Supplementary Material: Boost Video Frame Interpolation via Motion Adaptation

Haoning Wu[1]
whn15698781666@sjtu.edu.cn
Xiaoyun Zhang[†1]
xiaoyun.zhang@sjtu.edu.cn
Weidi Xie[1,2]
weidi@sjtu.edu.cn
Ya Zhang[1,2]
ya_zhang@sjtu.edu.cn
Yanfeng Wang[†1,2]
wangyanfeng622@sjtu.edu.cn

[1] Coop. Medianet Innovation Center, Shanghai Jiao Tong University, China
[2] Shanghai AI Laboratory, China

In this supplementary document, we start by giving more details on the implementation details of our proposed cycle-consistency motion adaptation strategy and plug-in adapter module in Section 1. Then, we demonstrate the ablation study on the influence of more adaptation steps and how cycle-consistency adaptation steadily works on several video frame interpolation (VFI) models in Section 2. Next, we supplement more qualitative comparison results on several benchmarks in Section 3. Finally, we illustrate the limitation of our method and our future work in Section 4.

1 Implementation Details

Cycle-consistency adaptation. Considering that each sample in common datasets [2, 7, 8] typically comprises a septuplet, with odd frames as input, denoted as $\{\mathcal{I}_1, \mathcal{I}_3, \mathcal{I}_5, \mathcal{I}_7\}$. We divide it into two triplets, *i.e.,* $\{\mathcal{I}_1, \mathcal{I}_3, \mathcal{I}_5\}$ and $\{\mathcal{I}_3, \mathcal{I}_5, \mathcal{I}_7\}$ and perform motion adaptation on these two triplets to adapt the model to the motion characteristics of the current video sequence. To be specific, we take the middle frame of each triplet as the ground truth and calculate loss between the generated frames and the target frames, denoted as $\mathcal{L}_3 = ||\hat{\mathcal{I}}_3 - \mathcal{I}_3||$ and $\mathcal{L}_5 = ||\hat{\mathcal{I}}_5 - \mathcal{I}_5||$. Then we take their average $\mathcal{L} = \frac{1}{2}(\mathcal{L}_3 + \mathcal{L}_5)$ to backward gradients for parameters updates, and regard this entire process as one adaptation step. This cycle-consistency adaptation strategy has been employed in end-to-end finetuning as well as plug-in adapter finetuning, and theoretically, it can be extended to longer video sequences.

Plug-in adapter. For flow-based VFI methods [3, 4, 5], we can freeze the pre-trained parameters and simply use our proposed lightweight plug-in adapter to adjust the estimated motion flow. Concretely, we reuse the convolutional layers from the pre-trained model's motion estimation module to align the channel dimensions of the extracted visual features with that of

†: Corresponding author.

