FMA-Net: Flow-Guided Dynamic Filtering and Iterative Feature Refinement with Multi-Attention for Joint Video Super-Resolution and Deblurring

Supplementary Material

In this supplementary material, we first describe the ablation studies for various components of our design on FMA-Net in Sec. A. Subsequently, in Sec. B, we introduce a lightweight version of FMA-Net and present the performance of VSRDB methods and all possible combinations of sequential cascade approaches in REDS4 [52]. Additionally, we also provide additional qualitative comparison results and a demo video. Finally, we discuss the limitations of our FMA-Net in Sec. C.

We also recommend the readers to refer to our project page at https://kaist-viclab.github.io/fmanet-site where the source codes and the pre-trained models are available for the sake of *reproducibility*.

A. Ablation Studies

A.1. Effect of flow-guided dynamic filtering (FGDF)

Fig. 8 shows the spatial quality (PSNR) and temporal consistency (tOF) performance over average motion magnitudes which are also tabulated in Table 3 of the main paper. As shown in Fig. 8, our Flow-Guided Dynamic Filtering (FGDF) significantly outperforms the conventional dynamic filtering [33, 35, 38] in terms of PSNR and tOF metrics across all average motion magnitudes. It is noted that our FGDF gets more superior as the average motion magnitudes increase, indicating the effectiveness of FGDF, which is aware of motion trajectory, over the conventional dynamic filtering based on fixed positions and surroundings.

A.2. Design choices for FMA-Net

Table 5 presents more detailed results of the ablation study from Table 4 of the main paper, additionally including the reconstruction performance of the degradation learning network Net^D. The tendency of performance changes on the selection of components for Net^D are similar to those for Net^R, demonstrating the effectiveness of our multi-flow-mask pairs (Table 5(a-b, j), loss functions (Table 5(d-e, j)), training strategy (Table 5(f, j)), and multi-attention module (Table 5(g-j)). It should be noted that the two reconstruction performances of Net^D in Table 5(h) (CO attn + SFT [68]) and Table 5(j) (CO attn + DA attn) are the same because the SFT [68] and DA attn are only utilized in Net^R. The same tendency is also observed in Table 5(g,i) because the same $NetD$ is used.

A.3. Number of input frames

Table 6 shows the performance of FMA-Net according to the different numbers of input frames T. It shows that as T increases, the performance of both Net^D and Net^R improves, indicating that the FMA-Net effectively utilizes long-term information. Considering the trade-off between computational complexity and performance, we finally adopted $T = 3$.

A.4. Iterative Feature Refinement

Fig. 9 illustrates the iterative refinement process of the warped feature $F_w^{R,i}$ in FRMA blocks of Net^R across three different scenes. In these scenes, it is evident that $F_w^{R,i}$ becomes sharper and more activated through iterative refinement, demonstrating the effectiveness of our iterative feature refinement with multi-attention (FRMA) block in improving the overall performance for VSRDB.

A.5. Multi-flow-mask pairs

Fig. 10 illustrates an example of multi-flow-mask pairs $\mathbf{f}^{R,M}$ in Net^R. In contrast to conventional sharp LR VSR methods [5, 9, 50, 62] that only utilize smooth optical flows with similar values among pixels belonging to the same object, the optical flows in Fig. 10 include not only smooth optical flows (# 2, # 7, and # 9 in Fig. 10) but also sharp optical flows (# 1, # 3-6, and # 8 in Fig. 10) with varying values among pixels belonging to the same object. This distinction arises from our multi-flow-mask pairs $\mathbf{f}$ not only *align* features as in conventional VSR methods, but also *sharpen* blurry features where the blur is pixel-wise-variant, even among pixels belonging to the same object. The smooth optical flows *align* features, while the sharp optical flows *sharpen* them. Fig. 9(c) shows the iterative refinement process of the aligned and sharpened warped feature F_w using the multi-flow-mask pairs $\mathbf{f}$ in the same scene as Fig. 10, demonstrating the effectiveness of our multi-flow-mask pairs for VSRDB.

A.6. Visualization of FGDF process

Fig. 11 illustrates the proposed flow-guided dynamic filtering (FGDF) process, where Fig. 11(a) shows the flow-guided dynamic downsampling process of Net^D, and Fig. 11(b) illustrates the flow-guided dynamic upsampling process of Net^R. In particular, in Fig. 11(a), the two degradation kernels of the neighboring frames tend to have peaky values around their own centers, because Net^D filters a sharp HR sequence Y_w aligned to the center frame index c

based on the image-mask pair $\mathbf{f}^Y$ for Y. This allows Net^D to effectively handle large motions with relatively small-sized kernels, as demonstrated in Fig. 8 and Table 3 of the main paper. Similar to Net^D, Net^R filters the aligned blurry LR sequence X_w to the center frame index c by the image flow-mask pair $\mathbf{f}^X$ for X. We normalized the restoration kernels K^R such that their kernel weights are allowed to take on positive and negative values, where the negative kernel weights can facilitate the deblurring process (dark regions of the kernels in Fig. 11(b) represent negative values), similar to [38]. We empirically found that this approach can restore the low-frequencies more effectively than simple interpolation methods such as bilinear and bicubic interpolations. Combining these restored low frequencies with the high-frequency details $\hat{Y}_r$ predicted by Net^R in a residual learning manner results in faster training convergence and better performance compared to residual learning with bicubic upsampling or without residual learning.

A.7. The Number of FRMA Blocks M

Table 7 shows the performance of FMA-Net according to the different numbers of FRMA Blocks M. It shows that as M increases, the performance of FMA-Net improves, indicating that the stacked FRMA blocks can effectively update features. Besides Table 7, as can also be seen in Table 1 of the main paper, our *smallest* FMA-Net variant ($M = 1$) even shows superior performance than the previous SOTA methods.

B. Detailed Experimental Results

B.1. FMA-Net$_s$

We first introduce FMA-Net$_s$, a lightweight model of FMA-Net. FMA-Net$_s$ is a model that changes the number of FRMA blocks, M, from the original 4 to 2, with no other modifications. Table 8 compares the quantitative performance of FMA-Net$_s$ on REDS4 [52] dataset with one VSRDB method (HOFFR [18]), four retrained SOTA methods (Restormer* [73], GShiftNet* [43], BasicVSR++* [9] , and RVRT* [48]) for VSRDB on REDS [52], and our FMA-Net. Our FMA-Net$_s$ demonstrates the second-best performance, maintaining performance while reducing memory usage and runtime.

