Supplemental File to "Temporal Modulation Network for Controllable Space-Time Video Super-Resolution"

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

1. Content

In this supplemental file, we provide more details of our Temporal Modulation Network (TMNet) for Space-Time Video Super-Resolution (STVSR). Specifically, we provide

- the detailed network structure of our TMNet in §2;
- more details of our two-step training scheme in §3;
- flexibility of our TMNet on interpolating arbitrary number of intermediate frames in §4;
- more visual comparisons of our TMNet with previous STVSR methods in §5;
- how the one-stage training (instead of two-stage) influences our TMNet with TMB on STVSR in §6.

2. Detailed Network Structure of Our TMNet

Here, we illustrate the detailed network architecture of our proposed TMNet in Figure 1.

We first extract the corresponding initial features $\mathcal{F}^L = \{\boldsymbol{F}^L_{2i-1}\}^n_{i=1}$ via five residual blocks. Each residual block contains a sequence of "Conv-ReLU-Conv" operations with a skip connection. The Controllable Feature Interpolation (CFI) is performed by the Pyramid, Cascading and Deformable (PCD) module [9] modulated by our proposed Temporal Modulation Block (TMB), which is illustrated in Figure 2 of our main paper. The detailed structure of our TMB block is shown in 2 (right). The proposed Locally-temporal Feature Comparison (LFC) module is presented in Figure 2 (left). The BDConvLSTM part is directly implemented by employing the Bi-directional Deformable ConvLSTM network in [11]. The Upsampling part contains operations of two "Convolutions (Conv), Pixel-Shuffle, and LeakyReLU", and one "Conv-LeakyReLU-Conv".

3. More Details of Two-step Training Scheme

Here, we provide more details of the two-step training strategy for our TMNet.

In `Step 1`, we use the Vimeo-90K `septuplet` dataset [12] as the training set, and the Vid4 [6], Vimeo-`Fast`, Vimeo-`Medium`, and Vimeo-`Slow` sets as the evaluation sets. The Vimeo-90K `septuplet`, Vimeo-`Fast`, Vimeo-`Medium`, and Vimeo-`Slow` datasets [12] consist of 7-frame video sequences, and the Vid4 [6] dataset contains 4 video clips, which contains 41, 34, 49 and 47 frames, respectively. All the frames in the Vid4 dataset [6] are split into sequences containing 7 continuous frames. We downsample all the original HR frames to obtain the low-resolution (LR) input frames via Bicubic interpolation, by a factor of 4. When we train our TMNet, we initialize the parameters of our TMNet by Kaiming initialization [4] without pre-trained weights. We set $t = 0.5$ to get rid of the TMB block and take the 1-st, 3-rd, 5-th, and 7-th LR frames of every sequence as a low-frame-rate and low-resolution input video to train our TMNet. Thus, with the supervision of the corresponding 7-frame HR video sequences in the Vimeo-90K `septuplet` dataset [12], our TMNet can learn to generate the 7-frame high-resolution and high-frame-rate video sequence. It costs 8.71 days (209.04 hours) to train our TMNet for 600,000 iterations.

In `Step 2`, we fix the weights of our main network learned in `Step 1` and only train our TMB block for temporal modulation. Here, we train our TMNet on the Adobe240fps dataset [8], which has 133 videos in 720P with high-frame-rate (240fps). At first, We randomly split the Adobe240fps dataset [8] into the `train`, `val`, and `test` subsets with 100, 16, and 17 videos, respectively. Then we split the frames from Adobe240fps `train`, `valid`, and `test` sets into sequences of 7 continuous frames. We first downsample the original HR frames with the resolution of 1280×720 by a factor of 2 and take them as the ground truths (GTs). Then we downsample the GTs

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

to create the corresponding LR input frames by a factor of 4. All the downsample operations are performed via Bicubic interpolation. The 1-st and 7-th LR frames of each video sequence are input to our TMNet. We set the temporal hyper-parameter $t{\in}\{\frac{1}{6},\frac{2}{6},\frac{3}{6},\frac{4}{6},\frac{5}{6}\}$ to interpolate 5 intermediate frames. Supervised by the corresponding 7-frame HR video sequences in Adobe240fps `test` set as GTs, our TMNet is able to flexibly interpolate intermediate frames according to the temporal hyper-parameter. It takes 35.26 minutes to train our TMB block for 1,500 iterations on the Adobe240fps dataset [8].

4. Flexible STVSR with Arbitrary Number of Intermediate Frames

To show the flexibility of our TMNet for interpolating arbitrary number of intermediate frames on STVSR, we provide the results generated by our TMNet between the input two frames using multiple temporal hyper-parameter t. As the motions in Adobe240fps [8] dataset are extremely slow, we validate the flexibility of our TMNet on the Vimeo-90K dataset [12]. To this end, we set the temporal hyper-parameter $t{\in}\{0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9\}$ to interpolate 9 intermediate frames between any two adjacent frames, though our TMNet is trained to interpolate 5 intermediate frames between Frame 1 and Frame 7. The results are shown in Figure 3. One can see that the interpolated frames vary continuously with the change of t from 0.1 to 0.9. This demonstrates that our TMNet is feasible to generate a number of intermediate frames, which is different from the training stage. That is, our TMNet is very flexible on interpolating arbitrary number of intermediate frames, according to the temporal hyper-parameter $t\in(0,1)$.

In Figure 4, we visualize the temporal consistency of our TMNet and Zooming SlowMo [11], on the Clip 0277 of "00006" from the Vimeo-`Fast` set [12]. Our TMNet interpolates 9 frames, while Zooming Slow-Mo [11] interpolates 1 frame, between Frames 1 and 3. To illustrate the temporal motion of the videos, we extract a 1D pixel vector over the whole frames from the red line shown in the left figure, and concatenate the 1D pixel vector into a 2D image. We observe that our TMNet (Figure 4, upper right) produces more consistent temporal motion trajectory than Zooming SlowMo [11] (Figure 4, lower right), which suffers from clear breaking variations. This demonstrates the superiority of our TMNet on flexible frame interpolation for STVSR.

