Temporal Modulation Network for Controllable Space-Time Video Super-Resolution

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

Abstract

Space-time video super-resolution (STVSR) aims to increase the spatial and temporal resolutions of low-resolution and low-frame-rate videos. Recently, deformable convolution based methods have achieved promising STVSR performance, but they could only infer the intermediate frame pre-defined in the training stage. Besides, these methods undervalued the short-term motion cues among adjacent frames. In this paper, we propose a Temporal Modulation Network (TMNet) to interpolate arbitrary intermediate frame(s) with accurate high-resolution reconstruction. Specifically, we propose a Temporal Modulation Block (TMB) to modulate deformable convolution kernels for controllable feature interpolation. To well exploit the temporal information, we propose a Locally-temporal Feature Comparison (LFC) module, along with the Bi-directional Deformable ConvLSTM, to extract short-term and long-term motion cues in videos. Experiments on three benchmark datasets demonstrate that our TMNet outperforms previous STVSR methods. The code is available at https://github.com/CS-GangXu/TMNet.

1. Introduction

Nowadays, flat-panel displays using liquid-crystal display (LCD) or light-emitting diode (LED) technologies can broadcast Ultra High Definition Television (UHD TV) videos with 4K (3840 \times 2160) or 8K (7680 \times 4320) full-color pixels, at the frame rate of 120 frames per second (FPS) or 240 FPS [39]. However, currently available videos are commonly in Full High Definition (FHD) with a resolution of 2K (1920 \times 1080) at 30 FPS [45]. To broadcast FHD videos on UHD TVs, it is necessary to increase their space-time resolutions comfortably with the broadcasting standard of UHD TVs. Although it is possible to increase the spatial resolution of videos frame-by-frame via single image super-resolution methods [4,22], the perceptual quality of the enhanced videos would be deteriorated by temporal distortion [17]. To this end, the space-time video super-resolution (STVSR) methods [31, 45] are developed to simultaneously increase the spatial and temporal resolutions of low-frame-rate and low-resolution videos.

Previous model-based STVSR methods [30–32] rely heavily on precise spatial and temporal registration [38], and would produce inferior reconstruction results when the registration is inaccurate. Besides, they usually require huge computational costs on solving complex optimization problems, resulting in low inference efficiency [21, 25]. Later, deep convolutional neural networks [12, 13, 34, 44] have been widely employed in video restoration tasks such as video super-resolution (VSR) [3, 37], video frame interpretation (VFI) [1, 27, 46], and the more challenging STVSR [17, 45]. A straightforward solution for STVSR is to perform VFI and VSR successively on low-resolution and low-frame-rate videos, to increase their spatial resolutions and frame rates [45]. However, these two-stage methods ignore the inherent correlation between temporal and spatial dimensions. That is, the videos with high-resolution frames contain richer details on moving object(s) and back-
ground, while those in high-frame-rate provide finer pixel alignment between adjacent frames [8]. Therefore, these two-stage STVSR methods would suffer from the temporal inconsistency problem [45] and produce artifacts, e.g., “the attentional blink phenomenon” [36] on STVSR.

To well exploit the correlation between the temporal and spatial dimensions in videos, several one-stage STVSR methods [8, 17, 45] have been proposed to simultaneously perform VFI and VSR reconstruction on low-frame-rate and low-resolution videos. The work of STARnet [8] estimates the motion cues with an additional optical flow branch [5], and performs feature warping of two adjacent frames to interpolate the intermediate frame. But this flow-based method [8] needs to learn an extra branch for optical flow estimation, which consumes expensive costs on computation and memory. To alleviate this problem, Xiang et al. [45] employed the deformable convolution backbone [41], and directly performed STVSR on the feature space. Though with promising performance, current STVSR networks could only generate the intermediate frames pre-defined in the network architecture, and thus are restricted to highly-controlled application scenarios with fixed frame-rate videos. However, in many commercial scenarios, such as sports events, it is very common for the user to flexibly adjust the intermediate video frames for better visualization. Thus, it is necessary to develop controllable STVSR methods for smooth motion synthesizing.

To fulfill the versatile requirements of broadcasting scenarios, in this paper, we propose a Temporal Modulation Network (TMNet) to interpolate an arbitrary number of intermediate frames for STVSR, as shown in Figure 1. But current deformable convolution based methods [45] could only generate pre-defined intermediate frame(s). To tackle this problem, we propose a Temporal Modulation Block (TMB) to incorporate motion cues into the feature interpolation of intermediate frames. Specifically, we first estimate the motion between two adjacent frames under the deformable convolution framework [41], and learn controllable interpolation at an arbitrary moment defined by a temporal parameter. In addition, we also propose a Locally-temporal Feature Comparison module to fuse multi-frame features for effective spatial alignment and feature warping, and a globally-temporal feature fusion to explore the long-term variations of the whole video. This two-stage temporal feature fusion scheme accurately interpolates the intermediate frames for STVSR. Extensive experiments on three benchmarks [23,35,46] demonstrate that our TMNet is able to interpolate an arbitrary number of intermediate frames, and achieves state-of-the-art performance on STVSR.