Methods	#Adapt Steps	Vimeo90K [8]	DAVIS [7]	SNU-FILM [4]			
				Easy	Medium	Hard	Extreme
SepConv-ours-e2e	0	33.72 / 0.9639	26.65 / 0.8611	40.21 / 0.9909	35.45 / 0.9785	29.62 / 0.9302	24.16 / 0.8457
	10	33.96 / 0.9650	26.83 / 0.8639	40.41 / 0.9911	35.71 / 0.9794	29.80 / 0.9313	24.26 / 0.8479
	20	34.17 / 0.9659	26.93 / 0.8649	40.56 / 0.9913	35.91 / 0.9800	29.88 / 0.9313	24.32 / 0.8484
	30	**34.29 / 0.9662**	**26.98 / 0.8651**	**40.66 / 0.9914**	**36.04 / 0.9804**	**29.93 / 0.9318**	**24.36 / 0.8491**
EDSC-ours-e2e	0	34.55 / 0.9677	26.83 / 0.8578	40.66 / 0.9915	35.77 / 0.9795	29.75 / 0.9301	24.12 / 0.8420
	10	34.73 / 0.9685	26.96 / 0.8600	40.88 / 0.9917	35.98 / 0.9803	29.85 / 0.9313	24.19 / 0.8436
	20	34.94 / 0.9693	27.07 / 0.8618	40.98 / 0.9919	36.18 / 0.9811	29.95 / 0.9322	24.23 / 0.8445
	30	**35.06 / 0.9699**	**27.14 / 0.8630**	**41.09 / 0.9920**	**36.33 / 0.9817**	**30.02 / 0.9334**	**24.28 / 0.8455**
RIFE-ours-e2e	0	35.28 / 0.9704	27.61 / 0.8760	40.75 / 0.9916	36.18 / 0.9808	30.30 / 0.9368	24.62 / 0.8531
	10	35.57 / 0.9717	27.81 / 0.8798	40.95 / 0.9918	36.42 / 0.9815	30.49 / 0.9386	24.71 / 0.8549
	20	35.80 / 0.9728	27.99 / 0.8830	41.10 / 0.9920	36.71 / 0.9827	30.67 / 0.9408	24.79 / 0.8570
	30	**35.93 / 0.9733**	**28.10 / 0.8850**	**41.20 / 0.9924**	**36.94 / 0.9835**	**30.83 / 0.9430**	**24.87 / 0.8589**
RIFE-ours-plugin	0	35.33 / 0.9706	27.64 / 0.8765	40.66 / 0.9915	36.12 / 0.9807	30.32 / 0.9371	24.67 / 0.8539
	10	35.56 / 0.9714	27.76 / 0.8871	40.99 / 0.9918	36.55 / 0.9825	30.48 / 0.9387	24.64 / 0.8533
	20	35.61 / 0.9719	27.79 / 0.8786	41.01 / 0.9919	36.55 / 0.9824	30.65 / 0.9404	24.79 / 0.8555
	30	**35.71 / 0.9722**	**27.88 / 0.8799**	**41.12 / 0.9920**	**36.77 / 0.9832**	**30.74 / 0.9404**	**24.84 / 0.8590**
IFRNet-ours-e2e	0	35.86 / 0.9729	28.03 / 0.8851	40.91 / 0.9916	36.18 / 0.9808	30.30 / 0.9368	24.62 / 0.8531
	10	36.38 / 0.9753	28.45 / 0.8936	41.21 / 0.9921	37.03 / 0.9832	31.10 / 0.9440	25.03 / 0.8634
	20	36.60 / 0.9759	28.69 / 0.8979	41.40 / 0.9923	37.36 / 0.9844	31.37 / 0.9476	25.18 / 0.8676
	30	**36.68 / 0.9760**	**28.78 / 0.8995**	**41.48 / 0.9923**	**37.57 / 0.9850**	**31.45 / 0.9482**	**25.22 / 0.8694**
IFRNet-ours-plugin	0	35.86 / 0.9729	28.02 / 0.8850	40.91 / 0.9918	36.58 / 0.9816	30.74 / 0.9404	24.84 / 0.8590
	10	36.01 / 0.9734	28.16 / 0.8825	41.06 / 0.9920	36.92 / 0.9834	30.88 / 0.9404	24.93 / 0.8599
	20	36.11 / 0.9738	28.26 / 0.8875	41.11 / 0.9921	37.01 / 0.9837	30.95 / 0.9414	24.96 / 0.8599
	30	**36.14 / 0.9742**	**28.33 / 0.8888**	**41.18 / 0.9923**	**37.14 / 0.9844**	**31.03 / 0.9419**	**24.97 / 0.8600**
UPRNet-ours-e2e	0	36.07 / 0.9735	28.38 / 0.8914	41.01 / 0.9919	36.80 / 0.9819	31.22 / 0.9422	25.39 / 0.8648
	10	36.68 / 0.9758	28.84 / 0.8997	41.31 / 0.9923	37.24 / 0.9836	31.66 / 0.9464	25.64 / 0.8699
	20	36.84 / 0.9766	29.07 / 0.9043	41.42 / 0.9924	37.52 / 0.9849	31.89 / 0.9500	25.85 / 0.8755
	30	**36.90 / 0.9768**	**29.15 / 0.9062**	**41.48 / 0.9925**	**37.66 / 0.9855**	**32.00 / 0.9519**	**25.99 / 0.8798**
UPRNet-ours-plugin	0	36.04 / 0.9734	28.31 / 0.8896	41.00 / 0.9919	36.77 / 0.9818	31.18 / 0.9418	25.38 / 0.8645
	10	36.44 / 0.9751	28.69 / 0.8945	41.32 / 0.9923	37.38 / 0.9843	31.64 / 0.9448	**25.69** / 0.8705
	20	36.52 / 0.9754	28.78 / 0.8963	41.37 / 0.9923	**37.59** / 0.9847	31.70 / 0.9461	25.68 / 0.8705
	30	**36.57 / 0.9756**	**28.90 / 0.8989**	**41.40 / 0.9924**	37.55 / **0.9849**	**31.75 / 0.9470**	**25.69 / 0.8706**