B.2. Clip-by-clip Results on REDS4

Table 9 shows the performance of the clip-by-clip results on REDS4 [52] for VSRDB methods and all possible combinations of the sequential cascade approaches. It shows that our FMA-Net exhibits the best performance on all REDS4 clips consisting of realistic and dynamic scenes. In particular, compared to RVRT* [48], our FMA-Net achieves PSNR improvement of 0.35 dB in Clip 000, a scene with small motion, improvements of 1.62 dB and 1.58 dB in Clips 011 and 020, scenes with large motion, respectively. This demonstrates the superiority of FMA-Net over existing SOTA methods, especially in scenes with large motion.

B.3. Visualization Results

We show more qualitative comparison results among the proposed FMA-Net and other SOTA methods on two benchmark datasets. The results for REDS4 [52] and GoPro [51] are shown in Figs. 12-13 and Fig. 14, respectively.

B.4. Visual Comparisons with Demo Video

We provide a video at https://www.youtube.com/watch?v=kO7KavOH6vw to compare our FMA-Net with existing SOTA methods [9, 43, 48]. The demo video includes comparisons between FMA-Net and SOTA methods on two clips from the REDS4 [52] dataset and one clip from the GoPro [51] dataset.

C. Discussions

C.1. Learning Scheme

We train FMA-Net in a 2-stage manner which requires additional training time rather than end-to-end. This choice is made because, during the multi-attention process of Net^R, the warped feature F_w is adjusted by the predicted degradation from Net^D in a globally adaptive manner. When the network is trained end-to-end, in the initial training stages, F_w is adjusted for incorrectly predicted kernels due to the random initialization of weights, which adversely affects the training process (The performance comparison between end-to-end and 2-stage strategies can be found in Table 5(f, j))). To address this, we adopt a pre-training strategy for Net^D, which inevitably leads to longer training times compared to the end-to-end approach.

C.2. Limitation: Object Rotation

In extreme conditions such as object rotation, it is challenging to predict accurate optical flow, making precise restoration difficult. Fig. 15 illustrates the restoration results in a scene with object rotation, showing the failure of all methods, including our FMA-Net, in restoring a rotating object. The introduction of learnable homography parameters or the adoption of quaternion representations could be one option to enhance the performance in handling rotational motions.

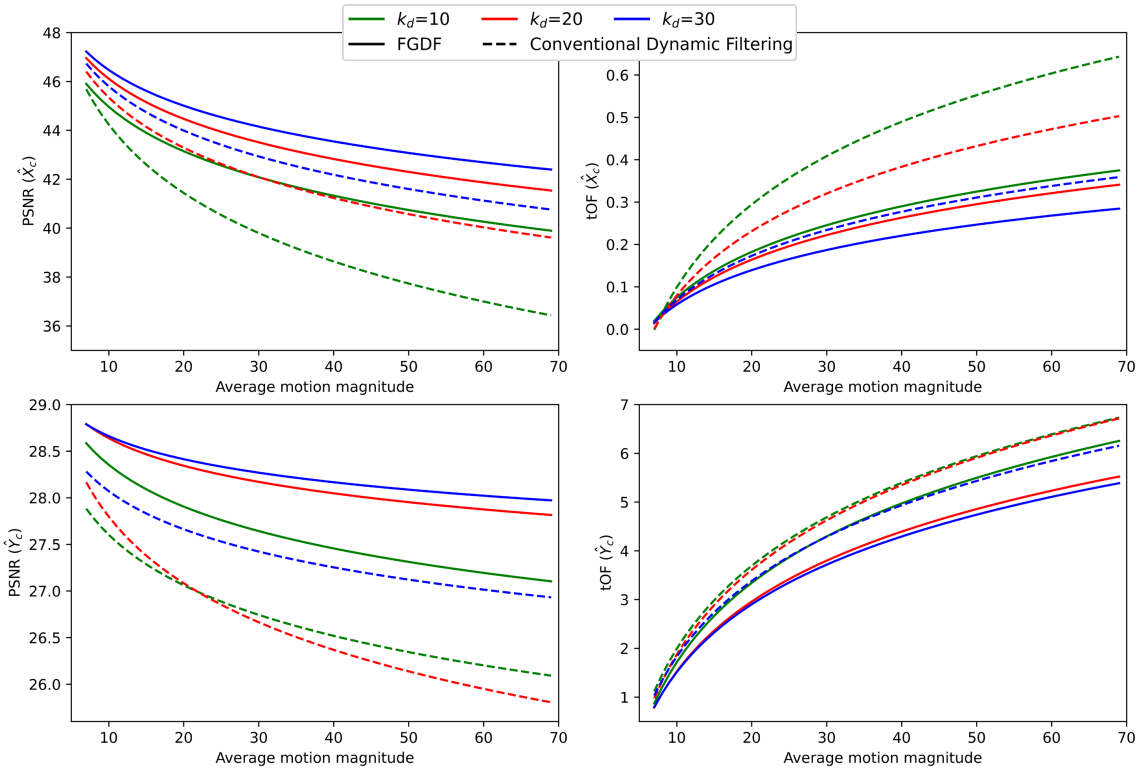


Figure 8. Flow-guided dynamic filtering (FGDF) vs. conventional dynamic filtering [33, 35, 38]. Trendline visualization for Table 3 of the main paper.

Methods	# Params (M)	Runtime (s)	Net^D (blurry LR $\hat{X}_c$) PSNR ↑ / SSIM ↑ / tOF ↓	Net^R (sharp HR $\hat{Y}_c$) PSNR ↑ / SSIM ↑ / tOF ↓
The number of multi-flow-mask pairs n				
(a) $n = 1$	9.15	0.424	44.80 / 0.9955 / 0.096	28.24 / 0.8151 / 2.224
(b) $n = 5$	9.29	0.429	45.37 / 0.9960 / 0.086	28.60 / 0.8258 / 2.054
Deformable Convolutions [13]				
(c) w/ DCNs (#offset = 9)	10.13	0.426	45.17 / 0.9956 / 0.093	28.52 / 0.8225 / 2.058
Loss Function and Training Strategy				
(d) w/o RAFT & TA Loss	9.62	0.434	45.28 / 0.9958 / 0.084	28.68 / 0.8274 / 2.003
(e) w/o TA Loss	9.62	0.434	45.33 / 0.9959 / 0.083	28.73 / 0.8288 / 1.956
(f) End-to-End Learning	9.62	0.434	44.14 / 0.9947 / 0.107	28.39 / 0.8190 / 2.152
Multi-Attention				
(g) self-attn [73] + SFT [68]	9.20	0.415	45.37 / 0.9959 / 0.085	28.50 / 0.8244 / 2.039
(h) CO attn + SFT [68]	9.20	0.416	**45.46** / **0.9961** / **0.082**	28.58 / 0.8262 / 1.938
(i) self-attn [73] + DA attn	9.62	0.434	45.37 / 0.9959 / 0.085	28.80 / 0.8298 / 1.956
(j) Ours	9.62	0.434	**45.46** / **0.9961** / **0.082**	**28.83** / **0.8315** / **1.918**

Table 5. Ablation study on the components in FMA-Net.