The contribution of this work are three-fold:

- **We present a two-stage temporal feature fusion scheme for effective STVSR.** Specifically, we propose a locally-temporal feature comparison module to exploit the short-term motion cues of adjacent frames, and perform globally-temporal feature fusion by exploring the long-term variations over the whole video.
- **Experiments on three benchmarks show that our TMNet is able to perform controllable frame interpolation at arbitrary frame-rate**, and outperforms state-of-the-art STVSR methods.

2. Related Work

**Video frame interpolation (VFI)** aims to synthesize new intermediate frames between adjacent frames [2, 15, 20]. Early VFI methods mainly resort to optical flow techniques for motion estimation [2, 15, 27]. Jiang et al. [15] modeled motion interpretation for arbitrary frame-rate VFI. Niklaus et al. [26] warped the input frames with contextual information, and interpolated context-aware intermediate frames. Bao et al. employed motion estimation and compensation for VFI in [2], and obtained improved performance by further exploring the depth information [1]. Niklaus et al. [27] tackled the conflicts of mapping multiple pixels to the same location in VFI by softmax splatting. However, these optical flow based methods need huge computational costs on motion estimation. Therefore, recently researchers exploited to learn spatially-adaptive convolution kernels [28] or deformable ones for VFI [20].

**Video super-resolution (VSR)** is the task of increasing the spatial resolutions of low-resolution (LR) videos [16, 37, 41]. Existing VSR methods [16, 37, 41] mainly aggregate spatial information of multiple frames for high-resolution (HR) reconstruction, with the help of optical flow techniques [5]. Jo et al. [16] generated dynamic upsampling filters to enhance the LR frames with residual learning [12]. Wang et al. [41] proposed the Pyramid, Cascading and Deformable (PCD) module to perform frame alignment, and then fused multiple frames into a single one by spatial and temporal attention. Haris et al. [7] designed an iterative refinement framework by integrating the spatial and temporal contexts of multiple frames. Tian et al. [37] utilized the learned sampling offsets of deformable convolution kernels to align the supporting frames with the reference ones, which are both used to reconstruct the HR frames.

**Space-time video super-resolution (STVSR)** aims to increase the spatial and temporal dimensions of the low-frame-rate and low-resolution videos [8, 17, 45]. Shechtman et al. [31] tackled the STVSR problem by employing a directional space-time smoothness regularization on the HR video reconstruction problem. Mudenagudi et al. [25] for-
mulated their STVSR method under the Markov Random Field framework [6]. STARnet [8] leveraged inherent motion relationship between spatial and temporal dimensions with an extra optical flow branch [5], and perform feature warping of two adjacent frames to interpolate the intermediate frame. Xiang et al. [45] developed a unified framework to interpolate the multi-frame features via PCD alignment modules [41], the intermediate features by bidirectional deformable ConvLSTM [34], and finally performed STVSR by multi-frame feature fusion. In this work, our goal is to develop a temporally controllable network for powerful and flexible STVSR. Though built upon [45], our TMNet arrives at better performance on benchmark datasets, owing to the proposed locally temporal feature comparison module.

Modulation networks. Recently, researchers proposed to control the restoration intensity of the main network by additional modulation branches [9, 10, 40, 42]. These modulation networks are trained to trade-off the restoration quality and flexibility, which are controlled by hyper-parameters. He et al. [9] put feature modulation filters after each convolution layer to modulate the output according to user’s preference. Later, He et al. [10] expanded this design to multiple dimensions, and modulated the output according to the levels of multiple degradation types. Wang et al. [40] learned the features from tuning blocks and residual ones with different objectives, to control the trade-off between noise reduction and detail preservation. In this work, we consider the modulation on temporal dimension, instead of on restoration intensity as in these modulation networks. As far as we know, our work is among the first to implement temporal modulation in the STVSR problem. As will be shown in §4, our TMNet can explore the potential of temporal modulation for controllable STVSR.

3. Proposed Method

In this section, we first overview our Temporal Modulation Network (TMNet) for STVSR in §3.1. Then, we introduce our Temporal Modulation Block for controllable feature interpolation in §3.2. We present temporal feature fusion in §3.3, and high-resolution reconstruction in §3.4. Finally, the training details are given in §3.5.

3.1. Network Overview

As illustrated in Fig. 2, our TMNet consists of three seamless stages: controllable feature interpolation, temporal feature fusion, and high-resolution reconstruction.

