上进行训练，并在 Adobe240fps `test` 测试 [34] 上进行评估。如表 6 所示，与我们的 TMNet-`two` 相比，变体 TMNet-`one` 在 Adobe240fps 上的 PSNR 性能下降了 1.84dB。这表明我们在一步方案中训练的 TMNet 无法估计运动线索，并在任意时刻插入中间帧 $t \in (0,1)$。主要原因是，在初始训练迭代中，我们从头开始训练的带有 TMB 的 TMNet 无法从视频中提取有用的运动线索，因此无法在任意时刻为有意义的特征优化 TMB 块的参数 t。

表 6: **一步或者两步训练出来的 TMNet 的 PSNR 结果** Adobe240fps 测试集 [34]。

Variant	TMNet-`one`	TMNet-`two`
PSNR (dB)	25.11	26.95

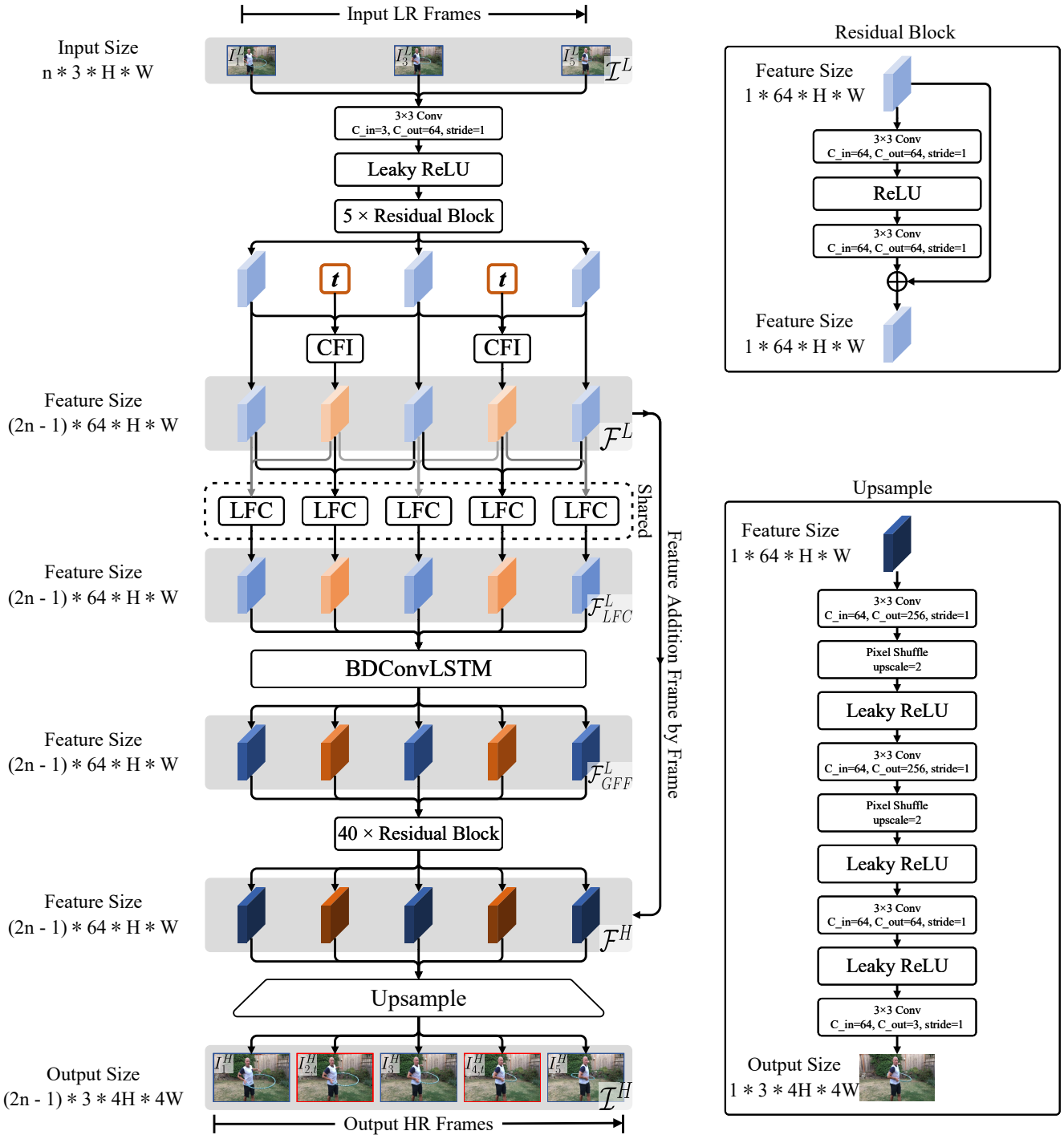

图 6: **TMNet 的主要结构**。“残差块”和“上采样”的基本部分在右侧说明。n 是输入帧的数量。H 和 W 表示图像或特征图的高度和宽度。C_in 和 C_out 分别表示输入和输出通道的数量。