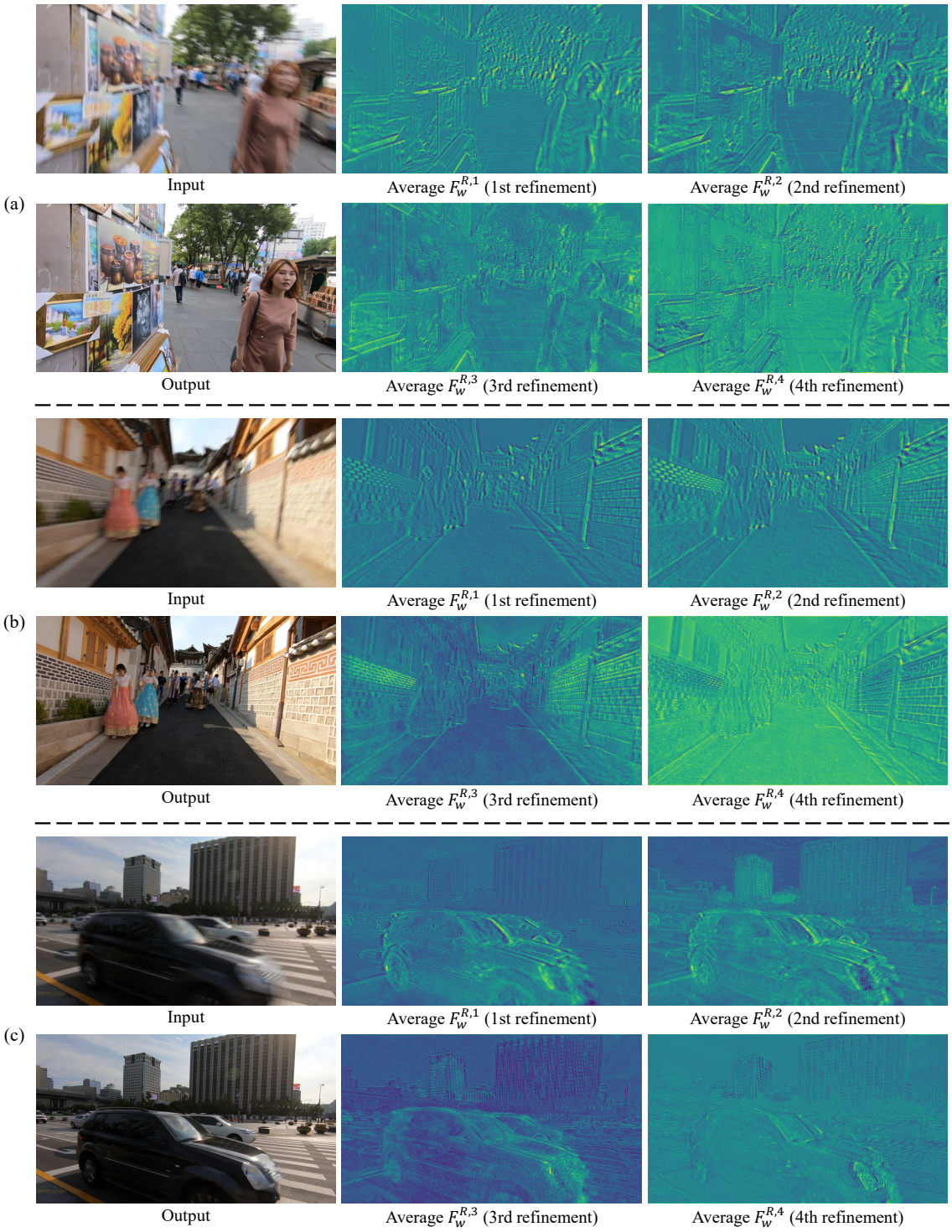
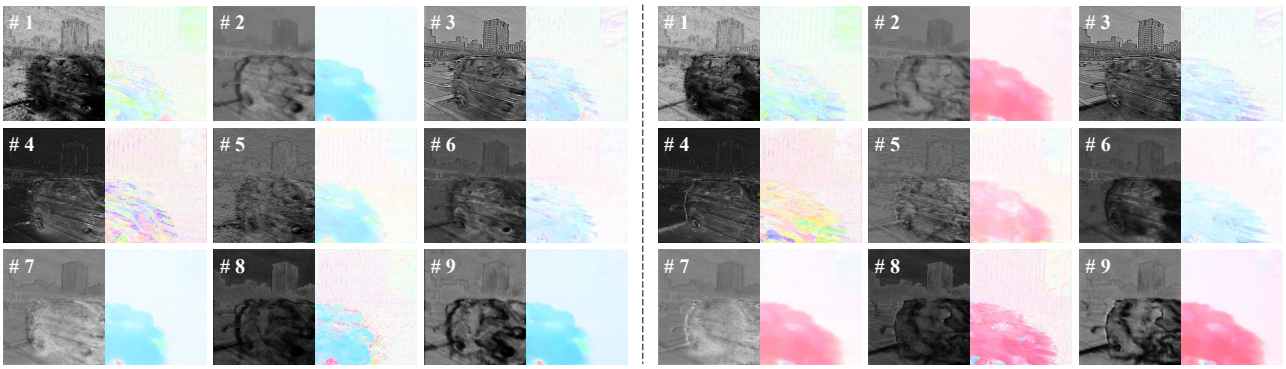


Figure 9. Visualization of iterative refinement process of warped feature $F_w^{R,i}$ in FRMA blocks of Net^R. The brighter the pixel, the more activated it is.