Table 1. Additional ablation study on adaptation strategies and steps. We inherit pre-trained VFI models [1, 3, 4, 5, 6], then perform end-to-end (e2e) and plug-in adapter (plugin) adaptation with different adaptation steps, comparing their performances on three commonly used benchmarks.

motion flow. Then a simple 1×1 convolution layer is utilized to predict pixel-wise weights α and biases β for motion adaptation, without introducing any additional activation layers.

2 Additional Ablation Study

In this section, we present more ablation results for our proposed optimisation-based VFI, including the motion adaptation strategy and the adaptation steps. Specifically, we conduct test-time motion adaptation experiments on three benchmarks, namely Vimeo90K [8], DAVIS [7], and SNU-FILM [2] with five VFI models, SepConv [6], EDSC [1], RIFE [3], IFRNet [5] and UPRNet [4], in two manners, end-to-end (e2e) and plug-in adapter (plugin) boosted. We further investigate the impact of different adaptation steps on performance.

Adaptation Strategy. The results in Table 1 further demonstrate the conclusions we have drawn in our paper: The performance of pre-trained VFI models can be enhanced via end-to-end motion adaptation during test-time. Furthermore, for flow-based methods such as RIFE [3], IFRNet [5] and UPRNet [4], the proposed plug-in adapter module can effectively

boost the performance of pre-trained models. Moreover, it achieves comparable performance improvements to that of end-to-end finetuning, with the same number of adaptation steps.

Adaptation Steps. As indicated in Table 1, with adaptation steps increasing, both end-to-end finetuning and plug-in adapter finetuning effectively boost the performance of VFI models, thus confirming the effectiveness of our proposed optimisation-based VFI. Furthermore, while cycle-consistency adaptation can steadily boost performance, models exhibit significant improvements within the first 10 adaptation steps, with performance gains gradually approaching saturation after more steps. Therefore, taking both efficiency and performance into consideration, we have chosen 10 steps of adaptation as our default setting.

3 More Qualitative Results

We provide more qualitative results in Figure 1, Figure 2 and Figure 3. Concretely, we compare UPRNet-ours-plugin (10-step-adaptation) and UPRNet-ours-e2e++ (30-step-adaptation) with other state-of-the-art VFI methods on benchmark datasets with different motion characteristics. We can observe that other VFI methods tend to generate images with motion blur or missing object details, while our method can consistently achieve better perceptual quality, *i.e.,* maintaining the object shape and textual information, and synthesizing shaper details. In summary, our proposed motion adaptation can steadily boost VFI models.

4 Limitation & Future Work

In this paper, we have demonstrated the effectiveness of our proposed cycle-consistency motion adaptation and lightweight adapter as a plug-in module for VFI. However, some limitations remain: it still suffers from the well-known problem of using test-time adaptation, that is, time consumption is still non-negligible, which poses limitations in increasing adaptation steps to pursue higher performance. Besides, due to the limitation of data and computing resources, our motion adaptation strategy is currently restricted to 2-frame VFI models and has not been extended to frame synthesis at arbitrary temporal positions. Moreover, in general, our proposed method is beneficial for better handling motions that are regular, such as rotations and large-scale motions, however, it remains challenging for irregular motions, such as sudden camera shaking, or illumination changes, due to the difficulty of inferring priors. Our future work will investigate more efficient plug-in adapter architecture and further extend it to flexible VFI models with multiple frames as inputs, which has the potential to further boost the performance and generalisation ability of video frame interpolation models.