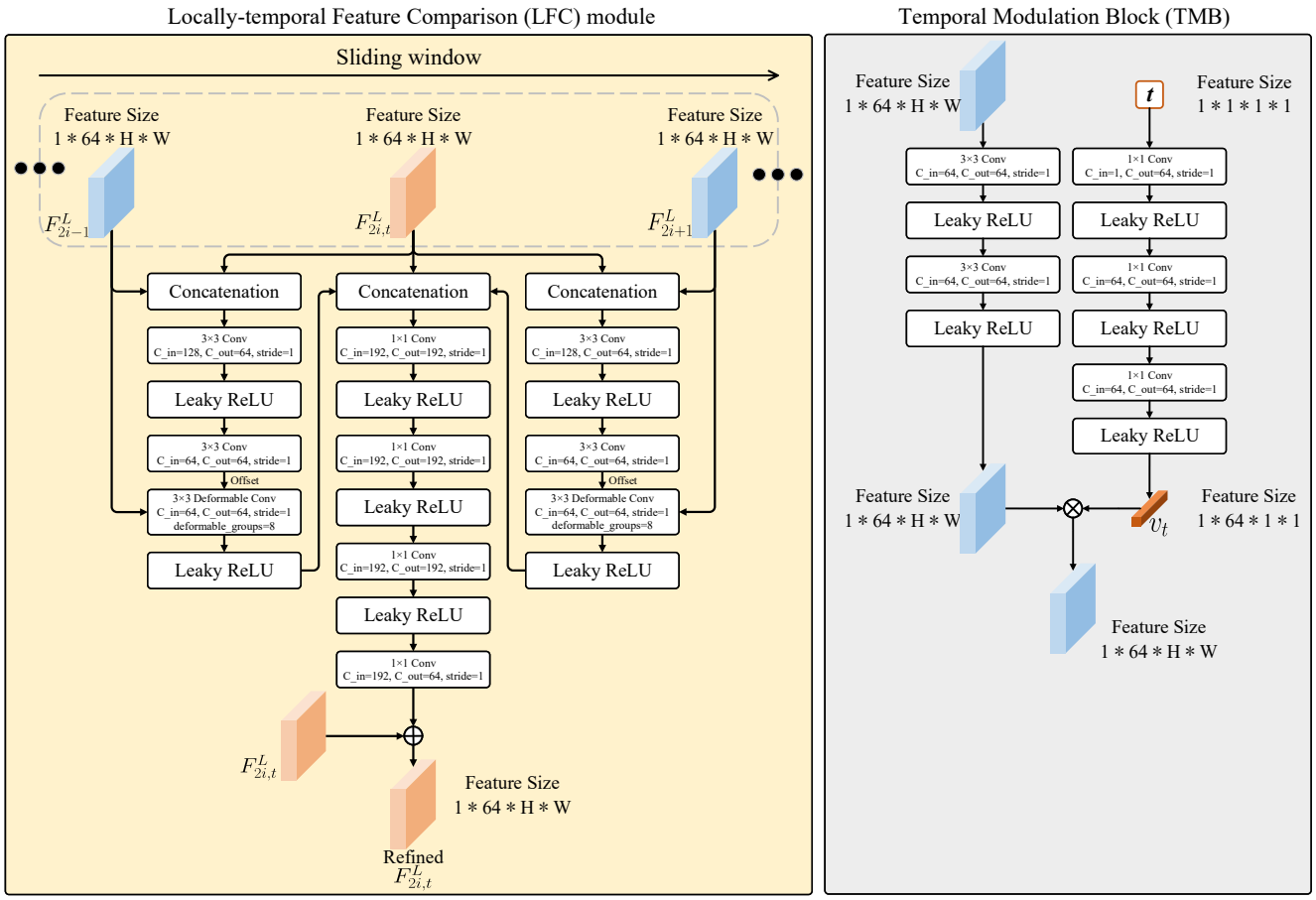

图 7: **局部时间特征比较（LFC）模块（左）和时间调制块（TMB）（右）的详细结构**。$2i-1$、$2i$ 和 $2i+1$ 是帧的索引。H 和 W 表示图像或特征图的高度和宽度。C_in 和 C_out 分别表示输入和输出通道的数量。

图 8: **我们的 TMNet（第 1、第 3 和第 5 列）和 Zooming Slow-Mo [44]（第 2、第 4 和第 6 列）在 STVSR 上的灵活性比较，所用测试视频片段为 Vimeo-Fast 数据集 [45] 的三个视频片段**。我们根据时间超参数 $t \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ 显示相邻两帧之间的中间帧

图 9: **TMNet 在 STVSR 上的时间一致性**。我们的 TMNet 插值了 9 帧，而 Zooming Slow-Mo [44] 在第 1 帧和第 3 帧之间插值了 1 帧。为了说明视频的时间运动，我们从左图所示的红线上提取整个帧的一维像素向量，并将一维像素向量连接成二维图像。我们观察到我们的 TMNet（图 9，右上）比 Zooming SlowMo [44]（图 9，右下）产生更一致的时间运动轨迹，Zooming SlowMo [44] 有明显的断裂变化。