(a) Optical flow $\{f^j_{c\to c-1}\}_{j=1:n}$ and corresponding mask $\{o^j_{c\to c-1}\}_{j=1:n}$ (b) Optical flow $\{f^j_{c\to c+1}\}_{j=1:n}$ and corresponding mask $\{o^j_{c\to c+1}\}_{j=1:n}$

Figure 10. Visualisation of multi-flow-mask pairs $\mathbf{f}^{R,M}$ in Net^R.

T	# Params (M)	Runtime (s)	Net^D PSNR ↑ / SSIM ↑ / tOF ↓	Net^R PSNR ↑ / SSIM ↑ / tOF ↓
1	9.03	0.206	42.04 / 0.9908 / 0.182	27.33 / 0.7866 / 2.672
3	9.62	0.434	45.46 / 0.9961 / 0.082	28.83 / 0.8315 / 1.918
5	9.94	0.737	45.74 / 0.9965 / 0.076	28.92 / 0.8347 / 1.909
7	16.61	1.425	**46.24** / **0.9969** / **0.068**	**29.00** / **0.8376** / **1.856**

Table 6. Ablation study on the number of input frames T.

M	# Params (M)	Runtime (s)	PSNR ↑ / SSIM ↑ / tOF ↓
1	6.3	0.147	28.07 / 0.8109 / 2.24
2	7.4	0.231	28.46 / 0.8212 / 2.08
4	9.6	0.427	**28.83** / **0.8315** / **1.92**

Table 7. Ablation study on the number of FRMA blocks M.

Methods	# Params (M)	Runtime (s)	REDS4 PSNR ↑ / SSIM ↑ / tOF ↓
HOFFR [18]	3.5	0.500	27.24 / 0.7870 / -
Restormer* [73]	26.5	0.081	27.29 / 0.7850 / 2.71
GShiftNet* [43]	13.5	0.185	25.77 / 0.7275 / 2.96
BasicVSR++* [9]	7.3	0.072	27.06 / 0.7752 / 2.70
RVRT* [48]	12.9	0.680	27.80 / 0.8025 / 2.40
FMA-Net$_s$ (Ours)	7.4	0.231	28.46 / 0.8212 / 2.08
FMA-Net (Ours)	9.6	0.427	**28.83** / **0.8315** / **1.92**

Table 8. Quantitative comparison on REDS4 for $\times 4$ VSRDB. All results are calculated on the RGB channel. **Red** and blue colors indicate the best and second-best performance, respectively. Runtime is calculated on an LR frame sequence of size 180×320. The superscript * indicates that the model is retrained on the REDS [52] training dataset for VSRDB.

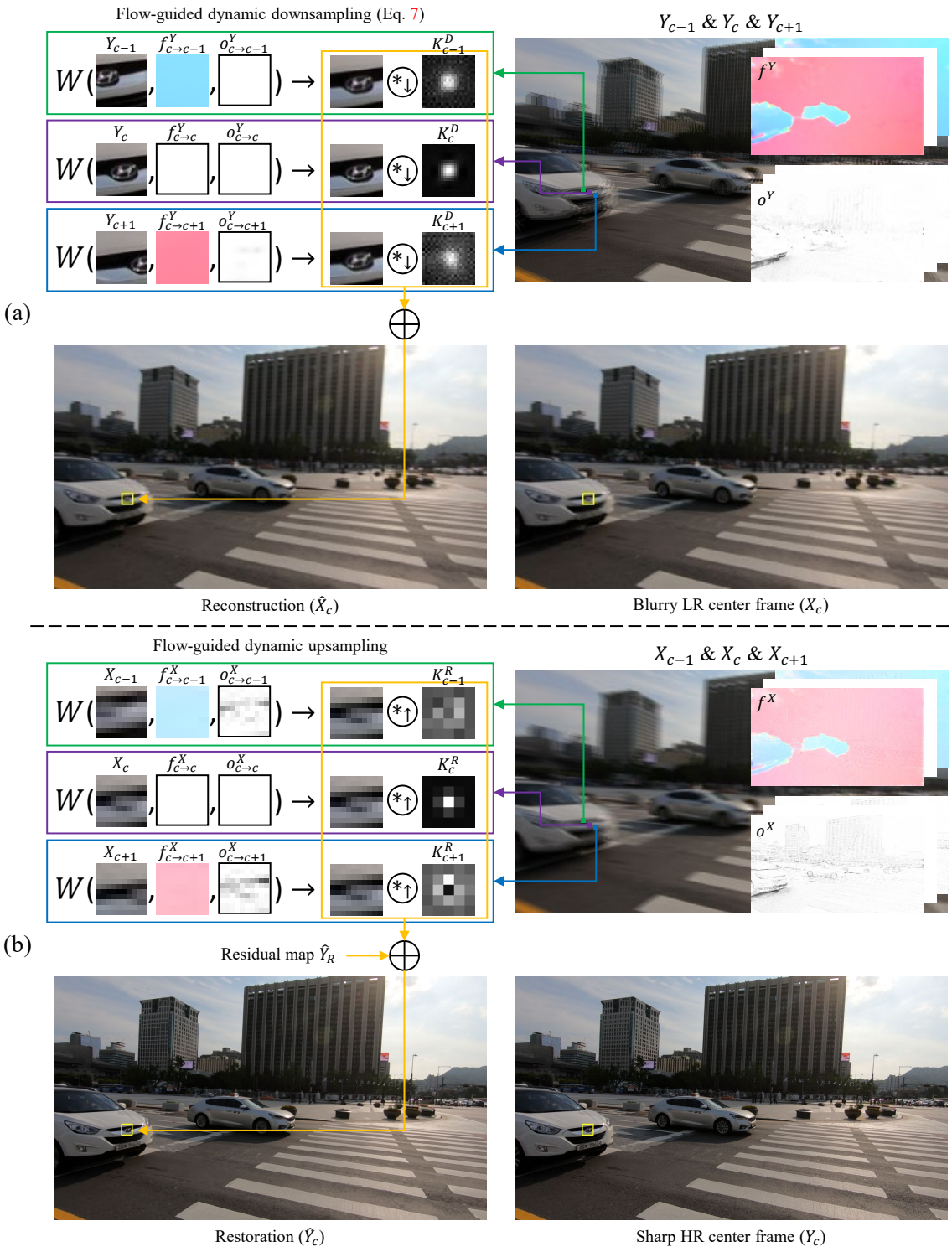


Figure 11. Visualization of the flow-guided dynamic filtering (FGDF) process, including two image flow-mask pairs ($\mathbf{f}^Y$ and $\mathbf{f}^X$) and two dynamic kernels (K^D and K^R): (a) Flow-guided dynamic downsampling (Eq. 7 of the main paper) with spatio-temporally variant degradation kernels K^D; (b) Flow-guided dynamic upsampling with spatio-temporally variant restoration kernels K^R.

