BiM-VFI: Bidirectional Motion Field-Guided Frame Interpolation for Video with Non-uniform Motions

Supplementary Material

In this supplementary material, we first provide additional details of our proposed BiM-VFI. Especially, the detailed network structure of CAUN and SN, loss functions, implementation details, and the proof of how BiM can describe bidirectional motion distinctly are explained in Appendix A. Subsequently, in Appendix B, we provided additional experimental results that could not be included in the main paper due to the page limitation. In Appendix B.1, pixel-centric metrics (PSNR and SSIM) in SNU-FILM-arb [6], Vimeo90K-triplet [41], SNU-FILM [3], and XTest-single [35], and $\times 8$ interpolation on XTest dataset are provided. Also, in Appendix B.4, we provided the results of the user study we conducted for interpolated videos from various VFI methods. Lastly, in Appendix B.5, we provided additional qualitative comparisons on SNU-FILM-arb [6] datasets.

A. Additional Details

A.1. Structure of Content-Aware Upsampling Network (CAUN)

Fig. 7 depicts the detailed architecture of our proposed Content-Aware Upsampling Network, CAUN (Sec. 3.3). CAUN is designed to construct adaptive upsampling kernels that upsample flows while preserving high-frequency details, especially sharp boundaries and small objects. For this, CAUN effectively utilizes and integrates multi-scale features. Context features $F_0^{l,c}$ and $F_1^{l,c}$ are consists of multi-scale features $(F_0^{l,c,0}, F_0^{l,c,1}, F_0^{l,c,2})$ and $(F_1^{l,c,0}, F_1^{l,c,1}, F_1^{l,c,2})$, respectively, where $F_i^{l,c,j}$ is $H/2^j \times W/2^j$-sized context feature map of I_i^l for $i \in \{0,1\}$ and $j \in \{0,1,2\}$. Note that $F_0^{l,c,2}$ and $F_1^{l,c,2}$ are of the same spatial sizes as $\tilde{\mathbf{V}}_{t\to 0}^l$ and $\tilde{\mathbf{V}}_{t\to 1}^l$. So, the context features $F_0^{l,c,2}$ and $F_1^{l,c,2}$ can be directly aligned to target time t by warping via $\tilde{\mathbf{V}}_{t\to 0}^l$ and $\tilde{\mathbf{V}}_{t\to 1}^l$, respectively. However, to warp $F_0^{l,c,1}$ and $F_1^{l,c,1}$, the two flows $\tilde{\mathbf{V}}_{t\to 0}^l$ and $\tilde{\mathbf{V}}_{t\to 1}^l$ must be bilinearly upsampled by a factor of 2 and their magnitudes are scaled by a factor of 2 to match with the spatial size of the features $F_0^{l,c,1}$ and $F_1^{l,c,1}$. In this sense, $\tilde{\mathbf{V}}_{t\to 0}^l$ and $\tilde{\mathbf{V}}_{t\to 1}^l$ are further upsampled by a factor of 4, and their magnitudes are scaled by a factor of 4 to warp the features $F_0^{l,c,0}$ and $F_1^{l,c,0}$. Then, the warped features are concatenated and further passed through several convolution layers and PixelShuffle layers to integrate multi-scale features. Finally, adaptive kernels $K_{t\to 0}^{l,\times 2}$, $K_{t\to 1}^{l,\times 2}$, $K_{t\to 0}^{l,\times 4}$, and $K_{t\to 0}^{l,\times 4}$ are obtained for input with the integrated multi-scale features, where $K_{t\to 0}^{l,\times 2}, K_{t\to 1}^{l,\times 2} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 9\times 4}$ and $K_{t\to 0}^{l,\times 4}, K_{t\to 1}^{l,\times 4} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 9\times 16}$. $K_{t\to 0}^{l,\times 2}$ and $K_{t\to 1}^{l,\times 2}$ are pixel-wise convolved with 3×3 neighboring pixels of $\tilde{\mathbf{V}}_{t\to 0}^l$ and $\tilde{\mathbf{V}}_{t\to 1}^l$, respectively, to adaptively upsample $\tilde{\mathbf{V}}_{t\to 0}^l$ and $\tilde{\mathbf{V}}_{t\to 1}^l$ by a factor of 2, thus yielding $\mathbf{V}_{t\to 0}^{l,\times 0.5}$ and $\mathbf{V}_{t\to 1}^{l,\times 0.5}$. Similarly, $K_{t\to 0}^{l,\times 4}$ and $K_{t\to 0}^{l,\times 4}$ are pixel-wise convolved with 3×3 neighboring pixels of $\tilde{\mathbf{V}}_{t\to 0}^l$ and $\tilde{\mathbf{V}}_{t\to 1}^l$, then yielding $\mathbf{V}_{t\to 0}^l$ and $\mathbf{V}_{t\to 1}^l$, respectively, where $\mathbf{V}_{t\to 0}^l$ and $\mathbf{V}_{t\to 1}^l$ are of the same sizes as those of the source images at l-th level, I_0^l and I_1^l. $\mathbf{V}_{t\to 0}^{l,\times 0.5}$, $\mathbf{V}_{t\to 1}^{l,\times 0.5}$, $\mathbf{V}_{t\to 0}^l$, and $\mathbf{V}_{t\to 1}^l$ are further utilized in Synthesis Network (SN) to warp the source images and their context features with more precise flows.