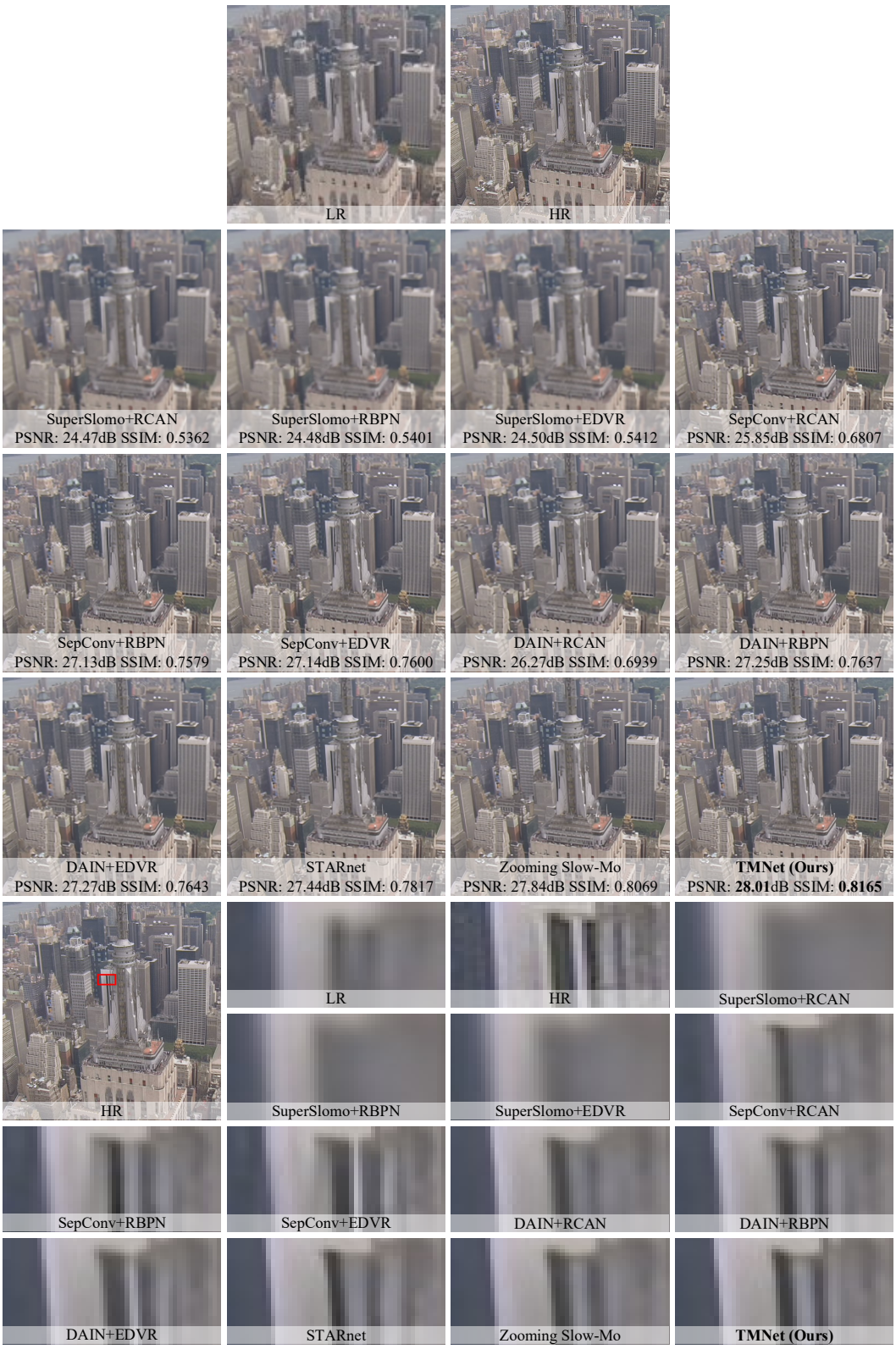

图 10: **TMNet 和其他 STVSR 方法在 Vid4** 中的 “city” 上的定量和定性结果。对于两阶段 STVSR 方法，我们使用 SuperSloMo [15]、SepConv [27] 或 DAIN [1] 用于 VFI 和 RCAN [46]、RBPN [7] 或 EDVR [40] 用于 VSR。对于单阶段 STVSR 方法，我们将我们的 TMNet 与 STARnet [8] 和 Zooming Slow-Mo [44]）进行比较。