REDS4 Methods	CLIP 000 PSNR ↑ / SSIM ↑ / tOF ↓	CLIP 011 PSNR ↑ / SSIM ↑ / tOF ↓	CLIP 015 PSNR ↑ / SSIM ↑ / tOF ↓	CLIP 020 PSNR ↑ / SSIM ↑ / tOF ↓	Average PSNR ↑ / SSIM ↑ / tOF ↓
Single Image Super-Resolution + Deblurring					
Bicubic + Restormer [73]	24.16 / 0.6488 / 0.95	22.92 / 0.6341 / 7.17	26.14 / 0.7565 / 4.73	21.24 / 0.6086 / 8.12	23.62 / 0.6620 / 5.24
Bicubic + FFTformer [41]	24.17 / 0.6432 / 0.90	22.90 / 0.6328 / 7.06	26.55 / 0.7669 / 4.85	21.27 / 0.6082 / 8.09	23.72 / 0.6628 / 5.23
Bicubic + RVRT [48]	24.21 / 0.6486 / 0.84	23.53 / 0.6818 / 4.87	26.58 / 0.7725 / 4.28	21.96 / 0.6641 / 5.90	24.07 / 0.6918 / 3.97
Bicubic + GShiftNet [43]	24.19 / 0.6468 / 0.80	23.36 / 0.6659 / 5.36	26.59 / 0.7742 / 4.31	21.76 / 0.6451 / 6.25	23.98 / 0.6830 / 4.18
SwinIR [46] + Restormer [73]	25.22 / 0.7136 / 0.68	23.17 / 0.6566 / 6.65	27.49 / 0.8142 / 4.18	21.47 / 0.6316 / 7.78	24.33 / 0.7040 / 4.82
SwinIR [46] [46] + FFTformer [41]	25.04 / 0.7096 / 0.66	23.06 / 0.6629 / 5.86	27.22 / 0.8183 / 3.94	21.40 / 0.6329 / 7.40	24.18 / 0.7059 / 4.47
SwinIR [46] + RVRT [48]	25.32 / 0.7261 / 0.58	23.97 / 0.7317 / 4.00	27.36 / 0.8232 / 3.64	22.11 / 0.6995 / 5.61	24.69 / 0.7451 / 3.46
SwinIR [46] + GShiftNet [43]	25.30 / 0.7221 / 0.58	23.59 / 0.6964 / 4.78	27.38 / 0.8219 / 3.67	21.81 / 0.6687 / 5.94	24.52 / 0.7273 / 3.74
HAT [10] + Restormer [73]	25.21 / 0.7151 / 0.68	23.18 / 0.6579 / 6.55	27.57 / 0.8184 / 4.07	21.48 / 0.6323 / 7.74	24.36 / 0.7059 / 4.76
HAT [10] + FFTformer [41]	25.11 / 0.7136 / 0.66	23.12 / 0.6670 / 5.75	27.26 / 0.8208 / 3.85	21.42 / 0.6348 / 7.35	24.22 / 0.7091 / 4.40
HAT [10] + RVRT [48]	25.39 / 0.7299 / 0.57	24.03 / 0.7354 / 3.93	27.48 / 0.8289 / 3.53	22.15 / 0.7022 / 5.54	24.76 / 0.7491 / 3.39
HAT [10] + GShiftNet [43]	25.36 / 0.7256 / 0.57	23.63 / 0.6989 / 4.73	27.48 / 0.8270 / 3.60	21.83 / 0.6699 / 5.91	24.58 / 0.7304 / 3.70
Video Super-Resolution + Deblurring					
BasicVSR++ [9] + Restormer [73]	26.35 / 0.7765 / 0.48	23.08 / 0.6527 / 6.83	28.07 / 0.8421 / 3.78	21.47 / 0.6325 / 7.83	24.74 / 0.7260 / 4.73
BasicVSR++ [9] + FFTformer [41]	26.20 / 0.7746 / 0.45	22.84 / 0.6479 / 6.34	27.46 / 0.8386 / 3.65	21.34 / 0.6286 / 7.62	24.46 / 0.7224 / 4.52
BasicVSR++ [9] + RVRT [48]	26.35 / 0.7897 / 0.40	23.70 / 0.7165 / 4.36	27.74 / 0.8438 / 3.32	21.98 / 0.6905 / 5.88	24.92 / 0.7604 / 3.49
BasicVSR++ [9] + GShiftNet [43]	26.30 / 0.7862 / 0.36	23.33 / 0.6824 / 4.97	27.65 / 0.8360 / 3.38	21.69 / 0.6627 / 6.03	24.74 / 0.7418 / 3.69
FTVSR [56] + Restormer [73]	26.31 / 0.7724 / 0.50	23.10 / 0.6533 / 6.81	27.92 / 0.8364 / 3.91	21.48 / 0.6329 / 7.83	24.70 / 0.7238 / 4.76
FTVSR [56] + FFTformer [41]	26.07 / 0.7679 / 0.51	22.83 / 0.6489 / 6.27	27.30 / 0.8328 / 3.76	21.32 / 0.6282 / 7.61	24.38 / 0.7195 / 4.54
FTVSR [56] + RVRT [48]	26.30 / 0.7863 / 0.42	23.73 / 0.7177 / 4.37	27.61 / 0.8392 / 3.40	22.02 / 0.6923 / 5.87	24.92 / 0.7589 / 3.52
FTVSR [56] + GShiftNet [43]	26.26 / 0.7827 / 0.38	23.36 / 0.6845 / 4.95	27.52 / 0.8329 / 3.45	21.75 / 0.6658 / 5.99	24.72 / 0.7415 / 3.69
Single Image Deblurring + Super-Resolution					
Restormer [73] + Bicubic	24.04 / 0.6404 / 0.93	22.96 / 0.6359 / 6.77	26.47 / 0.7613 / 5.00	21.44 / 0.6185 / 7.37	23.73 / 0.6640 / 5.02
Restormer [73] + SwinIR [46]	24.96 / 0.7135 / 0.68	23.21 / 0.6647 / 6.14	27.43 / 0.8117 / 4.18	21.58 / 0.6442 / 6.97	24.30 / 0.7085 / 4.49