A.2. Structure of Synthesis Network (SN)

We employed a simple U-Net [33] for our Synthesis Network (SN) as depicted in Fig. 8. The multi-scale flows from CAUN, which include $(\tilde{\mathbf{V}}_{t\to 0}^l, \tilde{\mathbf{V}}_{t\to 1}^l)$, $(\mathbf{V}_{t\to 1}^{0,\times 0.5}, \mathbf{V}_{t\to 1}^{l,\times 0.5})$ and $(\mathbf{V}_{t\to 0}^l, \mathbf{V}_{t\to 1}^l)$, are used to warp the multi-scale context features $(F_0^{l,c,2}, F_1^{l,c,2})$, $(F_0^{l,c,1}, F_1^{l,c,1})$, and $(F_0^{l,c,0}, F_1^{l,c,0})$. As depicted in Fig. 8, the warped multi-scale context features are passed through the U-Net, finally yielding a blending mask O^l and a residual image $\hat{I}_t^{l,\text{res}}$ at l-th level. The resulting $\hat{I}_t^{l,\text{res}}$ and O^l from the U-Net are employed to construct final interpolation result at l-th level, $\hat{I}_t^l$, as follow:

$$\begin{aligned}\hat{I}_t^l = &\,\text{bw}(I_0^l, \mathbf{V}_{t\to 0}^l) * \sigma(O^l) + \\ &\,\text{bw}(I_1^l, \mathbf{V}_{t\to 1}^l) * (1-\sigma(O^l)) + \hat{I}_t^{l,\text{res}},\end{aligned} \tag{6}$$

where $\text{bw}(\cdot,\cdot)$ is a backward warping function and $\sigma(\cdot)$ is a sigmoid function.

A.3. Loss Functions

Our training objectives consist of student loss $\mathcal{L}_{\mathcal{P}_\mathcal{S}}$, and teacher loss $\mathcal{L}_{\mathcal{P}_\mathcal{T}}$. The teacher loss will be discussed first, and followed by the student loss. To supervise photometric reconstruction of the teacher process, the charbonnier loss [1] $\mathcal{L}_{\text{char}}$ and the census loss [24] $\mathcal{L}_{\text{css}}$ are used as follow:

$$\begin{aligned}\mathcal{L}_{\text{char},\mathcal{P}_\mathcal{T}}^l &= \lambda_{\text{char},\mathcal{P}_\mathcal{T}} \mathcal{L}_{\text{char}}(\hat{I}^{l,\mathcal{P}_\mathcal{T}}, I^l), \\ \mathcal{L}_{\text{css},\mathcal{P}_\mathcal{T}}^l &= \lambda_{\text{css},\mathcal{P}_\mathcal{T}} \mathcal{L}_{\text{css}}(\hat{I}^{l,\mathcal{P}_\mathcal{T}}, I^l), \\ \mathcal{L}_{\text{pho},\mathcal{P}_\mathcal{T}}^l &= \mathcal{L}_{\text{char},\mathcal{P}_\mathcal{T}}^l + \mathcal{L}_{\text{css},\mathcal{P}_\mathcal{T}}^l,\end{aligned} \tag{7}$$

where l is the current pyramid level, $\lambda_{\text{char},\mathcal{P}_\mathcal{T}}$ and $\lambda_{\text{css},\mathcal{P}_\mathcal{T}}$ are weights for each loss. Furthermore, the first-order edge-aware smoothness loss [14] $\mathcal{L}_{\text{s1}}$ is used to ensure smooth