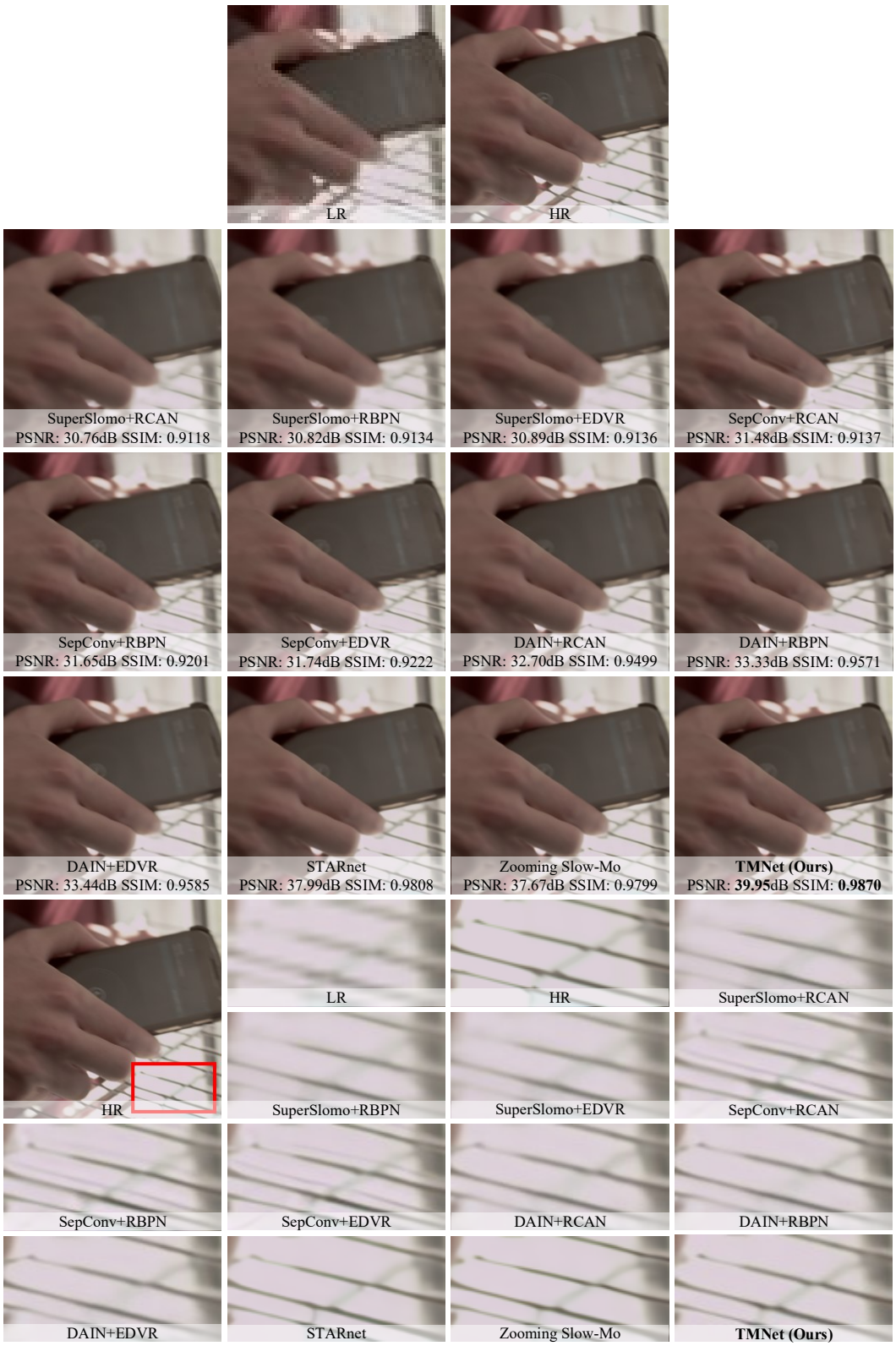

图 11: **TMNet 和其他 STVSR 方法在 Vimeo-Fast [45] 中的 0200 视频片段 “00026” 上的定量和定性结果。**对于两阶段 STVSR 方法，我们使用 SuperSloMo [15]、SepConv [27] 或 DAIN [1] 用于 VFI 和 RCAN [46]、RBPN [7] 或 EDVR [40] 用于 VSR。对于单阶段 STVSR 方法，我们将我们的 TMNet 与 STARnet [8] 和 Zooming Slow-Mo [44]）进行比较。

图 12: **TMNet 和其他 STVSR 方法在 Vimeo-Medium** [45] 中的 0723 视频片段“00085”上的定量和定性结果。对于两阶段 STVSR 方法，我们使用 SuperSloMo [15]、SepConv [27] 或 DAIN [1] 用于 VFI 和 RCAN [46]、RBPN [7] 或 EDVR [40] 用于 VSR。对于单阶段 STVSR 方法，我们将我们的 TMNet 与 STARnet [8] 和 Zooming Slow-Mo [44]）进行比较。

图 13: **TMNet 和其他 STVSR 方法在 Vimeo-Slow** [45] 中的 0679 视频片段 “00084” 上的定量和定性结果。对于两阶段 STVSR 方法，我们使用 SuperSloMo [15]、SepConv [27] 或 DAIN [1] 用于 VFI 和 RCAN [46]、RBPN [7] 或 EDVR [40] 用于 VSR。对于单阶段 STVSR 方法，我们将我们的 TMNet 与 STARnet [8] 和 Zooming Slow-Mo [44]）进行比较。