Controllable feature interpolation. Given a sequence of low-frame-rate and low-resolution video $I^L$, we first extract initial features and perform Controllable Feature Interpolation (CFI, implemented by our Temporal Modulation Block) for the intermediate frame at an arbitrary moment $t \in (0, 1)$. Then, we feed the obtained feature maps $F^L$ into a two-stage temporal feature fusion scheme. For short-term motion consistency, the feature maps $F^L$ are refined to $F^L_{LFC}$ by our Locally-temporal Feature Comparison (LFC) module. To exploit long-term motion cues, the feature maps $F^L_{LFC}$ are further improved to $F^L_{GFF}$ by globally-temporal feature fusion (GFF), implemented by Bi-directional Deformable ConvLSTM (BDConvLSTM) [45]. Finally, we employ 40 residual blocks to reconstruct high-resolution feature maps $F^H$, and two Pixel-Shuffle layers to output the high-frame-rate and high-resolution video $I^H$.

![Figure 2: Overview of our Temporal Modulation Network (TMNet) for STVSR. Given the input low-frame-rate and low-resolution video $I^L$, we first extract initial features and perform Controllable Feature Interpolation (CFI, implemented by our Temporal Modulation Block) for the intermediate frame at an arbitrary moment $t \in (0, 1)$. Then, we feed the obtained feature maps $F^L$ into a two-stage temporal feature fusion scheme. For short-term motion consistency, the feature maps $F^L$ are refined to $F^L_{LFC}$ by our Locally-temporal Feature Comparison (LFC) module. To exploit long-term motion cues, the feature maps $F^L_{LFC}$ are further improved to $F^L_{GFF}$ by globally-temporal feature fusion (GFF), implemented by Bi-directional Deformable ConvLSTM (BDConvLSTM) [45]. Finally, we employ 40 residual blocks to reconstruct high-resolution feature maps $F^H$, and two Pixel-Shuffle layers to output the high-frame-rate and high-resolution video $I^H$.](image-url)
pose a Locally-temporal Feature Comparison (LFC) module, to refine every feature map in $F^L$ with the help of the feature maps of adjacent frames. After the local feature refinement, we further improve the feature maps in $F^L$ by performing globally-temporal feature fusion (GFF). This is implemented by employing a Bi-directional Deformable ConvLSTM (BDConvLSTM) network [45], to consecutively aggregate the useful information from individual feature maps along the temporal direction. Both the LFC and GFF fusion modules well exploit the intra-correlation between spatial and temporal dimensions to improve the quality of feature maps in $F^L$. In the end, we obtain the sequence of improved feature maps $F^L_{GFF}$.

**High-resolution reconstruction.** Here, we feed the sequence of feature maps $F^L_{GFF}$ into 40 residual blocks to improve their quality along the spatial dimension. Next, we increase the spatial resolution of these improved feature maps via the widely used Pixel-Shuffle layers [33], and output the final high-frame-rate and high-resolution video sequence $I^H = \{I^H_1, I^H_2, \ldots, I^H_{2n-2}, I^H_{2n-1}\}$.

### 3.2. Controllable Feature Interpolation

Given a sequence of low-frame-rate and low-resolution video frames $I^L = \{I^L_{2i-1}\}_{i=1}^n$, we first extract the corresponding initial features $F^L = \{F^L_{2i-1}\}_{i=1}^n$ via five residual blocks. Each residual block contains a sequence of “Conv-ReLU-Conv” operations with a skip connection. For any two adjacent frames $I^L_{2i-1}$ and $I^L_{2i+1}$ ($i \in \{1, \ldots, n-1\}$), our goal here is to interpolate the feature of intermediate frame at an arbitrary moment $t \in (0, 1)$. To this end, we need to estimate the motion cues from $I^L_{2i-1}$ to the intermediate frame (forward) and that from $I^L_{2i+1}$ to the intermediate frame (backward). Previous STVSR methods [41, 45] utilized the Pyramid, Cascading and Deformable (PCD) module to estimate the offset between $F^L_{2i-1}$ and $F^L_{2i+1}$ as the motion cues, to align and interpolate the features of the intermediate frame under the deformable convolutional framework [48]. However, the vanilla PCD module could only estimate the motion to a predefined moment, which is fixed in both training and inference stages.

To overcome this limitation, we propose a Temporal Modulation Block (TMB) to modulate the learned offset between $F^L_{2i-1}$ and $F^L_{2i+1}$. The modulation is controlled by a hyper-parameter $t \in (0, 1)$, indicating an arbitrary moment that we plan to interpolate a new frame. This enables our TMBNet to control the feature interpolation process upon the initial feature maps $F^L_{2i-1}$ and $F^L_{2i+1}$ of two adjacent frames $I^L_{2i-1}$ and $I^L_{2i+1}$ in the input video. The PCD module modulated by our TMB block can estimate the forward and backward motion vectors and interpolate the feature map $F^L_{2i,t}$ of a new frame at the arbitrary moment $t \in (0, 1)$.