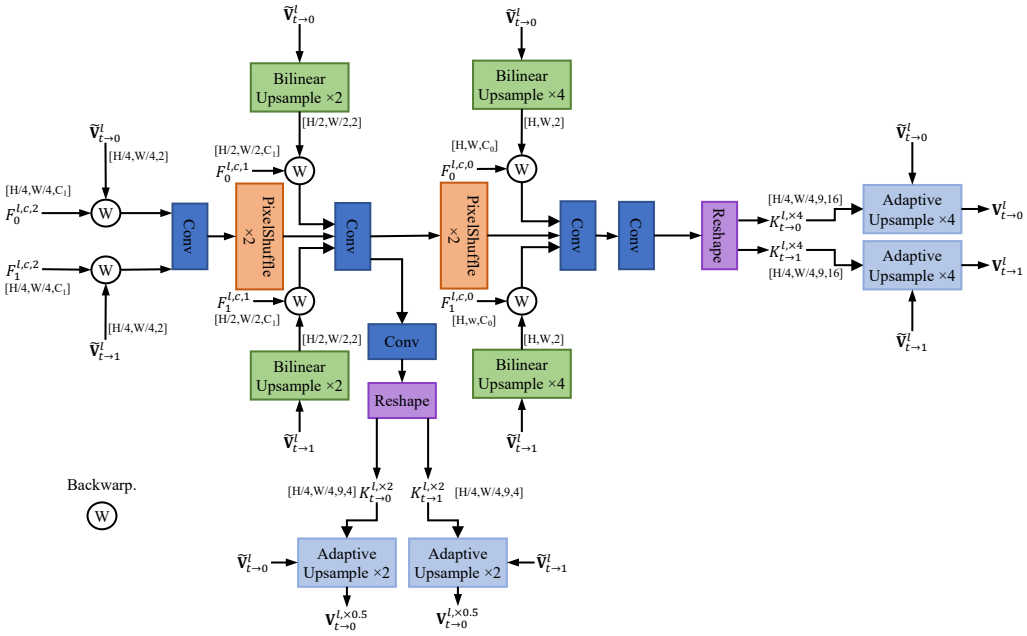


Figure 7. Detailed architecture of our Content-Aware Upsampling Network (CAUN).

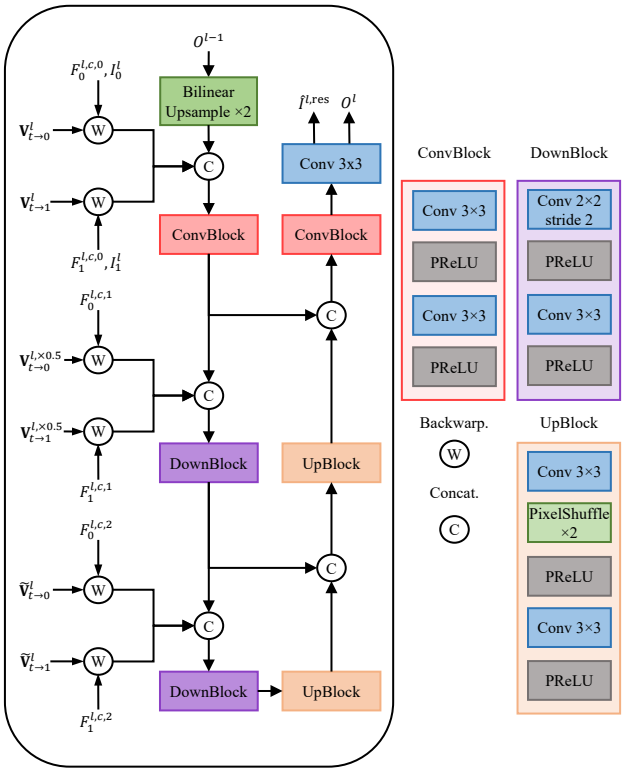


Figure 8. Detailed architecture of our Synthesis Network (SN).

teacher flows excluding object boundaries, and the regularization loss $\mathcal{L}_{\text{reg}}$ is utilized to force $V^{l,\mathcal{P}_\mathcal{T}}_{t\to t|0t}$ and $V^{l,\mathcal{P}_\mathcal{T}}_{t\to t|t1}$ to be uniform vector fields of all zeros, where the input BiM of teacher process (Eq. (4), Eq. (5)) is used to guide the two flows to be zero flows:

$$\begin{aligned}\mathcal{L}^l_{\text{s1}} &= \lambda_{\text{s1}}(\mathcal{L}_{\text{s1}}(V^{l,\mathcal{P}_\mathcal{T}}_{t\to 0}) + \mathcal{L}_{\text{s1}}(V^{l,\mathcal{P}_\mathcal{T}}_{t\to 1})),\\ \mathcal{L}^l_{\text{reg}} &= \lambda_{\text{reg}}(\mathcal{L}_2(V^{l,\mathcal{P}_\mathcal{T}}_{t\to t|0t}) + \mathcal{L}_2(V^{l,\mathcal{P}_\mathcal{T}}_{t\to t|t1})),\\ \mathcal{L}^l_{\text{flo},\mathcal{P}_\mathcal{T}} &= \mathcal{L}^l_{\text{s1}} + \mathcal{L}^l_{\text{reg}},\end{aligned} \tag{8}$$

where λ_{s1} and λ_{reg} are weights for each loss.