5. More Visual Comparisons on STVSR

On the Vid4 [6] and Vimeo-90K [8] datasets, we compare our TMNet with previous one-stage and two-stage STVSR methods. For one-stage STVSR methods, we compare our TMNet with Zooming SlowMo [11] and STARnet [3]. For the two-stage STVSR methods, we perform video frame interpolation (VFI) by SuperSloMo [5], DAIN [1], or SepConv [7], and perform video super-resolution (VSR) by RCAN [13], RBPN [2], or EDVR [9]. We set $t=0.5$ in our TMNet to generate the frame at the middle moment of any two adjacent frames, which means that 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 [12], and evaluated on the Vimeo-90K `test` set [12] and the Vid4 [8] dataset. The visualization results of the comparison result are shown in Figures 5-8.

6. Training our TMNet in One-step

Although trained by a two-step scheme, our TMNet can be directly trained with the proposed TMB block, resulting in a one-step training scheme. That is, in this one-step scheme, all the parameters of our main TMNet and the TMB block are optimized simultaneously without pre-training. In our two-step scheme, the two sets of parameters in our main TMNet and the TMB block are optimized separately (first the main TMNet, and then the TMB block). Here, we compare the performance of our TMNet trained with our two-step and the one-step schemes, resulting in two variants called TMNet-`two` (the original TMNet) and TMNet-`one`, respectively. Both variants are trained on the Adobe240fps `train` set [8] and evaluated on the Adobe240fps `test` set [8]. As shown in Table 1, comparing with our TMNet-`two`, the variant TMNet-`one` suffers from a performance drop of 1.84dB in terms of PSNR, on the Adobe240fps `test` set [8]. This demonstrates that our TMNet trained in a one-step scheme fail to estimate the motion cues, and interpolate the intermediate frames at an arbitrary moment $t\in(0,1)$. The main reason is that, in initial training iterations, our TMNet with TMB trained from scratch could not extract useful motion cues from videos, and thus fails to optimize the parameters of our TMB block for meaningful features at an arbitrary moment t.

Table 1: **PSNR results of our TMNet trained in two-step or one-step schemes** on Adobe240fps `test` set [8].

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

References

[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. 2, 7, 8, 9, 10

[2] 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, 9, 10

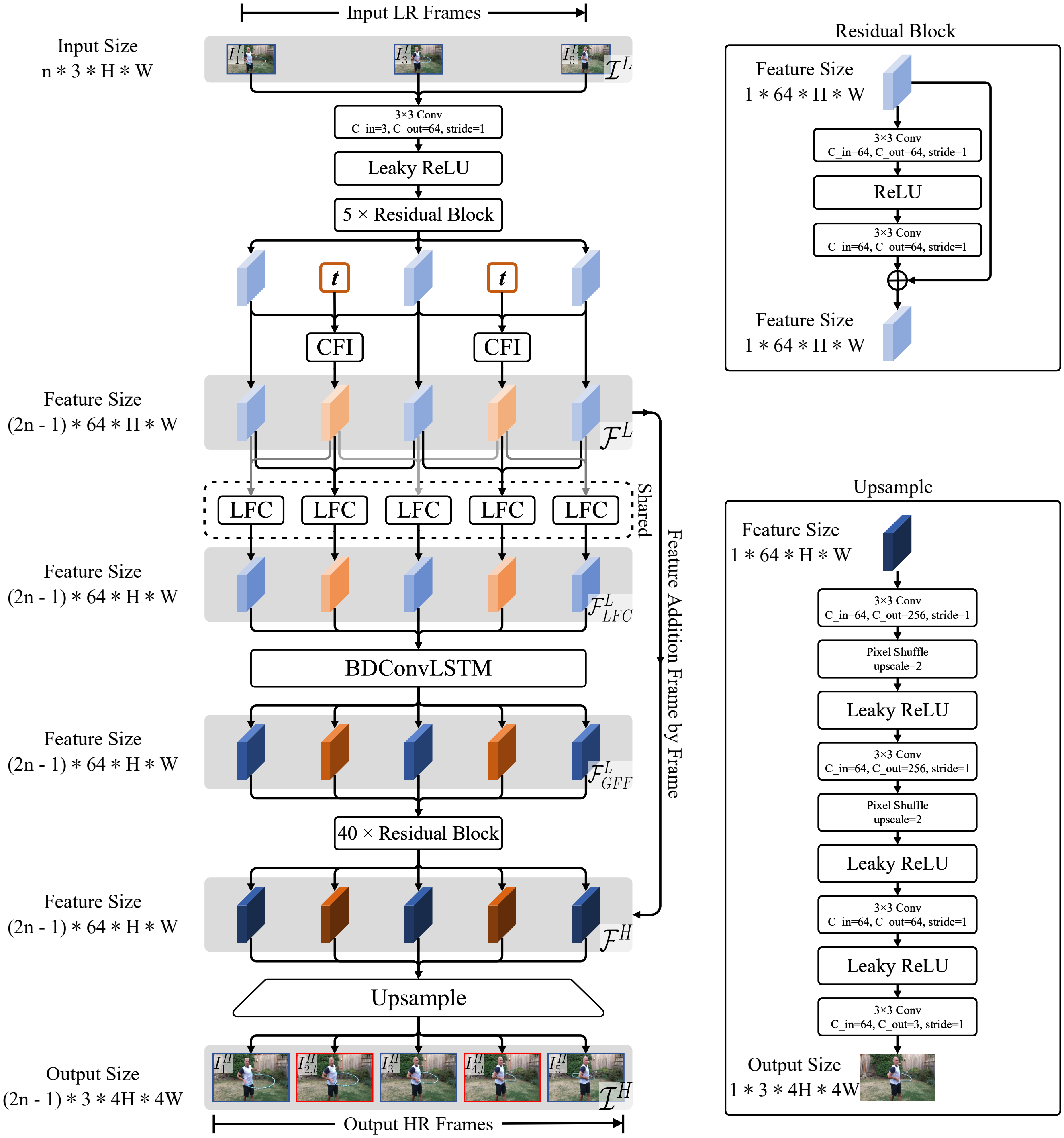

Figure 1: **Main structure of our TMNet**. The basic part of "Residual Block" and "Upsample" are illustrated on the right side. n is the number of input frames. H and W denote the height and width of the image or feature map. C_in and C_out denote the number of input and output channels, respectively.