Denote by $f_{TMB}^{0 \rightarrow 1}$ and $f_{TMB}^{1 \rightarrow 0}$ the PCD modules modulated by our TMB block, to model the forward and backward motions, respectively. Here, we perform modulated feature interpolation from the forward and backward directions:

$$F^L_{2i,0 \rightarrow t} = f_{TMB}^{0 \rightarrow 1}(F^L_{2i-1}, F^L_{2i+1}, t),$$

$$F^L_{2i,1 \rightarrow t} = f_{TMB}^{1 \rightarrow 0}(F^L_{2i-1}, F^L_{2i+1}, 1 - t),$$

where $F^L_{2i,0 \rightarrow t}$ and $F^L_{2i,1 \rightarrow t}$ are the interpolated features aligned from the feature maps $F^L_{2i-1}$ and $F^L_{2i+1}$ of the adjacent frames. Note that the two TMB-modulated PCD modules share the same network structure but have different weights. Here we only take the $f_{TMB}^{0 \rightarrow 1}$ as an example to explain how the PCD modules modulated by our TMB work.
on modeling the forward motion. The PCD module $f_{TMB}^L$ modeling the backward motion can be similarly explained.

As shown in Fig. 3, the PCD module has three levels to estimate the motion in different scales. To realize flexible modulation on the temporal dimension, we embed our TMB block into each level of the vanilla PCD module independently, to modulate the offset before the deformable convolutional network (DCN). The benefits of adding our TMB block to all three levels of the PCD module will be verified in §4. To adaptively modulate the offset by our TMB at different levels of PCD, we use three convolutional layers to map the temporal hyper-parameter $t$ onto a modulation vector $v_t$ of size $1 \times 1 \times 64$. To well exploit the motion cues for precise modulation, we feed the features in each vanilla PCD level into two convolutional layers to enlarge their receptive fields. Then, the generated feature is multiplied with the modulation vector $v_t$ along the channel dimension, to produce the TMB-modulated features. For robustness, we add the TMB-modulated features with the corresponding pre-modulated features before DCN.

Once obtaining the modulated feature maps $F_{2i,0\rightarrow t}^L$ and $F_{2i,1\rightarrow t}^L$, we interpolate the intermediate feature $F_{2i,t}^L$ via channel-wise concatenation $"[\cdot, \cdot]"$ followed by a $1 \times 1$ convolution layer $f_{1\times1}$ as:

$$F_{2i,t}^L = f_{1\times1}(F_{2i,0\rightarrow t}^L, F_{2i,1\rightarrow t}^L).$$

(2)

Now, we obtain the features of the interpolated sequence $F^L = \{F_1^L, F_{21,1}^L, F_{22,1}^L, ..., F_{2n-2,1}^L, F_{2n-1}^L\}$ for the high-frame-rate and low-resolution video. Next, we perform feature fusion along the temporal dimension.

3.3. Temporal Feature Fusion

Here, the initial features are extracted (or interpolated) from individual (or adjacent) frames. There is considerable leeway to improve their quality. But we also feed the initial features into the Pixel-Shuffle part of our TMNet.

**Locally-temporal feature comparison.** It is essential to maintain short-term temporal consistency for each current frame. For this purpose, we propose a Locally-temporal Feature Comparison (LFC) module to exploit the complementary information (e.g., motion cues) from adjacent frames. As illustrated in Fig. 4, to refine the feature map $F_{2i,t}^L$ of current frame from adjacent feature maps $F_{2i-1}^L$ and $F_{2i+1}^L$, we concatenate current frame ($F_{2i,t}^L$) and adjacent frames ($F_{2i-1}^L$, $F_{2i+1}^L$), and employ two convolutional layers to learn the offset in the deformable convolutional framework [48]. Note that we learn two offsets to describe the motion cues in the forward (from $I_{2i-1}^L$ to current frame) and the backward (from $I_{2i+1}^L$ to current frame) directions. Then, the learned offset from forward (or backward) direction is used to align the feature map $F_{2i-1}^L$ of previous (or $F_{2i+1}^L$ of next) frame with that of the current frame, via one deformable convolutional layer. After the alignment, we concatenate the aligned feature maps of two adjacent frames with that of the current frame, and perform feature comparison via four $1 \times 1$ convolutional layers and an addition operation. For the first (or last) frame, the previous (or next) adjacent frame is just itself. Now we get a refined feature sequence $F^L_{LFC}$.