The photometric loss for the student process is constructed in the same manner as the teacher process, which is given by:

$$\begin{aligned}\mathcal{L}^l_{\text{char},\mathcal{P}_\mathcal{S}} &= \lambda_{\text{char},\mathcal{P}_\mathcal{S}}\mathcal{L}_{\text{char}}(\hat{I}^{l,\mathcal{P}_\mathcal{S}}, I^l),\\ \mathcal{L}^l_{\text{css},\mathcal{P}_\mathcal{S}} &= \lambda_{\text{css},\mathcal{P}_\mathcal{S}}\mathcal{L}_{\text{css}}(\hat{I}^{l,\mathcal{P}_\mathcal{S}}, I^l),\\ \mathcal{L}^l_{\text{pho},\mathcal{P}_\mathcal{S}} &= \mathcal{L}^l_{\text{char},\mathcal{P}_\mathcal{S}} + \mathcal{L}^l_{\text{css},\mathcal{P}_\mathcal{S}},\end{aligned} \tag{9}$$

where $\lambda_{\text{char},\mathcal{P}_\mathcal{S}}$ and $\lambda_{\text{css},\mathcal{P}_\mathcal{S}}$ are the weights for their respective losses. The flows of the student process will be supervised by a flow distillation loss that enforces the flows of the student process to get closer to those of the teacher process, which is given by:

$$\begin{aligned}\mathcal{L}^l_{\text{flo},\mathcal{P}_\mathcal{S}} = \lambda_{\text{distill}}(&\mathcal{L}_2(V^{l,\mathcal{P}_\mathcal{S}}_{t\to 0} - \text{sg}(V^{l,\mathcal{P}_\mathcal{T}}_{t\to 0}))\\ &+\mathcal{L}_2(V^{l,\mathcal{P}_\mathcal{S}}_{t\to 1} - \text{sg}(V^{\mathcal{P}_\mathcal{T}}_{t\to 1}))),\end{aligned} \tag{10}$$

where λ_{distill} is a weighting factor, and $\text{sg}(\cdot)$ is a stop gradient function that is used to force the gradients to be only

Methods	SNU-FILM-arb						XTest				
	medium		hard		extreme		×8				
	psnr	ssim	psnr	ssim	psnr	ssim	psnr	ssim	lpips	stlpips	niqe
RIFE [9]	36.31	0.981	31.86	0.952	27.20	0.895	30.58	0.904	0.153	0.114	7.393
IFRNet [16]	34.82	0.976	31.11	0.947	26.29	0.882	26.36	0.826	0.198	0.147	**5.842**
M2M-PWC [7]	36.54	**0.982**	31.92	0.951	27.13	0.892	30.81	0.912	0.080	0.047	6.521
AMT-S [18]	34.42	0.974	30.98	0.947	26.42	0.887	28.16	0.873	0.187	0.134	7.082
UPRNet [13]	36.70	**0.982**	31.9	0.951	27.08	0.893	30.50	0.905	0.093	0.058	6.148
EMA-VFI [45]	**36.85**	**0.982**	**32.7**	**0.957**	**28.15**	**0.906**	**31.36**	**0.914**	0.165	0.130	7.77
RIFE[D,R] [44]	36.17	0.981	31.59	0.949	27.05	0.891	26.93	0.839	0.232	0.169	6.477
IFRNet[D,R] [44]	35.92	0.981	31.18	0.947	26.54	0.886	28.76	0.891	0.147	0.096	7.054
AMT-S[D,R] [44]	34.78	0.978	30.48	0.944	26.15	0.886	29.27	0.886	0.098	0.055	6.409
EMA-VFI[D,R] [44]	35.75	0.980	31.02	0.946	26.37	0.885	25.75	0.833	0.258	0.192	6.928
ours	36.57	**0.982**	31.92	0.949	27.22	0.891	30.80	**0.914**	**0.068**	**0.045**	6.449

Table 4. Additional quantitative comparisons on arbitrary time interpolation datasets.

activated for the student process. The overall loss for our BiM-VFI with KDVCF is defined as:

$$\mathcal{L} = \sum_{l=0}^{L-1} \gamma_{\text{pho}}^{l} (\mathcal{L}_{\text{pho},\mathcal{P}_\mathcal{T}}^{l} + \mathcal{L}_{\text{pho},\mathcal{P}_\mathcal{S}}) + \gamma_{\text{flo}}^{l} (\mathcal{L}_{\text{flo},\mathcal{P}_\mathcal{T}} + \mathcal{L}_{\text{flo},\mathcal{P}_\mathcal{S}}), \tag{11}$$