[3] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Space-time-aware multi-resolution video enhancement. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2, 7, 8, 9, 10

[4] 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. 1

[5] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo:

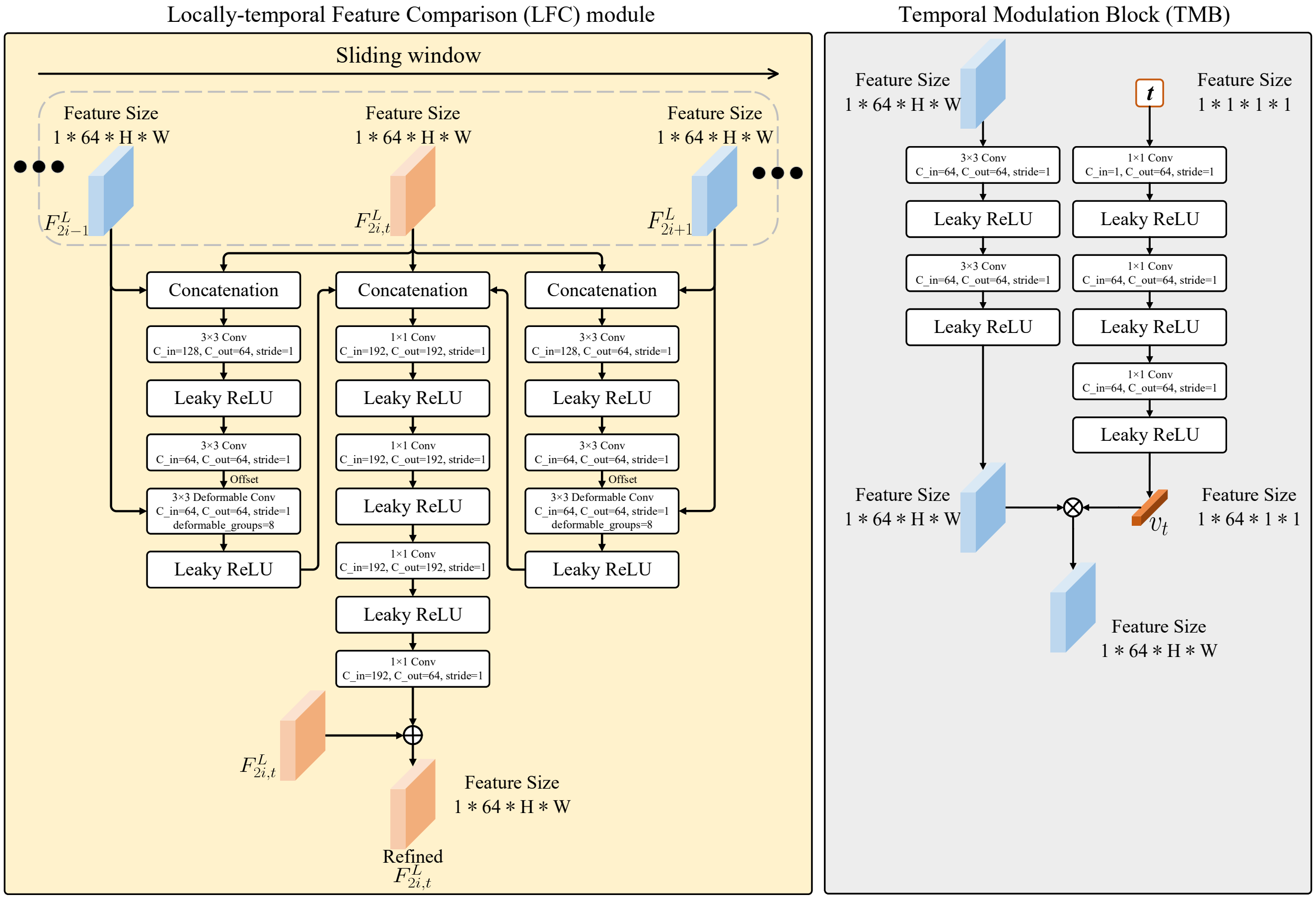

Figure 2: **Detailed structures of our Locally-temporal Feature Comparison (LFC) module (left) and Temporal Modulation Block (TMB) (right).** $2i-1$, $2i$, and $2i+1$ are the indexes of frames. H and W denote the height and width of the image or feature map. C_in and C_out denote the number of input and output channels, respectively.

High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9000–9008, 2018. 2, 7, 8, 9, 10

[6] 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. 1, 2

[7] 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, 9, 10

[8] 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. 1, 2, 7

[9] 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. 1, 2, 7, 8, 9, 10

[10] 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, 9, 10

[11] 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. Pattern Recog.*, pages 3370–3379, June 2020. 1, 2, 5, 6, 7, 8, 9, 10