Restormer [73] + HAT [10]	25.03 / 0.7162 / 0.67	23.22 / 0.6650 / 6.12	27.50 / 0.8142 / 4.12	21.58 / 0.6444 / 6.95	24.33 / 0.7100 / 4.47
Restormer [73] + BasicVSR++ [9]	25.80 / 0.7740 / 0.49	23.19 / 0.6638 / 6.12	27.78 / 0.8268 / 3.88	21.59 / 0.6472 / 6.93	24.59 / 0.7280 / 4.36
Restormer [73] + FTVSR [56]	25.79 / 0.7709 / 0.50	23.22 / 0.6651 / 6.19	27.71 / 0.8260 / 3.91	21.61 / 0.6481 / 6.98	24.58 / 0.7275 / 4.40
FFTformer [41] + Bicubic	23.98 / 0.6416 / 0.87	22.82 / 0.6382 / 6.50	26.38 / 0.7610 / 4.91	21.42 / 0.6207 / 7.22	23.65 / 0.6654 / 4.88
FFTformer [41] + SwinIR [46]	24.83 / 0.7149 / 0.67	23.01 / 0.6657 / 5.93	27.28 / 0.8115 / 4.17	21.53 / 0.6462 / 6.82	24.16 / 0.7096 / 4.40
FFTformer [41] + HAT [10]	24.90 / 0.7177 / 0.66	23.03 / 0.6661 / 5.90	27.37 / 0.8141 / 4.14	21.53 / 0.6464 / 6.81	24.21 / 0.7111 / 4.38
FFTformer [41] + BasicVSR++ [9]	25.72 / 0.7774 / 0.48	23.00 / 0.6654 / 5.92	27.61 / 0.8273 / 3.89	21.54 / 0.6492 / 6.78	24.47 / 0.7298 / 4.27
FFTformer [41] + FTVSR [56]	25.73 / 0.7742 / 0.52	23.03 / 0.6673 / 6.00	27.47 / 0.8257 / 3.95	21.56 / 0.6505 / 6.85	24.45 / 0.7294 / 4.33
Video Deblurring + Super-Resolution					
RVRT [48] + Bicubic	24.03 / 0.6356 / 0.98	23.33 / 0.6540 / 5.58	26.51 / 0.7645 / 4.57	21.91 / 0.6436 / 5.94	23.95 / 0.6744 / 4.27
RVRT [48] + SwinIR [46]	25.11 / 0.7092 / 0.71	23.57 / 0.6826 / 5.03	27.33 / 0.8121 / 3.80	22.04 / 0.6688 / 5.53	24.51 / 0.7182 / 3.77
RVRT [48] + HAT [10]	25.15 / 0.7114 / 0.70	23.58 / 0.6827 / 5.01	27.40 / 0.8143 / 3.75	22.04 / 0.6688 / 5.52	24.54 / 0.7193 / 3.75
RVRT [48] + BasicVSR++ [9]	25.96 / 0.7650 / 0.58	23.56 / 0.6832 / 5.01	27.56 / 0.8237 / 3.54	22.06 / 0.6725 / 5.49	24.79 / 0.7361 / 3.66
RVRT [48] + FTVSR [56]	25.94 / 0.7618 / 0.60	23.70 / 0.6964 / 5.34	27.49 / 0.8229 / 3.63	22.17 / 0.6848 / 6.02	24.83 / 0.7415 / 3.90
GShiftNet [43] + Bicubic	21.37 / 0.5874 / 1.28	23.20 / 0.6488 / 5.80	26.54 / 0.7650 / 4.61	21.72 / 0.6339 / 6.27	23.21 / 0.6588 / 4.49
GShiftNet [43] + SwinIR [46]	20.94 / 0.6163 / 1.07	23.34 / 0.6755 / 5.25	27.39 / 0.8140 / 3.85	21.80 / 0.6585 / 5.90	23.37 / 0.6911 / 4.02
GShiftNet [43] + HAT [10]	20.99 / 0.6182 / 1.06	23.36 / 0.6757 / 5.23	27.48 / 0.8168 / 3.80	21.80 / 0.6587 / 5.89	23.41 / 0.6924 / 4.00
GShiftNet [43] + BasicVSR++ [9]	20.98 / 0.6432 / 0.88	23.35 / 0.6766 / 5.21	27.66 / 0.8278 / 3.53	21.82 / 0.6621 / 5.84	23.45 / 0.7024 / 3.87
GShiftNet [43] + FTVSR [56]	21.05 / 0.6439 / 0.90	23.42 / 0.6814 / 5.41	27.56 / 0.8267 / 3.57	21.87 / 0.6657 / 6.03	23.47 / 0.7044 / 3.98
Joint Video Super-Resolution and Deblurring					
HOFFR [18]	- / - / -	- / - / -	- / - / -	- / - / -	27.24 / 0.7870 / -
Restormer* [73]	26.51 / 0.7551 / 0.47	27.09 / 0.7695 / 3.53	30.03 / 0.8579 / 2.82	25.52 / 0.7573 / 4.04	27.29 / 0.7850 / 2.72
GShiftNet* [43]	24.66 / 0.6730 / 0.93	25.66 / 0.7190 / 3.47	28.05 / 0.7995 / 3.50	24.69 / 0.7187 / 3.93	25.77 / 0.7275 / 2.96
BasicVSR++* [9]	25.90 / 0.7234 / 0.57	27.07 / 0.7699 / 3.36	29.67 / 0.8475 / 3.01	25.58 / 0.7601 / 3.86	27.06 / 0.7752 / 2.70
RVRT* [48]	26.84 / 0.7764 / 0.38	27.76 / 0.7903 / 2.95	30.66 / 0.8694 / 2.60	25.93 / 0.7740 / 3.65	27.80 / 0.8025 / 2.40
FMA-Net$_s$ (Ours)	27.08 / 0.7852 / 0.33	28.73 / 0.8164 / 2.46	30.98 / 0.8745 / 2.42	27.03 / 0.8089 / 3.10	28.46 / 0.8212 / 2.08
FMA-Net (Ours)	**27.19** / **0.7904** / **0.32**	**29.38** / **0.8308** / **2.19**	**31.36** / **0.8814** / **2.37**	**27.51** / **0.8232** / **2.79**	**28.83** / **0.8315** / **1.92**