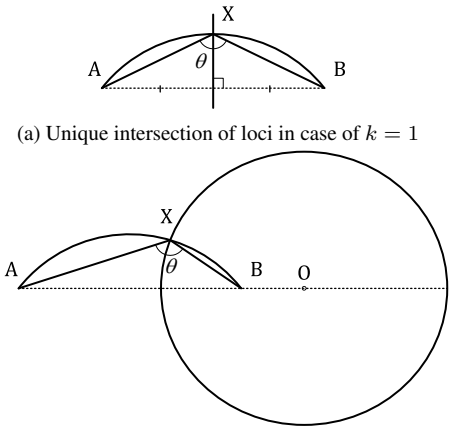
where L is the total number of pyramid levels used in training, and γ_{pho}, and γ_{flo} are exponential weights for the photometric loss and the flow-centric loss, respectively, which are employed to weigh more supervision on larger-sized image resolutions.

A.4. Implementation details

We trained our BiM-VFI with a training split of Vimeo90k septuplet datasets [41]. We randomly crop the images to a resolution of 256 × 256, flip horizontally and vertically, rotate, reverse temporally, and permute the color channels to augment the training data. We set the batch size to 32, and train the model for 400 epochs with an initial learning rate of 1×10^{-4}. We gradually decay the initial learning rate using a Cosine annealing scheduler [21] and optimize our model using the AdamW optimizer [20]. Also, because the architecture of our BiM-VFI is based on a recurrent pyramid architecture, we employed resolution-aware adaptation for the pyramid hierarchy depth proposed by Jin *et al*. [13]. For training on Vimeo90K, we used 3 pyramid levels, while 5 pyramid levels are used for SNU-FILM [3] and SNU-FILM-arb [6], and 7 pyramid levels for Xtest [35]. As mentioned, our proposed KDVCF computes the BiM during training, and for inference time, the BiM is represented according to Eq. (2) corresponding to a uniform motion scenario.

A.5. Distinct Description of BiM

As discussed in Sec. 3.1 of the main paper, we proposed BiM as a distinct motion descriptor for non-uniform motions, including accelerations, decelerations, and changing directions. To ensure the distinct descriptive power of BiM, we provide a mathematical analysis of how our BiM can explain the position of the intermediate pixel between given two corresponding pixels.



(a) Unique intersection of loci in case of $k = 1$

(b) Unique intersection of loci in case of $k \neq 1$

Figure 9. Visualization of intersection by two loci.

Theorem 1. *Let A and B be two fixed points, and let k be a positive real number. The point X such that the distance ratio $\frac{\overline{AX}}{\overline{BX}} = k$ and the angle $\angle AXB = \theta$ is unique.*

Proof. We start by describing the locus of points X' where $\angle AX'B = \theta$. This locus forms an arc $\overset{\frown}{AB}$ where any point X' on the arc $\overset{\frown}{AB}$ satisfying $\angle AX'B = \theta$.

The locus of points X'' where $\frac{\overline{AX''}}{\overline{BX''}} = k$ varies inshape depending on the value of k.

I) If $k = 1$ (Fig. 9a), this locus forms a perpendicular bisector of $\overline{AB}$. In this case, the intersection of the arc $\overset{\frown}{AB}$ and the perpendicular bisector of $\overline{AB}$ is unique, thus the

Methods	Vimeo 90K-triplet		SNU-FILM								XTest		Complexity	
			easy		medium		hard		extreme		single		FLOPs	Params
	psnr	ssim	psnr	ssim	psnr	ssim	psnr	ssim	psnr	ssim	psnr	ssim	(T)	(M)
AMT-G [18]	36.53	**0.982**	39.88	**0.991**	36.12	**0.981**	30.78	**0.981**	25.43	0.865	**30.34**	**0.904**	2.07	30.6
M2M-PWC [7]	35.49	0.978	39.66	**0.991**	35.74	0.980	30.32	0.980	25.07	0.863	**30.81**	0.900	0.26	7.6
UPRNet [13]	36.42	**0.982**	**40.44**	**0.991**	**36.29**	0.980	30.86	0.938	25.63	0.864	30.27	0.897	1.59	6.6
RIFE [9]	35.61	0.978	40.02	0.990	35.72	0.979	30.08	0.933	24.84	0.853	23.57	0.778	0.20	9.8
XVFI [35]	33.99	0.968	38.37	0.987	34.42	0.973	29.52	0.928	24.88	0.854	28.96	0.887	**0.21**	**5.7**
IFRNet [16]	36.16	0.980	40.10	**0.991**	36.12	0.978	30.63	0.936	25.27	0.861	27.53	0.847	0.79	19.7
EMA-VFI [45]	**36.64**	**0.982**	39.98	**0.991**	36.09	0.980	**30.94**	0.939	**25.69**	**0.866**	29.89	0.896	0.91	66.0
Ours	35.01	0.977	40.09	0.990	35.89	0.979	30.54	0.935	25.33	0.860	29.90	0.901	0.91	6.0

Table 5. Additional quantitative comparisons on fixed time interpolation datasets and the complexity of SOTA models.

point satisfying the distance ratio $\frac{\overline{AX}}{\overline{BX}} = k$ and the angle $\angle AXB = \theta$ is unique.