[12] 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, 5, 8, 9, 10

[13] 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. 2, 7, 8, 9, 10

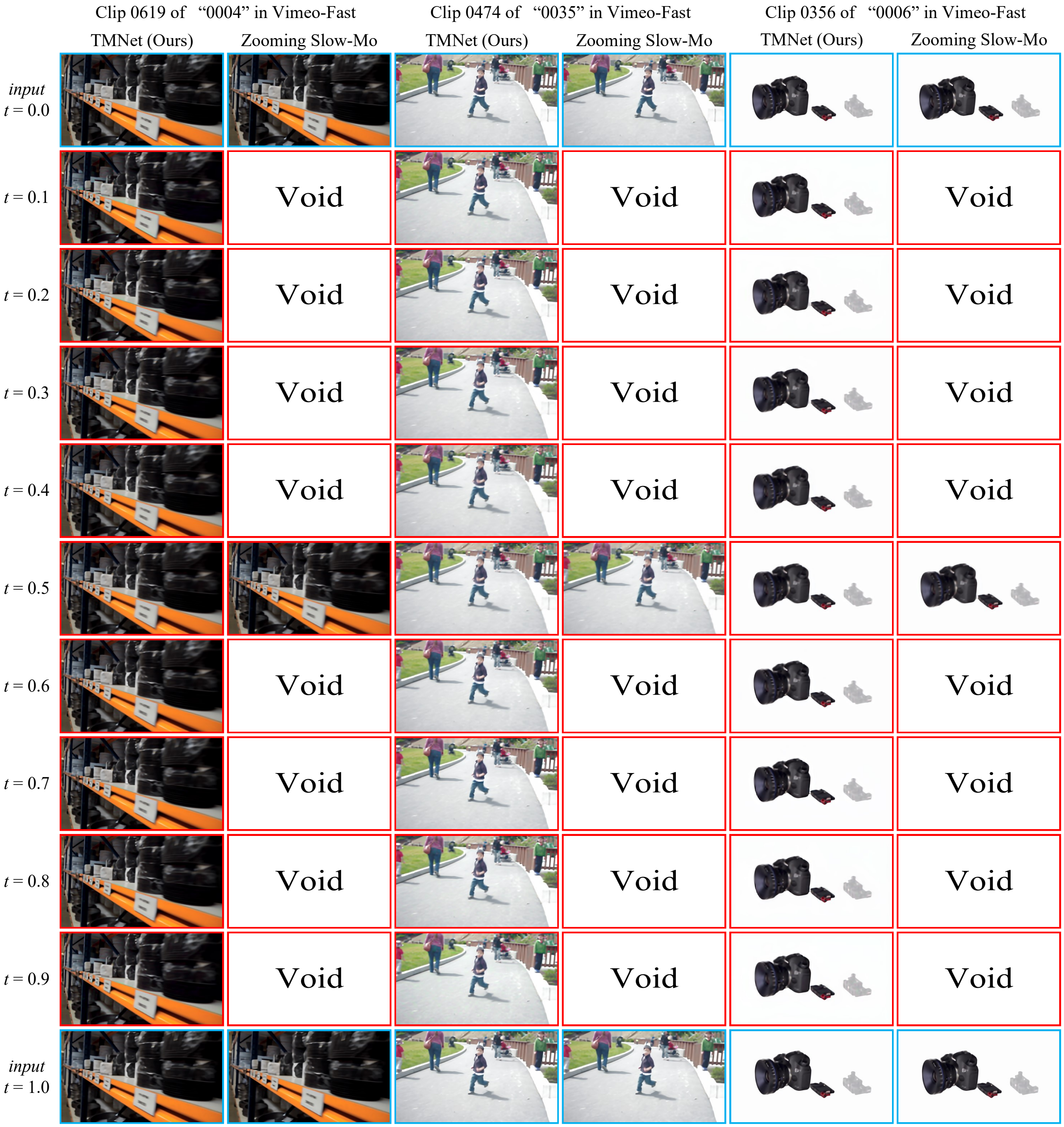

Figure 3: **Comparison of flexibility on STVSR by our TMNet (1-st, 3-rd, and 5-th columns) and Zooming Slow-Mo [11] (2-nd, 4-th, and 6-th columns)** on three video clips from the Vimeo-Fast dataset [12]. We show the intermediate frames between the adjacent two frames according to the temporal hyper-parameter $t\in\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$

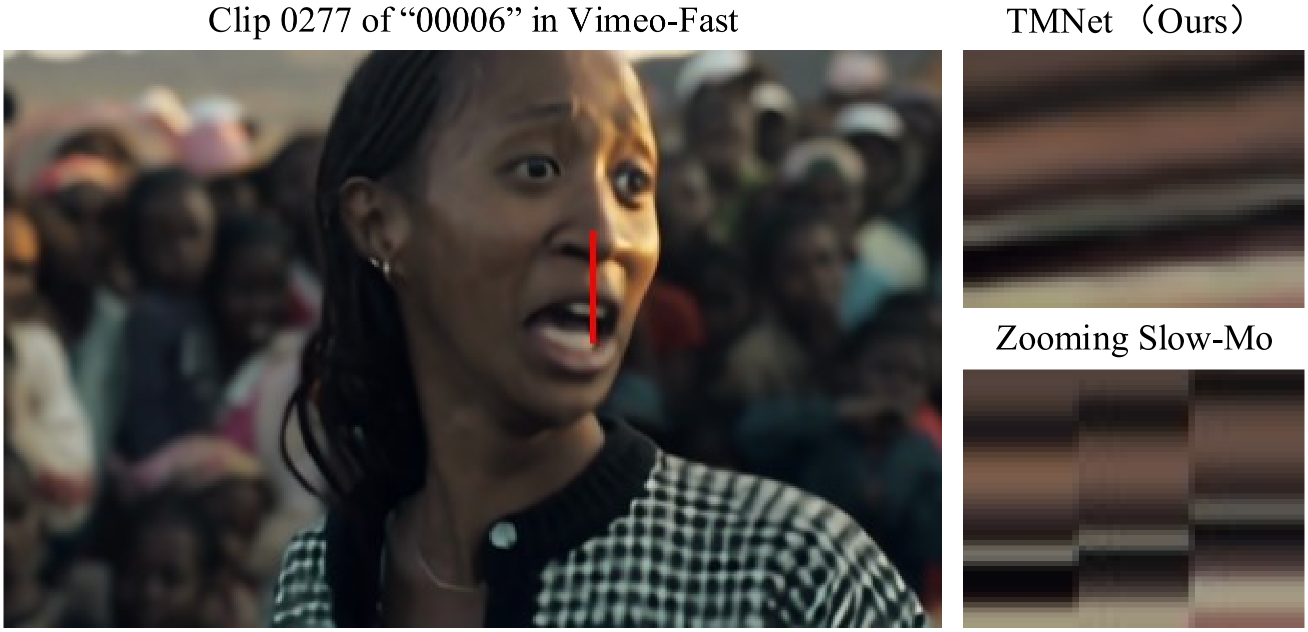

Figure 4: **Temporal consistency of our TMNet on STVSR**. OUr TMNet interpolates 9 frames, while Zooming Slow-Mo [11] interpolates 1 frame between Frames 1 and 3. We extract a 1D pixel vector over the whole frames from the red line shown in the left figure, and concatenate the 1D pixel vector into a 2D image, which is horizontally scaled to better visualize the temporal consistency of the videos. One can see that our TMNet (upper right) achieves clearly consistent temporal interpolation, while Zooming Slow-Mo [11] (lower right) suffers from clear breaking variations.