**Globally-temporal feature fusion.** The feature sequence refined by our LFC module is able to maintain short-term consistency in the interpolated video. But it would fail on large or fast motions, since LFC lacks the capability of modeling the motions over the whole video. To tackle this problem, we propose to exploit the long-term information in videos by globally-temporal feature fusion. Inspired by [45], we feed the feature sequence $F^L_{LFC}$ generated by our LFC into the BDConvLSTM network, and obtain the features $F^L_{GFF}$ in long-term temporal consistency.

As will be illustrated in the experimental section, our short-term LFC module and the long-term BDConvLSTM indeed boost the performance of our TMNet on STVSR.

3.4. High-Resolution Reconstruction

Until now, the intra-correlation of temporal and spatial dimensions is well explored to obtain the high-quality feature sequence $F^L_{GFF}$ of the whole video. Then, we perform spatial refinement for the feature maps via 40 residual blocks, and get the refined feature maps $F^H$. Then we add the features $F^H$ with the corresponding initial feature maps in $F^L$, and obtain the reconstructed feature maps $F^H_{final}$. Finally, we feed the reconstructed feature maps $F^H_{final}$ into two Pixel-Shuffle layers, followed by a sequence of “Conv-LeakyReLU-Conv” operations, to output the reconstructed HR video frames $T^H = \{I_1^H, I_{21,t}^H, ..., I_{2n-2,t}^H, I_{2n-1,t}^H\}$.

3.5. Training Details

**Implementation details.** We employ the Adam optimizer [18] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize our TMNet with the Charbonnier loss function [19], as suggested in [45]. The learning rate is initialized as $4 \times 10^{-4}$, and is decayed to $1 \times 10^{-7}$ with a cosine annealing [24] for
Table 1: Comparison of PSNR, SSIM [43], speed (in fps), and parameters (in million) by different STVSR methods on Vid4 [35], Vimeo-Fast, Vimeo-Medium, Vimeo-Slow [46]. "↑" means that larger is better. The speed is evaluated on Vid4 [35]. The best, second best and third best results are highlighted in red, blue and bold, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Vid4 [35]</th>
<th>Vimeo-Fast</th>
<th>Vimeo-Medium</th>
<th>Vimeo-Slow</th>
<th>Speed</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR↑</td>
<td>SSIM↑</td>
<td>PSNR↑</td>
<td>SSIM↑</td>
<td>PSNR↑</td>
<td>SSIM↑</td>
</tr>
<tr>
<td>SuperSloMo [15] + Bicubic</td>
<td>22.84</td>
<td>0.5772</td>
<td>31.88</td>
<td>0.8793</td>
<td>29.94</td>
<td>0.8477</td>
</tr>
<tr>
<td>SuperSloMo [15] + RCAN [47]</td>
<td>23.80</td>
<td>0.6397</td>
<td>34.52</td>
<td>0.9076</td>
<td>32.50</td>
<td>0.8884</td>
</tr>
<tr>
<td>SuperSloMo [15] + RBPN [7]</td>
<td>23.76</td>
<td>0.6362</td>
<td>34.73</td>
<td>0.9108</td>
<td>32.79</td>
<td>0.8930</td>
</tr>
<tr>
<td>SuperSloMo [15] + EDVR [41]</td>
<td>24.40</td>
<td>0.6706</td>
<td>35.05</td>
<td>0.9136</td>
<td>33.85</td>
<td>0.8967</td>
</tr>
<tr>
<td>SepConv [28] + Bicubic</td>
<td>23.51</td>
<td>0.6273</td>
<td>32.27</td>
<td>0.8890</td>
<td>30.61</td>
<td>0.8633</td>
</tr>
<tr>
<td>SepConv [28] + RCAN [47]</td>
<td>24.92</td>
<td>0.7236</td>
<td>34.97</td>
<td>0.9195</td>
<td>33.59</td>
<td>0.9125</td>
</tr>
<tr>
<td>SepConv [28] + RBPN [7]</td>
<td>26.08</td>
<td>0.7751</td>
<td>35.07</td>
<td>0.9238</td>
<td>34.09</td>
<td>0.9229</td>
</tr>
<tr>
<td>SepConv [28] + EDVR [41]</td>
<td>25.93</td>
<td>0.7792</td>
<td>35.23</td>
<td>0.9252</td>
<td>34.22</td>
<td>0.9240</td>
</tr>
<tr>
<td>DAIN [1] + Bicubic</td>
<td>23.55</td>
<td>0.6268</td>
<td>32.41</td>
<td>0.8910</td>
<td>30.67</td>
<td>0.8636</td>
</tr>
<tr>
<td>DAIN [1] + RCAN [47]</td>
<td>25.03</td>
<td>0.7261</td>
<td>35.27</td>
<td>0.9242</td>
<td>33.82</td>
<td>0.9146</td>
</tr>
<tr>
<td>DAIN [1] + RBPN [7]</td>
<td>25.96</td>
<td>0.7784</td>
<td>35.55</td>
<td>0.9300</td>
<td>34.45</td>
<td>0.9262</td>
</tr>
<tr>
<td>DAIN [1] + EDVR [41]</td>
<td>26.12</td>
<td>0.7836</td>
<td>35.81</td>
<td>0.9323</td>
<td>34.66</td>
<td>0.9281</td>
</tr>
<tr>
<td>STARnet [8]</td>
<td>26.06</td>
<td>0.3046</td>
<td>36.19</td>
<td>0.9368</td>
<td>34.86</td>
<td>0.9356</td>
</tr>
<tr>
<td>Zooming Slow-Mo [45]</td>
<td>26.31</td>
<td>0.7976</td>
<td>36.81</td>
<td>0.9415</td>
<td>35.41</td>
<td>0.9361</td>
</tr>
<tr>
<td>TMNet (Ours)</td>
<td>26.43</td>
<td>0.8016</td>
<td>37.04</td>
<td>0.9435</td>
<td>35.60</td>
<td>0.9380</td>
</tr>
</tbody>
</table>