II) If $k \neq 1$ (Fig. 9b), this locus forms an Apollonian circle [25]. In this case, if $k > 1$, point A is outside the circle, and point B is inside the circle. Conversely, if $k < 1$, point B is outside the circle, and point A is inside the circle. In any case, the resulting circle has only one intersection with the arc $\overset{\frown}{AB}$, thus the point satisfying the distance ratio $\frac{\overline{AX}}{\overline{BX}} = k$ and the angle $\angle AXB = \theta$ is unique.

By I) and II), for given two fixed points A and B, a positive real number k, it is concluded that the point X satisfying the distance ratio $\frac{\overline{AX}}{\overline{BX}} = k$ and the angle $\angle AXB = \theta$ is unique. □

B. Additional Experimental Results

B.1. Quantitative Results

We provided pixel-centric metrics (PSNR and SSIM) measured on SNU-FILM-arb [6] datasets and additional arbitrary time interpolation on XTest [35] to interpolate $\times 8$ frames, which are tabulated in Tab. 4. As discussed in Sec. 4.3 of the main paper, while our BiM-VFI underperforms in pixel-centric metrics, it consistently outperforms the other SOTA methods on XTest [35] $\times 8$ interpolation in terms of perceptual metrics, such as LPIPS and STLPIPS.

In Tab. 5, we also provided pixel-centric metrics measured on fixed-time datasets (Vimeo 90K-triplet [41], SNU-FILM [3], and XTest [35] single) and complexity comparisons between other SOTA methods.

B.2. Computational complexity

We additionally provide below #'s of parameters and FLOPs on each component for 256×256-sized images.

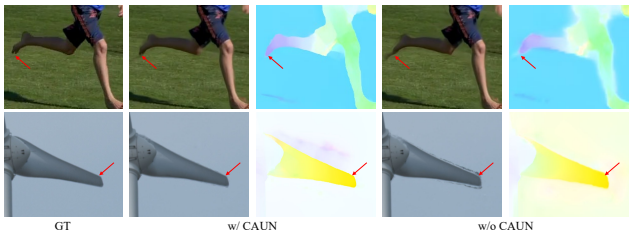
	MFE	CFE	BiMFN	CAUN	SN	Total
#Params(M)	0.58	0.58	3.41	0.61	1.7	6.88
FLOPs(G)	12.02	12.02	18.81	11	24.64	78.49

We measured FLOPs for interpolating 1280×720-sized source images and the total parameters used in the methods. As shown in Tab. 5, our BiM-VFI effectively reduced the number of parameters by employing a recurrent pyramid architecture, while having moderate computational complexity among the other SOTA methods in terms of FLOPs.

	GIMM-VFI-R	EMA-VFI	UPR	AMT	Ours
#Params(M)	19.73	65.66	6.56	30.64	**6.88**
FLOPs(G)	9187	1714	1228	2395	**1177**
Runtime(ms)	494	104	**53**	183	151

B.3. Additional ablation study

We provide below the detection performance of small objects and object boundaries with and without CAUN module. As shown, the CAUN can help capture well small toes (top) and detect tight boundaries of the windmill blade (bottom), while failing without it.



Also, as mentioned in *Suppl.*, while our KDVCF increases the training time from 2.5 to 4 days using 4 A100 GPUs due to the $\mathcal{P}_\mathcal{T}$ process, it does not increase the inference time. We also compared the #'s of parameters, FLOPs, and runtime (measured on an A100 GPU with 1280×768-sized images) in the below table. It can be noted that our BiM-VFI has a low number of parameters, showing moderate runtime.

B.4. User Study

We conducted a user study to show that our BiM-VFI with uniform motion BiM perceptually outperforms other SOTA methods. 21 participants were asked to choose

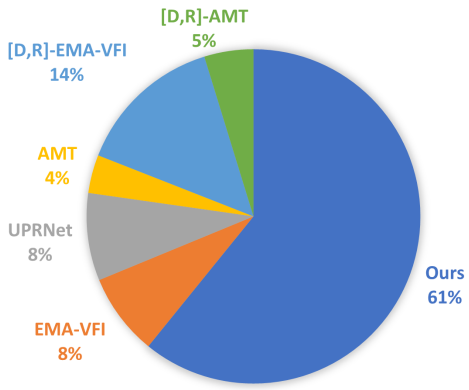


Figure 10. Preference of interpolated videos between our BiM-VFI and the other SOTA models measured by user study.

the best-interpolated videos among AMT-S [18], UPR-Net [13], EMA-VFI [45], [D,R]-AMT-S, and [D,R]-EMA-VFI, where [D,R] indicates that distance indexing and iterative reference-based estimation, proposed by Zhong *et al*. [44], are plugged into the method. We used 9 test videos for blind subjective tests where the six 8×-interpolated videos for the six VFI methods including our BiM-VFI are displayed simultaneously on the same screens for each test video. In order to remove any subjective bias to specific VFI methods, the six 8×-interpolated videos for each test video are randomly ordered and presented to the participants in the blind subjective test.