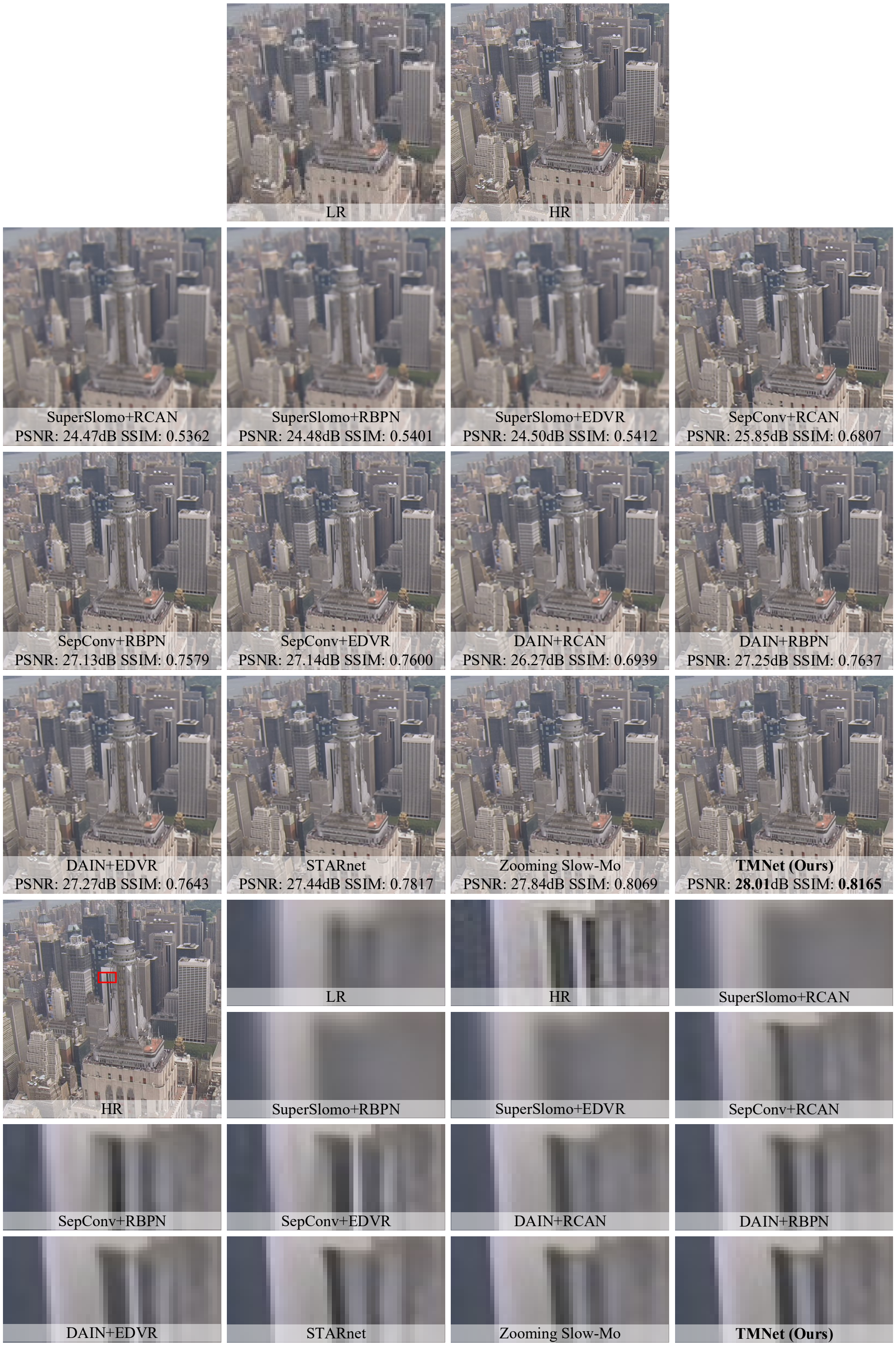
LR
HR
SuperSlomo+RCAN
PSNR: 24.47dB SSIM: 0.5362
SuperSlomo+RBPN
PSNR: 24.48dB SSIM: 0.5401
SuperSlomo+EDVR
PSNR: 24.50dB SSIM: 0.5412
SepConv+RCAN
PSNR: 25.85dB SSIM: 0.6807
SepConv+RBPN
PSNR: 27.13dB SSIM: 0.7579
SepConv+EDVR
PSNR: 27.14dB SSIM: 0.7600
DAIN+RCAN
PSNR: 26.27dB SSIM: 0.6939
DAIN+RBPN
PSNR: 27.25dB SSIM: 0.7637
DAIN+EDVR
PSNR: 27.27dB SSIM: 0.7643
STARnet
PSNR: 27.44dB SSIM: 0.7817
Zooming Slow-Mo
PSNR: 27.84dB SSIM: 0.8069
TMNet (Ours)
PSNR: 28.01dB SSIM: 0.8165
HR
LR
HR
SuperSlomo+RCAN
SuperSlomo+RBPN
SuperSlomo+EDVR
SepConv+RCAN
SepConv+RBPN
SepConv+EDVR
DAIN+RCAN
DAIN+RBPN
DAIN+EDVR
STARnet
Zooming Slow-Mo
TMNet (Ours)

Figure 5: **Quantitative and qualitative results of our TMNet and other STVSR methods on Clip "city" in the Vid4 dataset [8]**. For two-stage STVSR methods, we employ SuperSloMo [5], SepConv [7] or DAIN [1] for VFI and RCAN [13], RBPN [2] or EDVR [9] for VSR. For one-stage STVSR methods, we compare our TMNet with STARnet [3] and Zooming Slow-Mo [11]). The best results on PSNR (dB) and SSIM [10] are highlighted in **bold**.