**Network training.** When directly trained with the proposed TMB block, our TMNet suffers from clear performance drops on STVSR, as shown in our experiments. One possible reason is that our TMNet can not accurately estimate the motion cues to interpolate an intermediate frame at the arbitrary moment $t \in (0, 1)$ since our TMB does not know the moment before the training modulated feature. To resolve this problem, we propose to train our TMNet by a two-step strategy: Step 1, we train our main TMNet without the proposed TMB block; Step 2, we only train our TMB block while fixing the trained main network.

In Step 1, we train our TMNet on the Vimeo-90K dataset [46], which will be introduce in §4.1. This dataset consists of 7-frame video clips. For each clip, the 1-st, 3-rd, 5-th, and 7-th LR frames are input into our TMNet as low-frame-rate and low-resolution video. We set $t = 0.5$ to get rid of the TMB block from our TMNet, and learn to generate the 7-frame high-resolution and high-frame-rate video. This enables our TMNet to fairly compare with previous STVSR methods [8, 35, 41, 45]. For supervision, we calculate the loss function over the corresponding 7-frame HR video clip in the Vimeo-90K dataset [46].

In Step 2, we fix the learned weights of our main network, and only train our TMB block for temporal modulation. The training is performed on the Adobe240fps dataset [35], which is in high-frame-rate and suitable for training our TMB block. We also split it into groups of 7-frame video clips. For each clip, the 1-st and 7-th HR frames are downsampled as the inputs of our TMNet. We set the temporal hyper-parameter $t \in \{1, 2, 3, 4, 5\}$ to interpolate 5 intermediate frames. This step costs 35.26 minutes.

4. Experiments

4.1. Experimental Setup

**Dataset.** We use Vimeo-90K septuplet dataset [46] as the training set. It contains 91,701 video sequences, extracted from 39K video clips selected from Vimeo-90K. Each sequence contains 7 continuous frames of resolution $448 \times 256$. The Vid4 [23] and Vimeo-90K test set are used as evaluation datasets. As suggested in [45], we split the Vimeo-90K septuplet test set into three subsets of Fast motion, Medium motion, and Slow motion, which include 1225, 4977, and 1613 video clips, respectively. We also remove 5 video clips from the original Medium motion set and 3 clips from the Slow motion set, which contain only all-black backgrounds.

To make our TMNet feasible for controllable feature interpolation, we train our TMB block individually on the Adobe240fps dataset [35]. It has 133 videos (in 720P) taken with hand-held cameras, and is randomly split into the train, val, and test subsets with 100, 16, and 17 videos, respectively. For each video, we split it into groups of 7-frame video clips. We feed the 1-st and 7-th frames in each clip into our TMNet to generate 5 intermediate frames. We downsample the HR frames to create the LR frames by bicubic interpolation, with a factor of 4.
Evaluation metric. We employ the widely used Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [43] to evaluate different methods on the STVSR task. The PSNR and SSIM metrics are calculated on the Y channel of the YCbCr color space, as favored by previous VSR [7,41] and STVSR [45] methods.

4.2. Comparison to State-of-the-arts

Comparison methods. We compare our TMNet with state-of-the-art two-stage and one-stage STVSR methods. For the two-stage STVSR methods, we perform video frame interpolation (VFI) by SuperSloMo [15], DAIN [1] or SepConv [28], and perform video super-resolution (VSR) by Bicubic Interpolation (BI), RCAN [47], RBPN [7] or EDVR [41]. For one-stage STVSR methods, we compare our TMNet with the recently developed Zooming SlowMo [45] and STARnet [8]. To fairly compare with these competitors, we set $t = 0.5$ in our TMNet to generate the frame at the middle moment of any two adjacent frames. That is, the 1-st, 3-rd, 5-th, and 7-th LR frames of each clip in Vimeo-90K are fed into our TMNet to reconstruct the 7 HR frames. All these methods are trained on the Vimeo-90K septuplet dataset [46], evaluated on the Vimeo-90K test set [46] and the Vid4 [35] dataset.