As shown in Fig. 10, our BiM-VFI dominantly outperforms the other SOTA methods in the subjective tests, by 61% preference against the other six VFI methods.

B.5. Qualitative Results

We provided additional qualitative comparisons with the SOTA methods in SNU-FILM-arb [6] extreme datasets.

C. Limitation

Our KDVCF requires approximately twice the training time compared to training solely with the student process, as both the teacher and student processes are trained simultaneously. However, the model trained with KDVCF demonstrated its effectiveness in perceptual metrics compared to models supervised with pre-trained flow models or without flow supervision. It is also noteworthy that only the student process remains during inference, so the inference runtime is the same as that of models trained without KDVCF.

Figure 11. Additional qualitative comparisons on SNU-FILM-arb [6] extreme datasets.

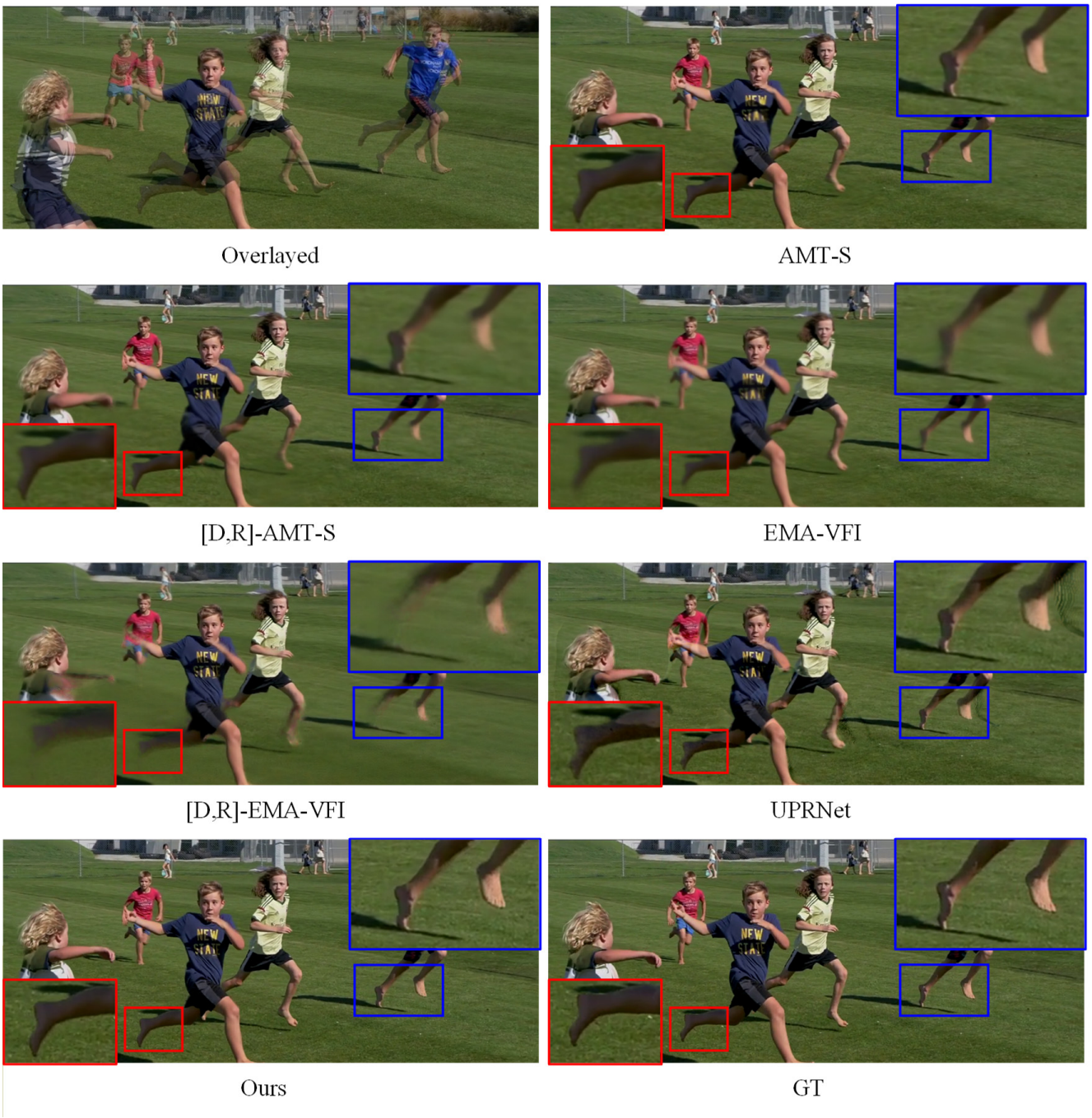


Figure 12. Additional qualitative comparisons on SNU-FILM-arb [6] extreme datasets.