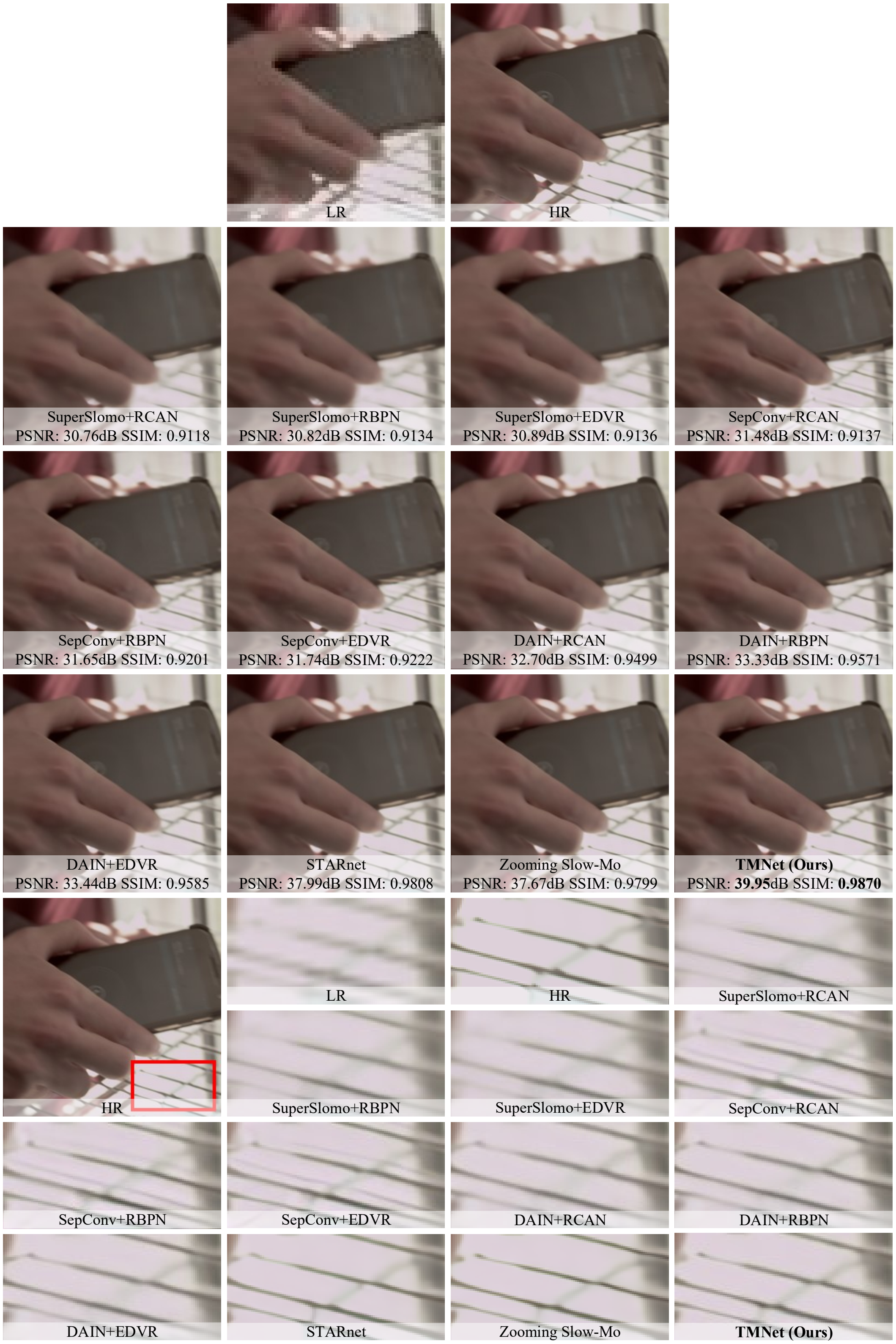
LR
HR
SuperSlomo+RCAN
PSNR: 30.76dB SSIM: 0.9118
SuperSlomo+RBPN
PSNR: 30.82dB SSIM: 0.9134
SuperSlomo+EDVR
PSNR: 30.89dB SSIM: 0.9136
SepConv+RCAN
PSNR: 31.48dB SSIM: 0.9137
SepConv+RBPN
PSNR: 31.65dB SSIM: 0.9201
SepConv+EDVR
PSNR: 31.74dB SSIM: 0.9222
DAIN+RCAN
PSNR: 32.70dB SSIM: 0.9499
DAIN+RBPN
PSNR: 33.33dB SSIM: 0.9571
DAIN+EDVR
PSNR: 33.44dB SSIM: 0.9585
STARnet
PSNR: 37.99dB SSIM: 0.9808
Zooming Slow-Mo
PSNR: 37.67dB SSIM: 0.9799
TMNet (Ours)
PSNR: 39.95dB SSIM: 0.9870
HR
LR
HR
SuperSlomo+RCAN
SuperSlomo+RBPN
SuperSlomo+EDVR
SepConv+RCAN
SepConv+RBPN
SepConv+EDVR
DAIN+RCAN
DAIN+RBPN
DAIN+EDVR
STARnet
Zooming Slow-Mo
TMNet (Ours)

Figure 6: **Quantitative and qualitative results of our TMNet and other STVSR methods on Clip 0200 of "00026" in Vimeo-`Fast` [12].** For two-stage STVSR methods, we employ SuperSloMo [5], SepConv [7] or DAIN [1] for VFI and RCAN [13], RBPN [2] or EDVR [9] for VSR. For one-stage STVSR methods, we compare our TMNet with STARnet [3] and Zooming Slow-Mo [11]). The best results on PSNR (dB) and SSIM [10] are highlighted in **bold**.