Objective results. We list the quantitative comparison results in Table 1. As suggested in [45], we omit the baseline models with Bicubic Interpolation when comparing the speed. One can see that our TMNet outperforms the Zooming SlowMo [45] by 0.12dB, 0.23dB, 0.19dB, and 0.15dB on the Vid4, Vimeo-Fast, Vimeo-Medium, and Vimeo-Slow datasets in terms of PSNR. On SSIM [43], our TMNet achieves better results than the competitors in most cases, but is only slightly inferior to STARNet [8] on Vid4 [23] and Vimeo-Slow. However, our TMNet needs only one-ninth of the parameters in STARNet. On speed, one-stage methods [8,45] run much faster than two-stage ones [1,7,15,28,41]. Our TMNet runs at 14.69fps, and is only slower than Zooming Slow-Mo [45]. All these results validate the effectiveness of our TMNet on STVSR.

Visualization. In the 1-st row of Figure 5, we present the 5 intermediate frames (Frame 1550 to Frame 1554) interpolated by our TMNet on the sequence “0056” from the Adobe240fps test set [35], given the Frame 1549 and Frame 1555 as inputs. It can be seen that our TMNet is able to perform flexible frame interpolation for STVSR. In the 2-nd row of Figure 5, we show the reconstructed frames by different STVSR methods from Vimeo-Fast and Vimeo-Slow datasets [46] generated by the competing methods respectively. We observe that our TMNet, with the proposed LFC module, can restore more clearly the structures and textures than the competitors. For example, on the Clip “0001” in the sequence “0070” of Vimeo-Slow datasets, our TMNet reconstructs clearly the texture pattern on the bag. In summary, our TMNet demonstrates flexible and powerful STVSR ability quantitatively and qualitatively. More visual comparison on the Vid4 [23], Vimeo-90K test set [35], and Adobe240fps [46] datasets are provided in the Supplementary File, because of page limitation.

4.3. Ablation Study

Here, we conduct detailed examinations of our TMNet on STVSR. Specifically, we assess 1) the importance of our Temporal Modulation Block (TMB) for controllable feature interpolation; 2) different strategies that our TMB block modulates the PCD module; 3) how to design our TMB block; 4) how our Locally-temporal Feature Comparison (LFC) module contribute to the temporal feature fusion in our TMNet; 5) the combination of high-quality feature maps $F^H$ and initial feature maps $F^L$ for STVSR.

1. Does our TMB block contribute to controllable feature interpolation? To answer this question, we compare our TMNet with previous STVSR methods [8,45] on generating intermediate frames from two adjacent frames. Due to limited space, we provide the comparison of visual results on Adobe240fps test set [35] in the Supplementary File. We observe that our TMNet with TMB block indeed
exhibits temporally controllable STVSR performance.

2. How different strategies that our TMB block modulate the PCD module influence our TMNet on STVSR? The PCD module [41] has a three-level pyramid structure: the 1-st level L1; the 2-nd level L2 is downsampled from the features in L1 by convolution filters at a stride of 2; similarly, the 3-rd level L3 is downsampled from L2 by a stride of 2. In our TMNet, the proposed TMB modulates all the three levels of the PCD module. But our TMB can also modulate only one level (L1, L2, or L3) of PCD, resulting in three variants of our TMNet called TMB-L1, TMB-L2, and TMB-L3. These variants are trained on the Adobe240fps train set [35] and evaluate them on test set. As shown in Table 2, the three variants perform in descending order, indicating that the 1-st level of PCD is more important for temporal modulation. By modulating all three levels of PCD, our TMNet outperforms the three variants on STVSR, by better exploiting the motion cues of videos.

Table 2: PSNR results on Adobe240fps test set by different strategies that our TMB modules the PCD.

<table>
<thead>
<tr>
<th>Variant</th>
<th>TMB-L1</th>
<th>TMB-L2</th>
<th>TMB-L3</th>
<th>TMNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (dB)</td>
<td>26.92</td>
<td>26.82</td>
<td>26.60</td>
<td>26.95</td>
</tr>
</tbody>
</table>

3. How to design our TMB block? The goal of our TMB is to transform the hyper-parameter \( t \) into a modulation vector \( v_t \) comfortable with the PCD module. A trivial design of our TMB is a linear convolutional layer. We call it TMB-Linear. We train our TMB and the TMB-Linear on Adobe240fps train set [35] and evaluate them on test set. The PSNR results are listed in Table 3, in which the TMB-Linear is 0.02dB lower than our TMB with three nonlinear convolutional layers. This shows that nonlinear transformation is only a little better than the linear one.