Table 9. Quantitative comparison on REDS4 for $\times 4$ VSRDB. All results are calculated on the RGB channel. **Red** and blue colors indicate the best and second-best performance, respectively. The superscript * indicates that the model is retrained on the REDS [52] training dataset for VSRDB.

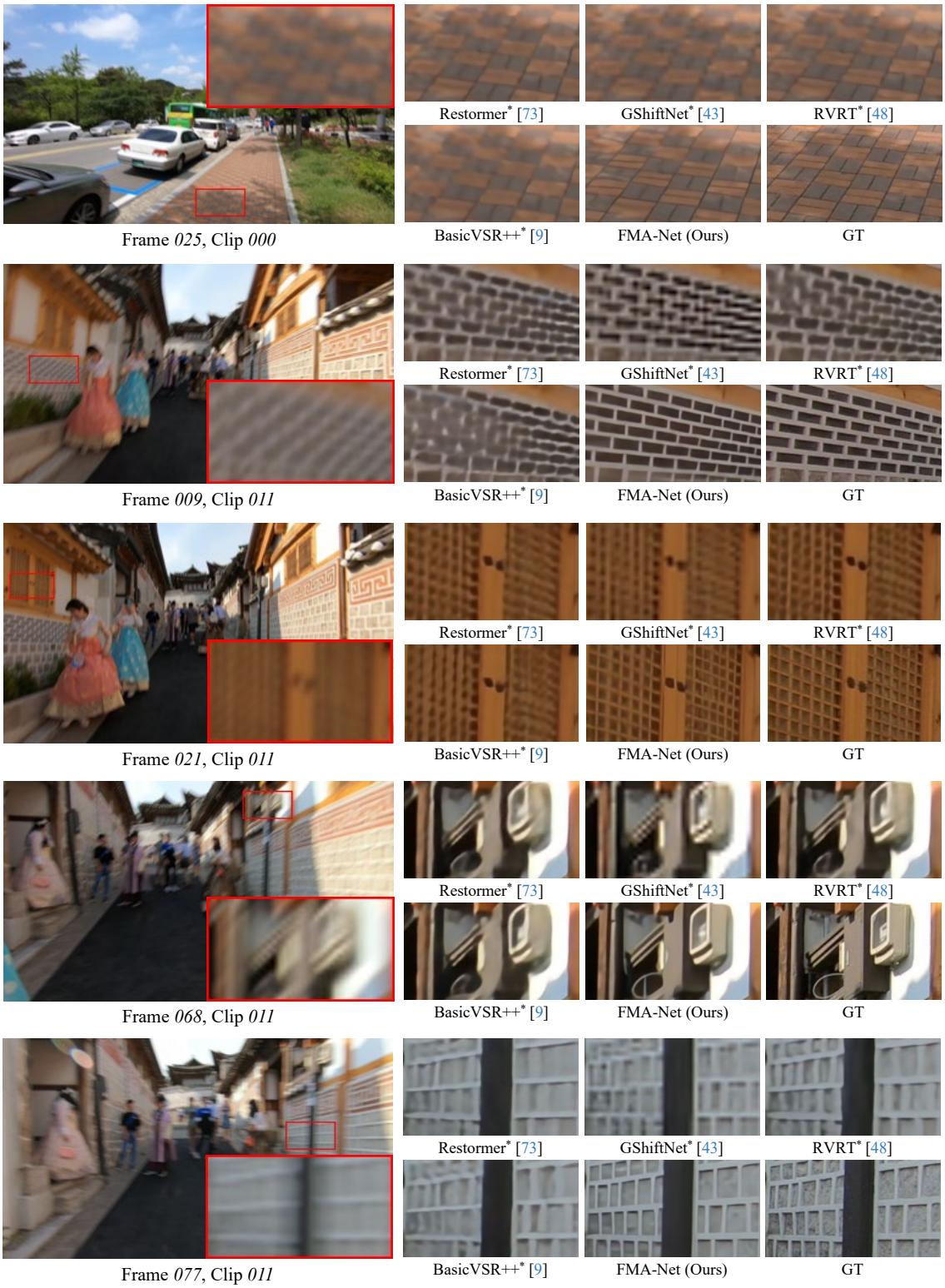


Figure 12. Visual results of different methods on REDS4 [52]. *Best viewed in zoom.*