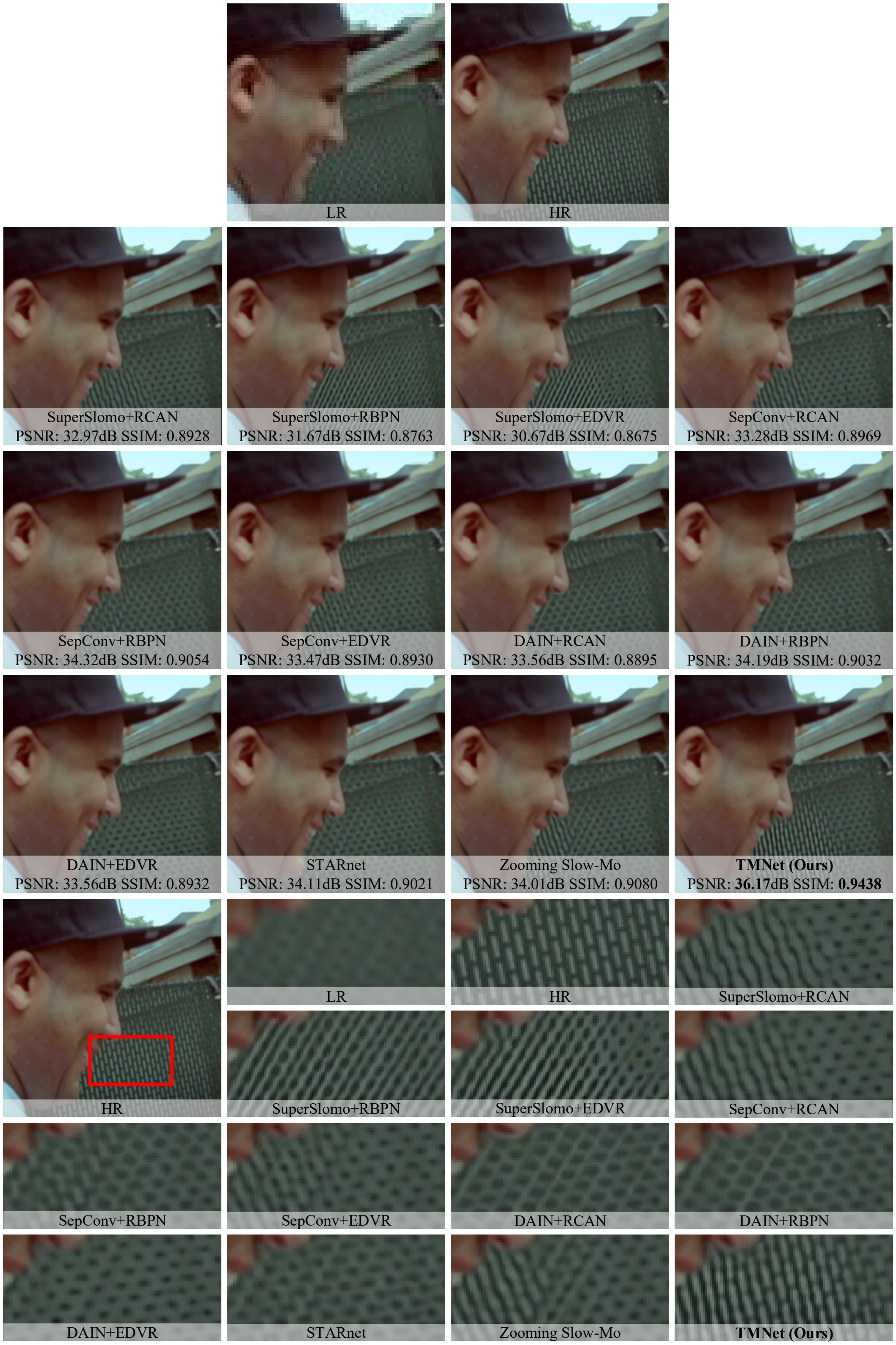
LR
HR
SuperSlomo+RCAN
PSNR: 32.97dB SSIM: 0.8928
SuperSlomo+RBPN
PSNR: 31.67dB SSIM: 0.8763
SuperSlomo+EDVR
PSNR: 30.67dB SSIM: 0.8675
SepConv+RCAN
PSNR: 33.28dB SSIM: 0.8969
SepConv+RBPN
PSNR: 34.32dB SSIM: 0.9054
SepConv+EDVR
PSNR: 33.47dB SSIM: 0.8930
DAIN+RCAN
PSNR: 33.56dB SSIM: 0.8895
DAIN+RBPN
PSNR: 34.19dB SSIM: 0.9032
DAIN+EDVR
PSNR: 33.56dB SSIM: 0.8932
STARnet
PSNR: 34.11dB SSIM: 0.9021
Zooming Slow-Mo
PSNR: 34.01dB SSIM: 0.9080
TMNet (Ours)
PSNR: 36.17dB SSIM: 0.9438
HR
LR
HR
SuperSlomo+RCAN
SuperSlomo+RBPN
SuperSlomo+EDVR
SepConv+RCAN
SepConv+RBPN
SepConv+EDVR
DAIN+RCAN
DAIN+RBPN
DAIN+EDVR
STARnet
Zooming Slow-Mo
TMNet (Ours)

Figure 7: **Quantitative and qualitative results of our TMNet and other STVSR methods on Clip 0723 of "00085" in Vimeo-Medium [12].** For two-stage STVSR methods, we employ SuperSloMo [5], SepConv [7] or DAIN [1] for VFI and RCAN [13], RBPN [2] or EDVR [9] for VSR. For one-stage STVSR methods, we compare our TMNet with STARnet [3] and Zooming Slow-Mo [11]). The best results on PSNR (dB) and SSIM [10] are highlighted in **bold**.