Table 3: PSNR results on Adobe240fps test set by our TMB with linear or nonlinear design.

<table>
<thead>
<tr>
<th>Variant</th>
<th>TMB-Linear</th>
<th>TMB</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (dB)</td>
<td>26.93</td>
<td>26.95</td>
</tr>
</tbody>
</table>

4. How important is the proposed LFC module to our TMNet? Our TMNet performs a two-stage temporal feature fusion: first local fusion via LFC and then global fusion via GFF. Thus, our TMNet can be called “LFC→GFF”. Inverting the order, i.e., GFF→LFC, makes our TMNet collapse during the training. The main reason is that performing GFF before LFC brings noisy long-term information, confusing the learning of deformable convolution in LFC. Thus, we do not evaluate this variant. To study how our LFC contributes to the two-stage fusion in our TMNet, we remove LPC from our TMNet and call this variant “GFF”. Besides, the features from LFC and GFF can be concatenated and fused by a convolutional layer, resulting in a variant “LPC+GFF”. We train our TMNet and its variants on the Vimeo-90K septuplet dataset and evaluate them on the Vid4 [23], Vimeo-Fast, Vimeo-Medium, and Vimeo-Slow datasets. The PSNR results are listed in Table 4. One can see that our TMNet (LFC→GFF) achieves the best results on all cases, and outperforms GFF by 0.07dB on Vid4, 0.17dB on Vimeo-Fast, 0.15dB on Vimeo-Medium, and 0.11dB on Vimeo-Slow. This indicates that our LFC module is essential to the success of our TMNet on STVSR, by exploiting short-term motion cues among adjacent frames.

Table 4: Comparison of PSNR (dB) results by different variants of our TMNet on STVSR datasets.

<table>
<thead>
<tr>
<th>Variant</th>
<th>GFF</th>
<th>LFC+GFF</th>
<th>LFC→GFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vid4 [23]</td>
<td>26.36</td>
<td>26.35</td>
<td>26.43</td>
</tr>
<tr>
<td>Vimeo-Fast</td>
<td>36.87</td>
<td>36.90</td>
<td>37.04</td>
</tr>
<tr>
<td>Vimeo-Medium</td>
<td>35.45</td>
<td>35.47</td>
<td>35.60</td>
</tr>
<tr>
<td>Vimeo-Slow</td>
<td>33.40</td>
<td>33.43</td>
<td>33.51</td>
</tr>
</tbody>
</table>

5. The benefits of combining the high-quality feature maps \( F^H \) and the initial feature maps \( F^L \) for STVSR. In our TMNet, we combine the high-quality features \( F^H \) with the initial features \( F^L \) before the Pixel-Shuffle layers for final STVSR. Since the initial features \( F^L \) largely influence our LFC module, we remove them both from our TMNet and obtain a variant “Baseline”. Then we add \( F^L \) to the “Baseline”, and obtain a variant model “+\( F^L \)”. We train our TMNet and the two variants on the Vimeo-90K septuplet dataset [46], and evaluate them on Vimeo-90K test and Vid4 [23] datasets. As shown in Table 5, the variant “+\( F^L \)” clearly exceeds the “Baseline”. This validates that combining high-quality features \( F^H \) with initial ones \( F^L \) is helpful to our TMNet on STVSR.

Table 5: Comparison of PSNR (dB) by our TMNet and its variants on different STVSR datasets.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Baseline</th>
<th>+( F^L )</th>
<th>TMNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vid4 [23]</td>
<td>26.33</td>
<td>26.36</td>
<td>26.43</td>
</tr>
<tr>
<td>Vimeo-Fast</td>
<td>36.75</td>
<td>36.90</td>
<td>37.04</td>
</tr>
<tr>
<td>Vimeo-Medium</td>
<td>35.35</td>
<td>35.45</td>
<td>35.60</td>
</tr>
<tr>
<td>Vimeo-Slow</td>
<td>33.28</td>
<td>33.40</td>
<td>33.51</td>
</tr>
</tbody>
</table>

5. Conclusion

In this work, we proposed a Temporal Modulation Network (TMNet) to flexibly interpolate intermediate frames for space-time video super-resolution (STVSR). Specifically, we introduced a Temporal Modulation Block to modulate the learning of the deformable convolution framework for controllable feature interpolation. To well exploit motion cues, we performed short-term and long-term temporal feature fusion consisting of our proposed Locally-temporal Feature Comparison (LFC) module and a Bi-directional Deformable ConvLSTM, respectively. Experiments on three benchmarks demonstrated the flexibility of our TMNet on interpolating intermediate frames, quantitative and qualitative advantages of our TMNet over previous methods, and effectiveness of our LFC module, for STVSR.
References


[29] 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,