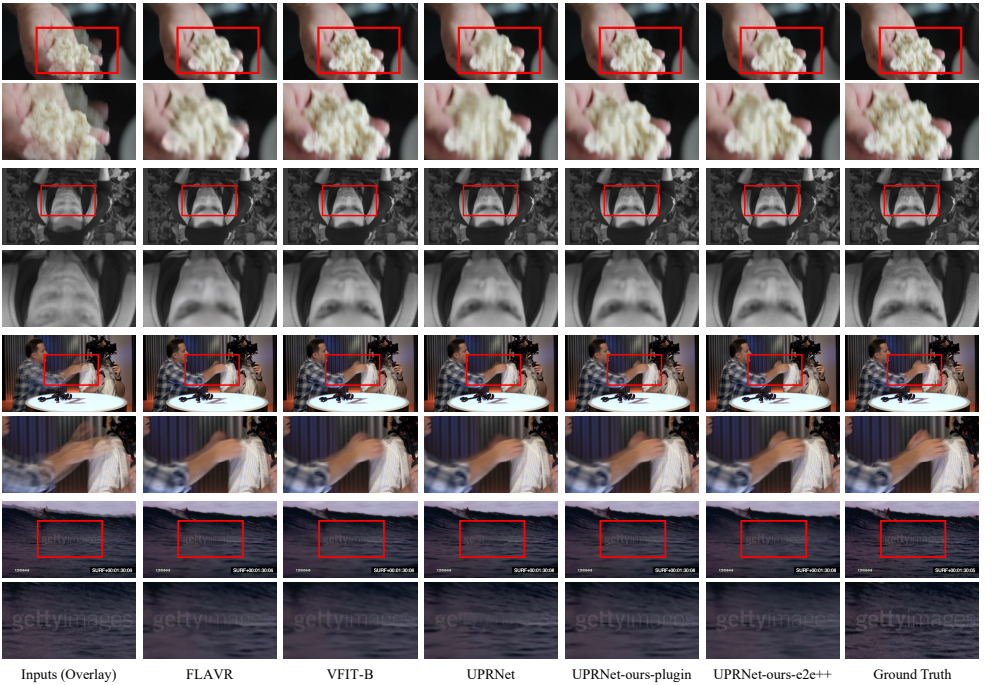

Figure 1. Qualitative comparison against the state-of-the-art VFI algorithms. We show visualization on Vimeo90K [8] benchmark for comparison. The patches for careful comparison are marked with red in the original images. Our boosted models can generate higher-quality results with clearer structures and fewer distortions.