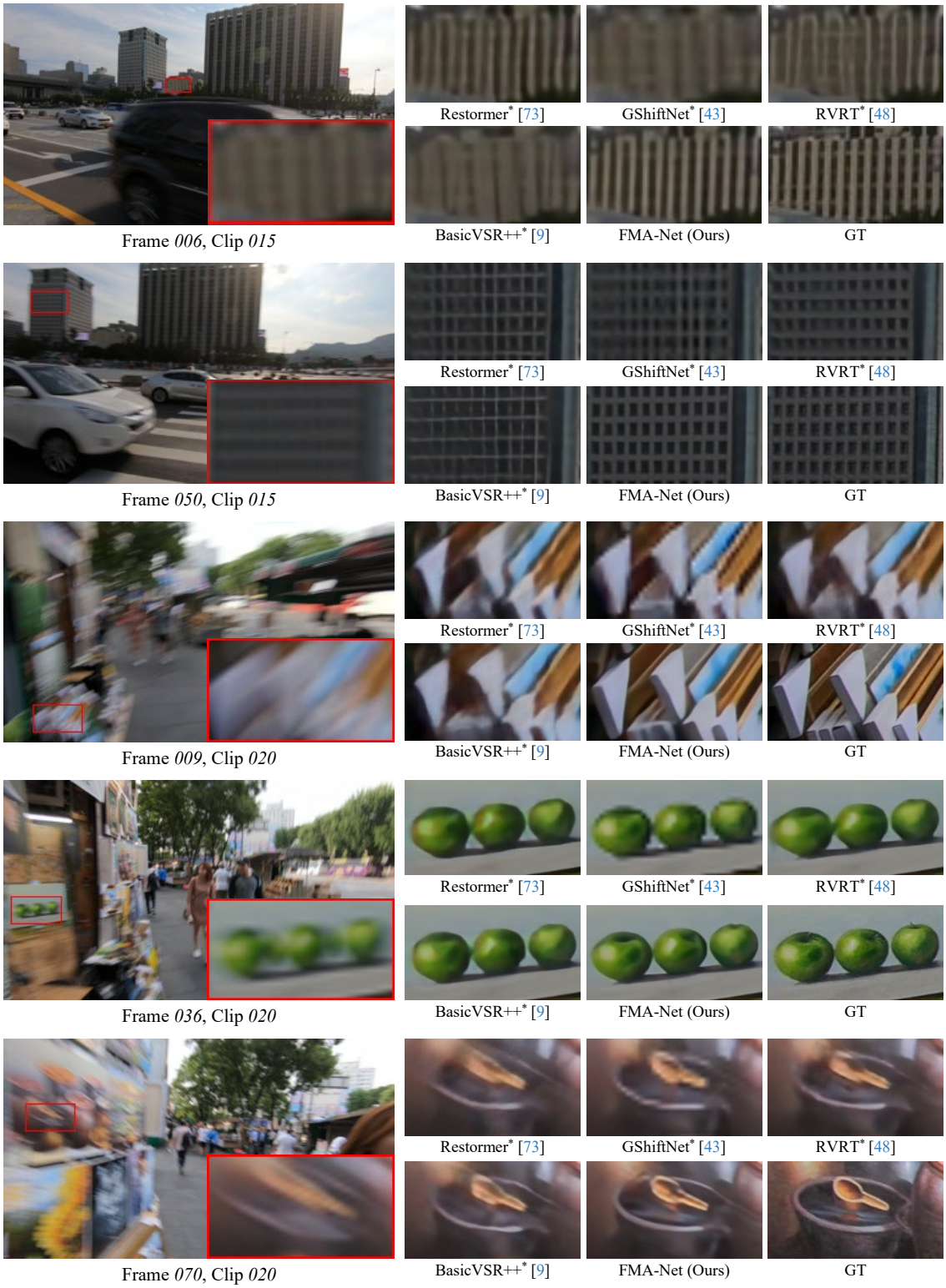


Figure 13. Visual results of different methods on REDS4 [52]. *Best viewed in zoom.*

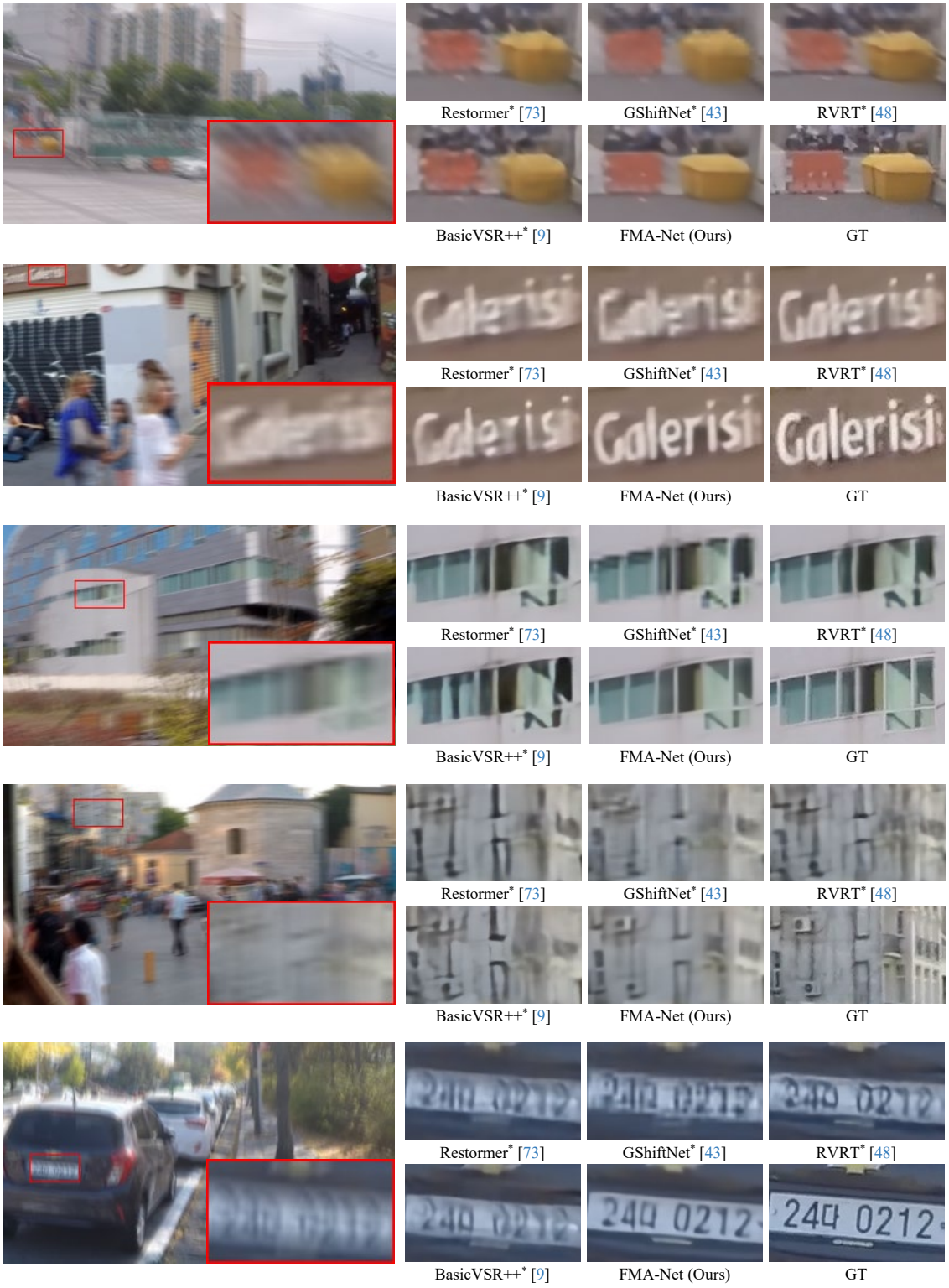


Figure 14. Visual results of different methods on GoPro [52] test set. *Best viewed in zoom.*

Figure 15. Qualitative comparison for the extreme scene including object rotation.