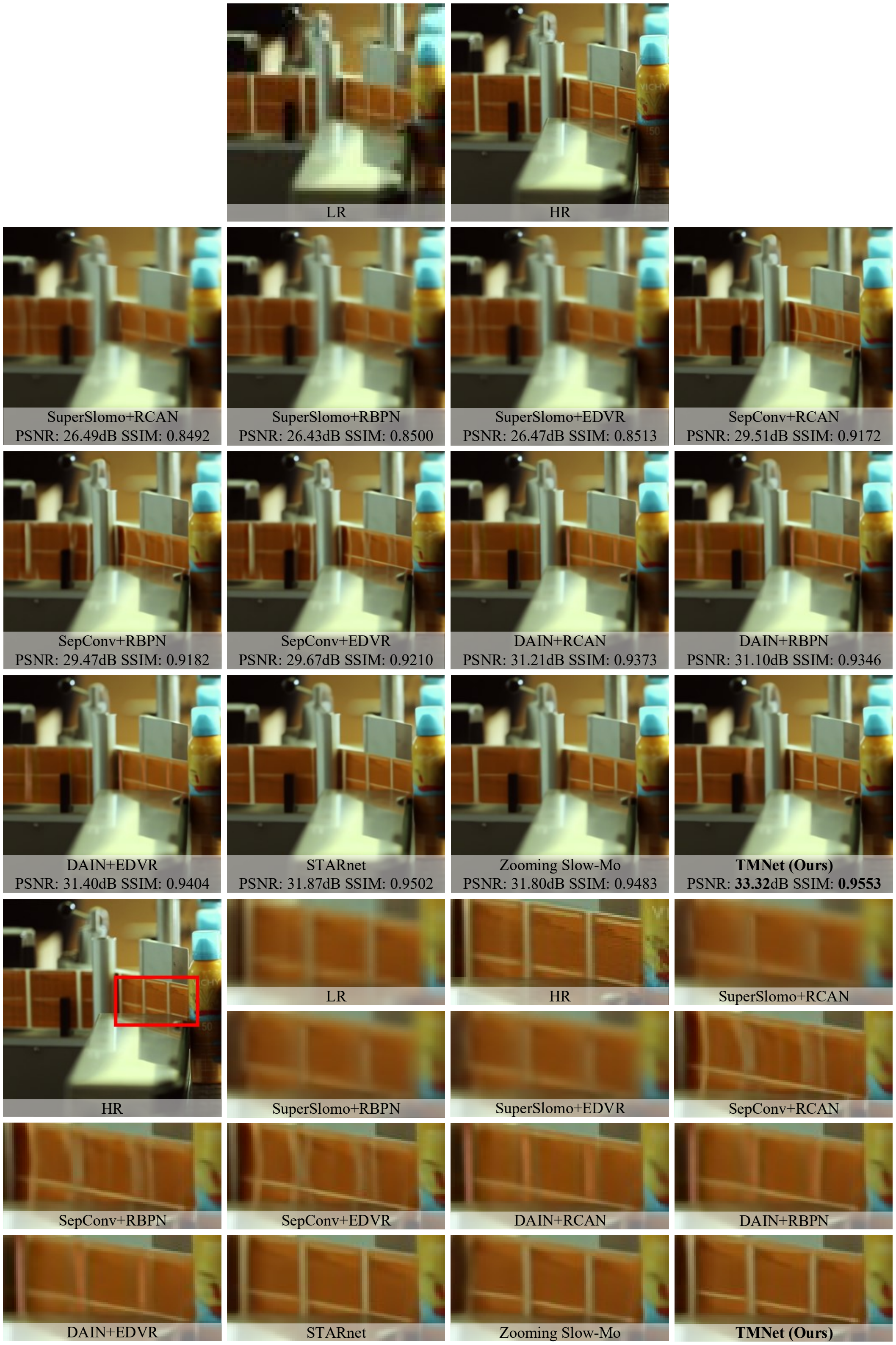
LR
HR
SuperSlomo+RCAN
PSNR: 26.49dB SSIM: 0.8492
SuperSlomo+RBPN
PSNR: 26.43dB SSIM: 0.8500
SuperSlomo+EDVR
PSNR: 26.47dB SSIM: 0.8513
SepConv+RCAN
PSNR: 29.51dB SSIM: 0.9172
SepConv+RBPN
PSNR: 29.47dB SSIM: 0.9182
SepConv+EDVR
PSNR: 29.67dB SSIM: 0.9210
DAIN+RCAN
PSNR: 31.21dB SSIM: 0.9373
DAIN+RBPN
PSNR: 31.10dB SSIM: 0.9346
DAIN+EDVR
PSNR: 31.40dB SSIM: 0.9404
STARnet
PSNR: 31.87dB SSIM: 0.9502
Zooming Slow-Mo
PSNR: 31.80dB SSIM: 0.9483
TMNet (Ours)
PSNR: 33.32dB SSIM: 0.9553
HR
LR
HR
SuperSlomo+RCAN
SuperSlomo+RBPN
SuperSlomo+EDVR
SepConv+RCAN
SepConv+RBPN
SepConv+EDVR
DAIN+RCAN
DAIN+RBPN
DAIN+EDVR
STARnet
Zooming Slow-Mo
TMNet (Ours)

Figure 8: **Quantitative and qualitative results of our TMNet and other STVSR methods on Clip 0679 of "00084" in Vimeo-`Slow` [12].** For two-stage STVSR methods, we employ SuperSloMo [5], SepConv [7] or DAIN [1] for VFI and RCAN [13], RBPN [2] or EDVR [9] for VSR. For one-stage STVSR methods, we compare our TMNet with STARnet [3] and Zooming Slow-Mo [11]). The best results on PSNR (dB) and SSIM [10] are highlighted in **bold**.