Figure 13. Additional qualitative comparisons on SNU-FILM-arb [6] extreme datasets.

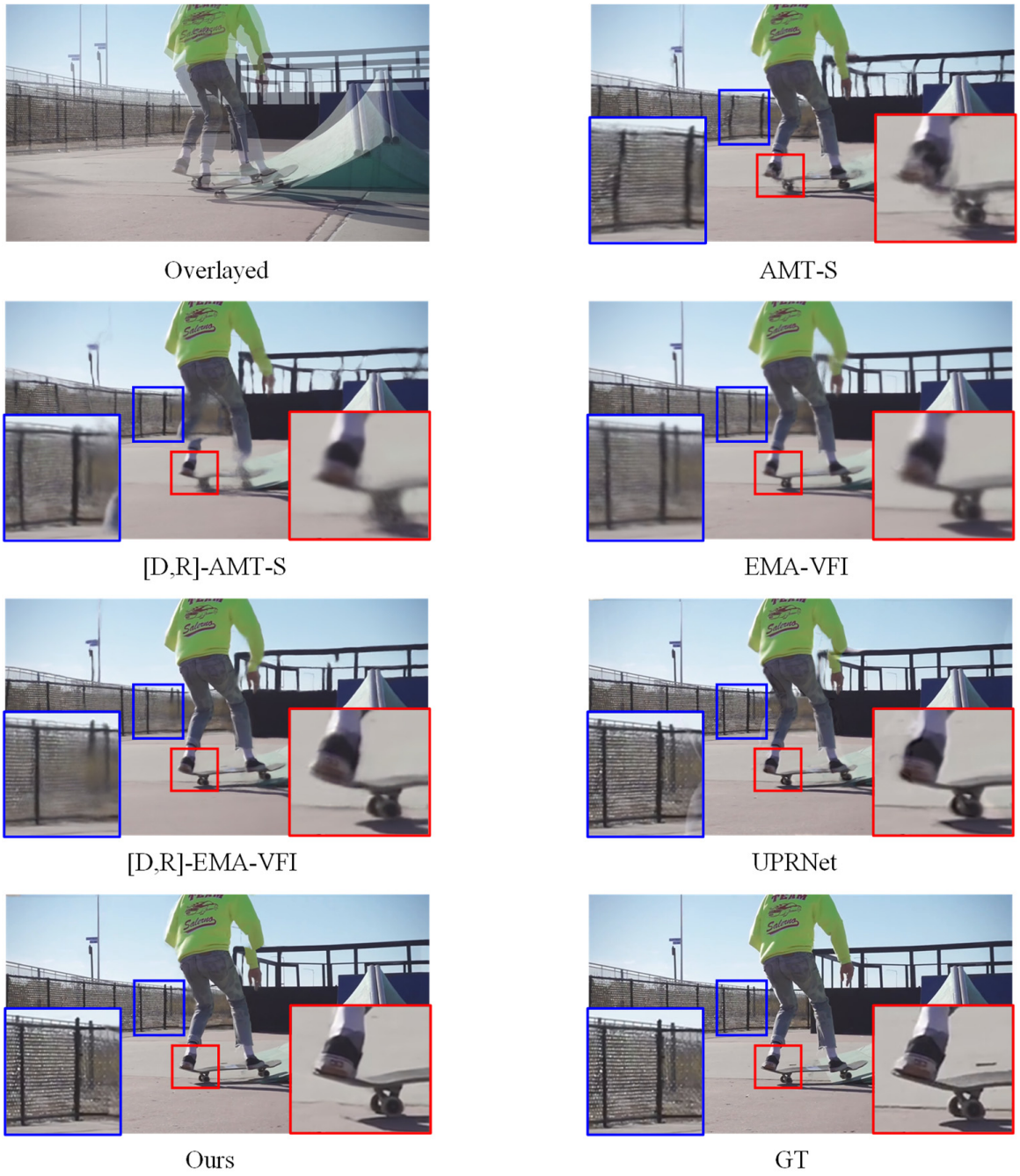


Figure 14. Additional qualitative comparisons on SNU-FILM-arb [6] extreme datasets.