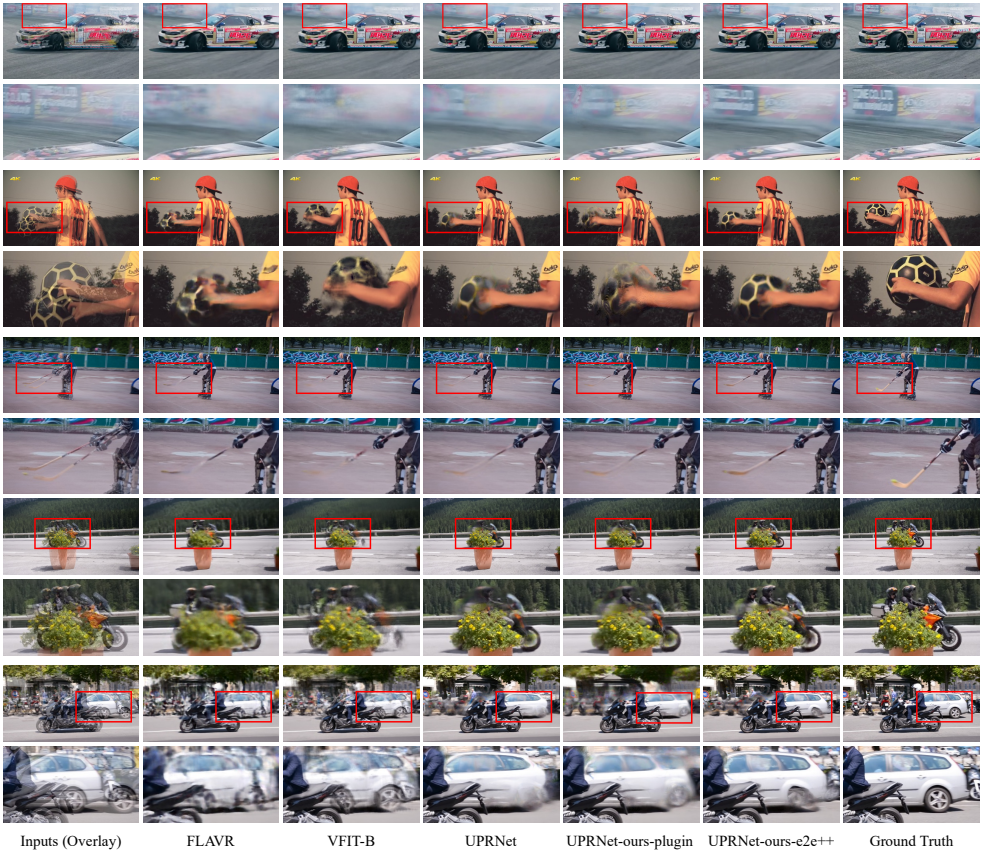

Figure 2. Qualitative comparison against the state-of-the-art VFI algorithms. We show visualization on DAVIS [7] benchmark for comparison. The patches for careful comparison are marked with red in the original images. Our boosted models can generate higher-quality results with clearer structures and fewer distortions.

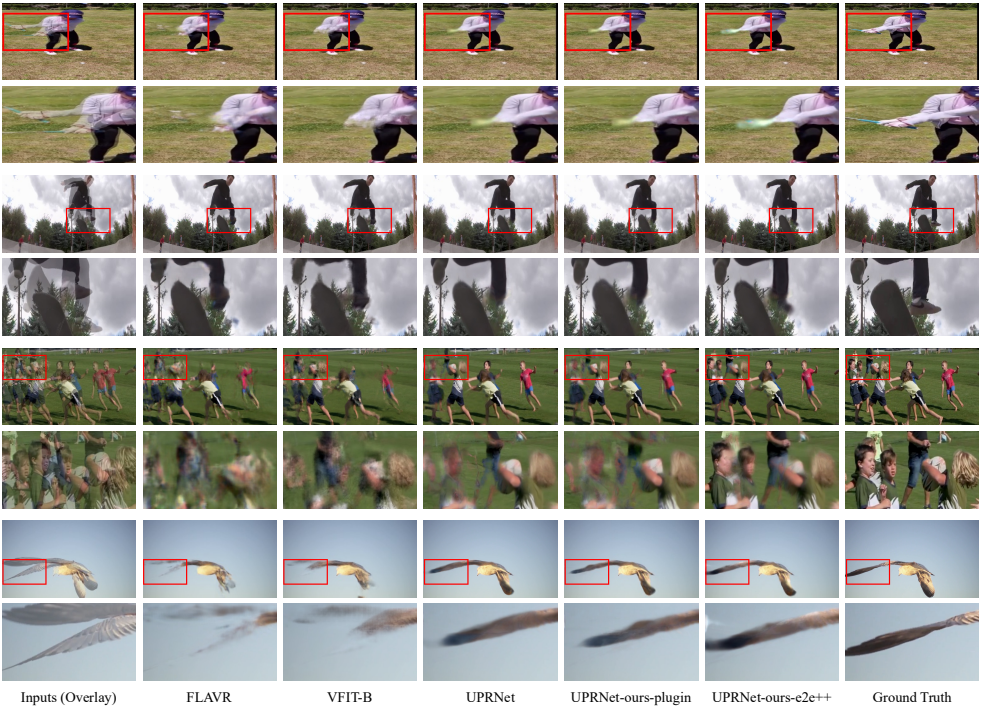

Figure 3. Qualitative comparison against the state-of-the-art VFI algorithms. We show visualization on SNU-FILM [2] benchmark for comparison. The patches for careful comparison are marked with red in the original images. Our boosted models can generate higher-quality results with clearer structures and fewer distortions.

References

[1] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7029–7045, 2021.

[2] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10663–10671, 2020.

[3] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *Proceedings of the European Conference on Computer Vision*, pages 624–642, 2022.

[4] Xin Jin, Longhai Wu, Jie Chen, Youxin Chen, Jayoon Koo, and Cheul-hee Hahm. A unified pyramid recurrent network for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[5] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1969–1978, 2022.

[6] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the International Conference on Computer Vision*, pages 261–270, 2017.

[7] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

[8] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127: 1106–1125, 2019.